AWS Elemental MediaStore

ユーザーガイド

AWS Elemental MediaStore: ユーザーガイド

Table of Contents

MediaStore とは何ですか? 1
概念と用語 1
関連サービス 2
MediaStore へのアクセス 2
料金表 3
リージョンとエンドポイント 3
セットアップ 4
AWS にサインアップする 4
管理者の IAM ユーザーの作成 4
管理者以外の IAM ユーザーの作成 5
ステップ 1: ユーザーグループを作成する 6
ステップ 2: ユーザーを作成する 6
開始方法 8
ステップ 1: AWS Elemental MediaStore にアクセスする 8
ステップ 2: コンテナを作成する 8
ステップ 3: オブジェクトをアップロードする 9
ステップ 4: オブジェクトにアクセスする 9
コンテナ 10
コンテナ名に関するルール 10
コンテナの作成 10
コンテナの詳細の表示 11
コンテナのリストの表示 12
コンテナの削除 12
ポリシー 14
コンテナポリシー 14
コンテナポリシーを表示する 14
コンテナポリシーを編集する 15
コンテナポリシーの例 16
CORS ポリシー 22
ユースケースのシナリオ 22
CORS ポリシーの追加 23
CORS ポリシーの表示 24
CORS ポリシーの編集 24
CORS ポリシーの削除 25
トラブルシューティング 26
CORS ポリシーの例 26
オブジェクトのライフサイクルポリシー 27
オブジェクトのライフサイクルポリシーのコンポーネント 28
オブジェクトのライフサイクルポリシーを追加する 32
オブジェクトのライフサイクルポリシーを表示する 33
オブジェクトのライフサイクルポリシーを編集する 34
オブジェクトのライフサイクルポリシーを削除する 35
オブジェクトのライフサイクルポリシーの例 35
メトリクスポリシー 38
メトリクスポリシーの追加 39
メトリクスポリシーの表示 39
メトリクスポリシーの編集 39
メトリクスポリシーの例 42
フォルダ 45
フォルダ名に関するルール 45
フォルダの作成 46
フォルダの削除 46
オブジェクト 47
オブジェクトのアップロード 47

リストの表示 48
オブジェクトの詳細の表示 50
オブジェクトのダウンロード 50
オブジェクトの削除 51
1 つのオブジェクトを削除する 52
コンテナを空にする 52
Security 54
データ保護 54
Identity and access management 55
Audience 55
アイデンティティを使用した認証 56
ポリシーを使用したアクセスの管理 58
詳細はこちら 59
MediaStore と IAM の連携 60
アイデンティティベースのポリシーの例 63
リソースベースのポリシーの例 65
トラブルシューティング 68
ログ記録とモニタリング 70
Amazon CloudWatch アラーム 70
AWS CloudTrail ログ 70
AWS Trusted Advisor 70
コンプライアンス検証 70
弾力 71
インフラストラクチャセキュリティ 71
モニタリングとタグ付け 72
CloudTrail による API コールのログ記録 72
CloudTrail での MediaStore 情報 73
例: ログファイルエントリ 74
CloudWatch によるモニタリング 75
CloudWatch Logs 75
CloudWatch イベント 81
CloudWatch のメトリクス 84
タグ付け 86
AWS Elemental MediaStore でサポートされているリソース 87
タグの命名規則と使用規則 87
タグを管理する 88
CDN の使用 89
コンテナへのアクセスを CloudFront に許可する 89
MediaStore による HTTP キャッシュの操作 90
条件付きリクエスト 91
クォータ 92
関連情報 94
ドキュメント履歴 95
AWS の用語集 98
..... xcix

AWS Elemental MediaStore とは何ですか?

AWS Elemental MediaStore は、ライブ配信に必要な高パフォーマンスと即時の整合性を実現する、動画配信およびストレージサービスです。MediaStore では、動画アセットをコンテナのオブジェクトとして管理し、信頼できるクラウドベースのメディアワークフローを構築できます。

サービスを使用するには、エンコーダーやデータフィードなどのソースからオブジェクトを MediaStore で作成したコンテナにアップロードします。

MediaStore は、断片化された動画ファイルを保存するための優れたオプションで、厳密な整合性、低レイテンシーの読み取りと書き込み、および同時リクエストの大量処理が要求される場合に最適です。動画のライブストリーミングを配信しない場合は、代わりに Amazon Simple Storage Service (Amazon S3) を使用することをおすすめします。

トピック

- AWS Elemental MediaStore の概念と用語 (p. 1)
- 関連サービス (p. 2)
- AWS Elemental MediaStore へのアクセス (p. 2)
- AWS Elemental MediaStore の料金 (p. 3)
- の地域とエンドポイント AWS Elemental MediaStore (p. 3)

AWS Elemental MediaStore の概念と用語

ARN

Amazon リソースネーム。

本文

オブジェクトにアップロードするデータ。

(バイト) 範囲

対象のオブジェクトデータのサブセット。詳細については、HTTP 仕様の「range (範囲)」を参照してください。

Container

オブジェクトを保持する名前空間。コンテナのエンドポイントを使用してオブジェクトの書き込みと取得、アクセスポリシーのアタッチを行うことができます。

エンドポイント

エントリーポイント MediaStore HTTPSルートURLとして提供されるサービス。

ETag

エンティティタグ。オブジェクトデータのハッシュです。

フォルダ

コンテナの区分。フォルダはオブジェクトおよび他のフォルダを保持できます。

Item (項目)

オブジェクトとフォルダを指す用語。

オブジェクト

アセット。Amazon S3 オブジェクトに似ています。オブジェクトは、MediaStore に保存される基本エンティティです。すべてのファイルタイプがサポートされます。

配信サービス

MediaStore は、メディアコンテンツ配信の起点であるため、配信サービスと見なされます。

パス

オブジェクトまたはフォルダの一意の識別子。コンテナ内の位置を示します。

パーツ

オブジェクトのデータ (チャンク) のサブセット。

ポリシー

IAM ポリシー。

Resource

操作可能な AWS のエンティティ。各 AWS リソースには、一意の識別子として機能する Amazon リソースが割り当てられています。MediaStore では、これはリソースとその ARN 形式です。

- コンテナ: `aws:mediastore:region:account-id:container/:containerName`

関連サービス

- Amazon CloudFront は、グローバルコンテンツ配信ネットワーク (CDN) サービスであり、データやビデオを視聴者に安全に配信します。CloudFront では、最大限のパフォーマンスでコンテンツを配信します。詳細については、Amazon CloudFront 開発者ガイド を参照してください。
- AWS CloudFormationは、AWSのモデル化およびセットアップに役立つサービスです。使用するすべての AWS リソース (MediaStore コンテナなど) を説明するテンプレートを作成すれば、AWS CloudFormation がそれらのリソースのプロビジョニングや設定をお客様に代わって行います。AWS リソースを個別に作成、設計して、それぞれの依存関係を考える必要はありません。AWS CloudFormation がすべてを処理します。詳細については、AWS CloudFormation ユーザーガイド を参照してください。
- AWS CloudTrailは、アカウントで AWS マネジメントコンソール、AWS CLI、その他のサービスからの CloudTrailAPI コールをモニタリングできるサービスです。詳細については、AWS CloudTrail User Guide を参照してください。
- Amazon CloudWatch は、AWS クラウドリソースと、AWS で実行するアプリケーションのモニタリングサービスです。CloudWatch イベント を使用して、MediaStore のコンテナやオブジェクトのステータスの変化を追跡します。詳細については、Amazon CloudWatch ドキュメントを参照してください。
- AWS Identity and Access Management (IAM) は、AWS リソースへのアクセスを安全に制御するためのウェブサービスです。IAM では、どのユーザーが AWS リソースを使用できるかを制御し (認証)、さらに、どのリソースをユーザーがどのように使用できるかを制御します (権限付与)。詳細については、セットアップ (p. 4) を参照してください。
- Amazon Simple Storage Service (Amazon S3) は、どこからでも任意の量のデータの保存および取得できるオブジェクトストレージです。詳細については、Amazon S3 ドキュメントを参照してください。

AWS Elemental MediaStore へのアクセス

次のいずれかの方法で MediaStore にアクセスできます。

- AWS マネジメントコンソール - このガイドの手順では、AWS マネジメントコンソールを使用して MediaStore のタスクを実行する方法について説明しています。アクセスするには MediaStore コンソールの使用:

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

- AWS Command Line Interface - 詳細については、AWS Command Line Interface ユーザーガイド を参照してください。アクセスするには MediaStore CLIエンドポイントの使用:

```
aws mediastore
```

- MediaStore API – SDK が提供されていないプログラミング言語を使用している場合、API アクションの情報と API リクエストの作成方法については、AWS Elemental MediaStore API リファレンスを参照してください。アクセスするには MediaStore REST APIエンドポイントの使用:

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

- AWS(AWS) SDKs – AWSがSDKを提供するプログラミング言語を使用している場合は、SDKを使用して MediaStore. SDKs 認証を簡素化し、開発環境と簡単に統合し、 MediaStore コマンド。詳細については、「アマゾン ウェブ サービスのツール」を参照してください。
- Windows用AWSツール PowerShell – 詳細については、 AWS Tools for Windows PowerShell ユーザーガイド.

AWS Elemental MediaStore の料金

他の AWS 製品と同様、MediaStore を使用するための契約や最低契約金はありません。コンテンツをサービスに取り込んだときに 1 GB あたりの取り込み料金が発生し、コンテンツをサービスに保存したときに 1 GB あたりの月額料金が発生します。詳細については、「AWS Elemental MediaStore 料金表」を参照してください。

の地域とエンドポイント AWS Elemental MediaStore

アプリケーションでのデータ· レーテンシーを低減するには、 MediaStore は、お客様のリクエストを作成するための地域エンドポイントを提供します。

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

AWS 地域の全リストを表示するには、 MediaStore が利用可能です。を参照してください。 AWS Elemental MediaStore エンドポイントとクォータ 参照してください。

AWS Elemental MediaStore のセットアップ

AWS Elemental MediaStore の使用を開始する前に、AWS にサインアップして (まだ AWS アカウントをお持ちでない場合)、MediaStore へのアクセスを許可するための IAM ユーザーとロールを作成する必要があります。これには、自分自身の IAM ロールを作成することが含まれます。

トピック

- AWS にサインアップする (p. 4)
- 管理者の IAM ユーザーの作成 (p. 4)
- 管理者以外の IAM ユーザーの作成 (p. 5)

AWS にサインアップする

AWS アカウントをお持ちでない場合は、次に説明する手順に従ってアカウントを作成してください。

AWS にサインアップするには

1. https://aws.amazon.com/ を開き、[AWS アカウントの作成] を選択します。
2. オンラインの手順に従います。

管理者の IAM ユーザーの作成

AWS アカウントを初めて作成する場合は、このアカウントのすべての AWS サービスとリソースに対して完全なアクセス権限を持つシングルサインインアイデンティティで始めます。このアイデンティティは AWS アカウント ルートユーザー と呼ばれ、アカウントの作成に使用した E メールアドレスとパスワードでのサインインによりアクセスします。強くお勧めしているのは、日常的なタスクには、それが管理者タスクであっても、ルートユーザーを使用しないことです。代わりに、最初の IAM ユーザーを作成するためだけに ルートユーザー を使用するというベストプラクティスに従います。その後、ルートユーザー認証情報を安全な場所に保管し、それらを使用して少数のアカウントおよびサービス管理タスクのみを実行します。

この手順では、AWS アカウントのルートユーザー を使用して最初の IAM ユーザーを作成します。この IAM ユーザーを管理者グループに追加し、アカウントですべてのサービスおよび各サービスのリソースにアクセスできるようにします。次回 AWS アカウントにアクセスするときは、この IAM ユーザーの認証情報を使用してサインインします。

アクセス許可が制限された IAM ユーザーを作成するには、「the section called “管理者以外の IAM ユーザーの作成” (p. 5)」を参照してください。

自分用の管理者ユーザーを作成し、そのユーザーを管理者グループに追加するには (コンソール)

1. ルートユーザー を選択し AWS アカウントの E メールアドレスを入力して、アカウントの所有者としてIAM コンソールにサインインします。次のページでパスワードを入力します。

Note

以下の**Administrator**IAMユーザーの使用に関するベストプラクティスに従い、ルートユーザー認証情報を安全な場所に保管しておくことを強くお勧めします。ルートユーザーとしてサインインして、少数のアカウントおよびサービス管理タスクのみを実行します。

2. ナビゲーションペインで [Users]、[Add user] の順に選択します。
3. [ユーザー名] に「**Administrator**」と入力します。
4. [AWS マネジメントコンソール access (アクセス)] の横にあるチェックボックスをオンにします。[Custom password (カスタムパスワード)] を選択し、その後テキストボックスに新しいパスワードを入力します。
5. (オプション) AWS では、デフォルトで、新しいユーザーに対して初回のサインイン時に新しいパスワードを作成することが必要です。必要に応じて [User must create a new password at next sign-in (ユーザーは次回のサインイン時に新しいパスワードを作成する必要がある)] のチェックボックスをオフにして、新しいユーザーがサインインしてからパスワードをリセットできるようにできます。
6. [Next: Permissions (次へ: アクセス許可)] を選択します。
7. [Set permissions (アクセス許可の設定)] で、[Add user to group (ユーザーをグループに追加)] を選択します。
8. [Create group] を選択します。
9. [グループの作成] ダイアログボックスで、[グループ名] に「**Administrators**」と入力します。
10. [Filter policies (フィルタポリシー)] を選択し、その後 [AWS managed -job function (AWS 管理ジョブの機能] を選択してテーブルのコンテンツをフィルタリングします。
11. ポリシーリストで、[AdministratorAccess] のチェックボックスをオンにします。次に、[Create group] を選択します。

    Note

    `AdministratorAccess` アクセス許可を使用して、AWS Billing and Cost Management コンソールを使用する前に、IAM ユーザーおよびロールの請求へのアクセスをアクティブ化する必要があります。これを行うには、請求コンソールへのアクセスの委任に関するチュートリアルのステップ 1 の手順に従ってください。
12. グループのリストに戻り、新しいグループのチェックボックスをオンにします。必要に応じて [Refresh] を選択し、リスト内のグループを表示します。
13. [次へ: タグ] を選択します。
14. (オプション) タグをキー - 値のペアとしてアタッチして、メタデータをユーザーに追加します。IAM でのタグの使用の詳細については、IAM ユーザーガイド の「IAM エンティティのタグ付け」を参照してください。
15. [Next: Review] を選択して、新しいユーザーに追加するグループメンバーシップのリストを表示します。続行する準備ができたら、[Create user] を選択します。

この同じプロセスを繰り返して新しいグループとユーザーを作成し、AWS アカウントのリソースへのアクセス権をユーザーに付与できます。ポリシーを使用して特定の AWS リソースに対するユーザーのアクセス許可を制限する方法については、「アクセス管理」と「ポリシーの例」を参照してください。

管理者以外の IAM ユーザーの作成

アカウントの管理者グループに属するユーザーは、そのアカウントのすべての AWS のサービスとリソースにアクセスできます。このセクションでは、アクセス権限が AWS Elemental MediaStore に制限されたユーザーを作成する方法について説明します。

トピック

- ステップ 1: ユーザーグループを作成する (p. 6)
- ステップ 2: ユーザーを作成する (p. 6)

ステップ 1: ユーザーグループを作成する

ユーザーごとに個別のポリシーをアタッチするのではなく、AWS Elemental MediaStore ポリシーごとにユーザーグループを作成してユーザーをグループに割り当てることができます。次の手順を使用して、2 つのユーザーグループを作成します。1 つは `AWSElementalMediaStoreFullAccess` ポリシー用、もう 1 つは `AWSElementalMediaStoreReadOnly` ポリシー用です。

> Note
>
> `AWSElementalMediaStoreFullAccess` および `AWSElementalMediaStoreReadOnly` は AWS管理ポリシーです。

ユーザーグループを作成するには

1. IAM コンソールのナビゲーションペインで、[グループ]、[新しいグループの作成] の順に選択します。
2. [Groups (グループ)] ページで、[Create New Group (新しいグループの作成)] を選択し、`AWSElementalMediaStoreFullAccess` ポリシーを使用して管理者グループを作成します。
   a. [グループ名の設定] ページで、グループの名前 (「**`MediaStoreAdmins`**」など) を入力します。
   b. [次のステップ] を選択します。
   c. [Attach Policy (ポリシーのアタッチ)] ページの [Filter (フィルター)] で、[AWS Managed] を選択し、「**`mediastore`**」と入力します。
   d. ポリシーリストで[AWSElementalMediaStoreFullAccess] ポリシーを選択します。
   e. [次のステップ] を選択します。
   f. [Review] ページで、[Create Group] を選択します。
3. [Groups (グループ)] ページで、[Create New Group (新しいグループの作成)] を選択し、`AWSElementalMediaStoreReadOnly` ポリシーを使用して読み取り専用グループを作成します。
   a. [グループ名の設定] ページで、グループの名前 (「**`MediaStoreReaders`**」など) を入力します。
   b. [次のステップ] を選択します。
   c. [Attach Policy (ポリシーのアタッチ)] ページの [Filter (フィルター)] で、[AWS Managed] を選択し、「**`mediastore`**」と入力します。
   d. ポリシーリストで[AWSElementalMediaStoreFullAccess] ポリシーを選択します。
   e. [次のステップ] を選択します。
   f. [Review] ページで、[Create Group] を選択します。

ステップ 2: ユーザーを作成する

AWS Elemental MediaStore へのアクセスが必要な個人に対して IAM ユーザーを作成し、各ユーザーを適切なユーザーグループに追加することで、各ユーザーに適切なレベルのアクセス許可を確実に割り当てます。

ユーザーを作成するには

1. IAM コンソールのナビゲーションペインで、[ユーザー]、[ユーザーの追加] の順に選択します。
2. [ユーザー名] に、MediaStore へのサインインに使用する名前を入力します。
3. [AWS マネジメントコンソールへのアクセス] の横のチェックボックスを選択し、[カスタムパスワード] を選択して、新しいユーザーのパスワードをボックスに入力します。オプションとして [Require

password reset] (パスワードのリセットの強制) を選択し、ユーザーが次回サインインしたときにパスワードを作成することを強制できます。

4. [Next: Permissions (次へ: アクセス許可)] を選択します。
5. [Set permissions for user] ページで、[Add user to group] を選択します。
6. グループリストで、適切なポリシーがアタッチされたグループを選択します。アクセス許可レベルは以下のとおりです。

   - MediaStoreAdmins グループのアクセス権限では MediaStore のすべてのリソースに対するすべてのアクションを許可します。
   - MediaStoreReaders グループのアクセス権限では、MediaStore のすべてのリソースへの読み取り専用アクセスを許可します。
7. [Next: Review] を選択して、新しいユーザーに追加するグループメンバーシップのリストを表示します。
8. 続行する準備ができたら、[Create user] を選択します。

AWS Elemental MediaStore の開始方法

この「開始方法」チュートリアルでは、AWS Elemental MediaStore を使用してコンテナを作成し、オブジェクトをアップロードする方法を示します。

トピック

- ステップ 1: AWS Elemental MediaStore にアクセスする (p. 8)
- ステップ 2: コンテナを作成する (p. 8)
- ステップ 3: オブジェクトをアップロードする (p. 9)
- ステップ 4: オブジェクトにアクセスする (p. 9)

ステップ 1: AWS Elemental MediaStore にアクセスする

AWS アカウントをセットアップして IAM ユーザーおよびロールを作成したら、AWS Elemental MediaStore のコンソールにサインインします。

AWS Elemental MediaStore にアクセスするには

- AWS マネジメントコンソール にサインインし、MediaStore コンソール (https://console.aws.amazon.com/mediastore/) を開きます。

  Note

  ログインには、このアカウントで作成した IAM 認証情報のいずれかを使用できます。IAM 認証情報の作成については、「セットアップ (p. 4)」を参照してください。

ステップ 2: コンテナを作成する

AWS Elemental MediaStore のコンテナを使用してフォルダとオブジェクトを保存します。コンテナを使用して関連するオブジェクトをグループ化できます。ファイルシステムでディレクトリを使用してファイルをグループ化するのと同じ方法です。コンテナの作成時に料金は発生しません。オブジェクトをコンテナにアプロードしたときにのみ料金が発生します。

コンテナを作成するには

1. [Containers (コンテナ)] ページで、[Create container (コンテナの作成)] を選択します。
2. [Container name (コンテナ名)] に、コンテナの名前を入力します。詳細については、コンテナ名に関するルール (p. 10)を参照してください。
3. [Create container (コンテナの作成)] を選択します。AWS Elemental MediaStore のコンテナのリストに新しいコンテナが追加されます。コンテナの最初のステータスである [Creating (作成中)] が [Active (アクティブ)] に変わります。

ステップ 3: オブジェクトをアップロードする

オブジェクト (最大 25 MB/オブジェクト) をコンテナ、またはコンテナ内のフォルダにアップロードします。オブジェクトをフォルダにアップロードするには、フォルダへのパスを指定します。フォルダが既に存在する場合、AWS Elemental MediaStore はそのフォルダにオブジェクトを保存します。フォルダが存在しない場合は、フォルダが自動的に作成されて、そのフォルダにオブジェクトが保存されます。

> Note
>
> オブジェクトのファイル名には、文字、数字、ピリオド (.)、アンダースコア (_)、チルダ (~)、およびハイフン (-) のみを使用できます。

オブジェクトをアップロードするには

1. [Containers (コンテナ)] ページで、先ほど作成したコンテナの名前を選択します。コンテナの詳細ページが表示されます。
2. [Upload object (オブジェクトのアップロード)] を選択します。
3. [Target path (ターゲットのパス)] に、フォルダへのパスを入力しますたとえば、「`premium/canada`」と入力します。パスのフォルダがまだ存在していない場合は、AWS Elemental MediaStore により自動的に作成されます。
4. [オブジェクト] で、[参照] を選択します。
5. 適切なフォルダに移動し、アップロードする 1 つのオブジェクトを選択します。
6. [Open (開く)]、[Upload (アップロード)] の順に選択します。

ステップ 4: オブジェクトにアクセスする

オブジェクトを指定先のエンドポイントにダウンロードできます。

1. [Containers (コンテナ)] ページで、ダウンロードするオブジェクトが含まれているコンテナの名前を選択します。
2. ダウンロードするオブジェクトがサブフォルダ内にある場合は、オブジェクトが表示されるまで繰り返しフォルダ名を選択します。
3. オブジェクトの名前を選択します。
4. オブジェクトの詳細ページで、[Download (ダウンロード)] を選択します。

AWS Elemental MediaStore のコンテナ

MediaStore のコンテナを使用してフォルダとオブジェクトを保存します。コンテナを使用して関連するオブジェクトをグループ化できます。ファイルシステムでディレクトリを使用してファイルをグループ化するのと同じ方法です。コンテナの作成時に料金は発生しません。オブジェクトをコンテナにアプロードしたときにのみ料金が発生します。料金の詳細については、「AWS Elemental MediaStore 料金」を参照してください。

トピック

- コンテナ名に関するルール (p. 10)
- コンテナの作成 (p. 10)
- コンテナの詳細の表示 (p. 11)
- コンテナのリストの表示 (p. 12)
- コンテナの削除 (p. 12)

コンテナ名に関するルール

コンテナの名前を選択するときは、以下の点に留意してください。

- 名前は現在の AWS リージョンの現在のアカウント内で一意である必要があります。
- 名前には大文字、小文字、数字、およびアンダースコア (_) を使用できます。
- 名前は 1 ~ 255 文字の長さである必要があります。
- 名前では、大文字と小文字が区別されます。たとえば、`myContainer` コンテナと `mycontainer` フォルダは一意の名前であるため、一緒に使用できます。
- コンテナを作成した後に名前を変更することはできません。

コンテナの作成

AWS アカウントごとに最大 100 個のコンテナを作成できます。コンテナ内で 10 レベルを超えて入れ子にしない限り、無制限の数のフォルダを作成できます。さらに、各コンテナに必要な数のオブジェクトをアップロードできます。

> Tip
>
> AWS CloudFormationテンプレートを使用して自動的にコンテナを作成することもできます。AWS CloudFormation テンプレートは 5 つの API アクションのデータを管理し、コンテナの作成、アクセスのログ記録の設定を追加、デフォルトのコンテナポリシーの更新、Cross-Origin Resource Sharing (CORS) ポリシーの追加、およびライフサイクルポリシーオブジェクトを追加します。詳細については、「AWS CloudFormation ユーザーガイド」を参照してください。

コンテナを作成するには (コンソール)

1. MediaStore コンソール (https://console.aws.amazon.com/mediastore/) を開きます。
2. [Containers (コンテナ)] ページで、[Create container (コンテナの作成)] を選択します。
3. [Container name (コンテナ名)] に、コンテナの名前を入力します。詳細については、「コンテナ名に関するルール (p. 10)」を参照してください。

4. [Create container (コンテナの作成)] を選択します。AWS Elemental MediaStore のコンテナのリストに新しいコンテナが追加されます。コンテナの最初のステータスである [Creating (作成中)] が [Active (アクティブ)] に変わります。

コンテナを作成するには (AWS CLI)

- AWS CLI で、`create-container` コマンドを使用します。

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

戻り値の例を以下に示しています。

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

コンテナの詳細の表示

コンテナの詳細には、コンテナポリシー、エンドポイント、ARN、作成日時などが含まれます。

コンテナの詳細を表示するには (コンソール)

1. MediaStore コンソール (https://console.aws.amazon.com/mediastore/) を開きます。
2. [Containers (コンテナ)] ページで、コンテナの名前を選択します。

   コンテナの詳細ページが表示されます。このページは 2 つのセクションに分かれています。

   - [オブジェクト] セクションには、コンテナ内のオブジェクトとフォルダが一覧表示されます。
   - [Container policy (コンテナポリシー)] セクションには、このコンテナに関連付けられているリソースベースのポリシーが表示されます。リソースポリシーの詳細については、「コンテナポリシー (p. 14)」を参照してください。

コンテナの詳細を表示するには (AWS CLI)

- AWS CLI で、`describe-container` コマンドを使用します。

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

戻り値の例を以下に示しています。

```
{
    "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"
```

```
    }
}
```

コンテナのリストの表示

アカウントに関連付けられているすべてのコンテナのリストを表示できます。

コンテナのリストを表示するには (コンソール)

- MediaStore コンソール (https://console.aws.amazon.com/mediastore/) を開きます。

  [Containers (コンテナ)] ページに、アカウントに関連付けられているすべてのコンテナが一覧表示されます。

コンテナのリストを表示するには (AWS CLI)

- AWS CLI で、`list-containers` コマンドを使用します。

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

戻り値の例を以下に示しています。

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

コンテナの削除

コンテナにオブジェクトが含まれていない場合に限り、コンテナを削除できます。

コンテナを削除するには (コンソール)

1. MediaStore コンソール (https://console.aws.amazon.com/mediastore/) を開きます。

2. [Containers (コンテナ)] ページで、コンテナ名の左にあるオプションを選択します。
3. [削除] を選択します。

コンテナを削除するには (AWS CLI)

- AWS CLI で、`delete-container` コマンドを使用します。

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

このコマンドの戻り値はありません。

AWS Elemental MediaStore のポリシー

以下のポリシーを 1 つ以上 AWS Elemental MediaStore コンテナに適用できます。

- コンテナポリシー (p. 14) - コンテナ内のすべてのフォルダとオブジェクトへのアクセス権を設定します。MediaStore は、ユーザーがコンテナですべての MediaStore オペレーションを実行できるようにするデフォルトのポリシーを設定します。このポリシーは、すべてのオペレーションを HTTPS で実行する必要があることを指定します。コンテナを作成すると、コンテナポリシーを編集できます。
- クロスオリジンリソース共有 (CORS) ポリシー (p. 22) - 特定のドメインのクライアントウェブアプリケーションが別のドメインのリソースと通信できます。MediaStore はデフォルトの CORS ポリシーを設定しません。
- メトリクスポリシー (p. 38) - MediaStore がメトリクスを Amazon CloudWatch に送信できるようにします。MediaStore はデフォルトのメトリクスポリシーを設定しません。
- オブジェクトライフサイクルポリシー (p. 27) - オブジェクトが MediaStore コンテナに残る期間を制御します。MediaStore は、デフォルトのオブジェクトライフサイクルポリシーを設定しません。

AWS Elemental MediaStore のコンテナポリシー

コンテナごとにリソースベースのポリシーが割り当てられています。このポリシーで、コンテナ内のすべてのフォルダとオブジェクトへのアクセス権限を管理します。すべての新しいコンテナに自動的に割り当てられるデフォルトポリシーでは、コンテナに対するすべての AWS Elemental MediaStore オペレーションへのアクセスが許可されます。このアクセスの条件として、オペレーションでは HTTPS を使用する必要があります。コンテナを作成したら、そのコンテナにアタッチされているポリシーを編集することができます。

また、オブジェクトのライフサイクルポリシー (p. 27)を指定して、コンテナ内のオブジェクトの有効期限を管理することもできます。指定したオブジェクトが有効期限に達すると、コンテナからオブジェクトが自動的に削除されます。

トピック

- コンテナポリシーを表示する (p. 14)
- コンテナポリシーを編集する (p. 15)
- コンテナポリシーの例 (p. 16)

コンテナポリシーを表示する

コンソールまたは AWS CLI を使用して、コンテナのリソースベースのポリシーを表示できます。

コンテナポリシーを表示するには (コンソール)

1. MediaStore コンソール (https://console.aws.amazon.com/mediastore/) を開きます。
2. [Containers (コンテナ)] ページで、コンテナの名前を選択します。

コンテナの詳細ページが表示されます。ポリシーは [Container policy (コンテナポリシー)] セクションに表示されます。

コンテナポリシーを表示するには (AWS CLI)

- AWS CLI で、`get-container-policy` コマンドを使用します。

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

戻り値の例を以下に示します。

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

コンテナポリシーを編集する

デフォルトのコンテナポリシーへのアクセス許可を編集するか、新しいポリシーを作成してデフォルトのポリシーと置き換えることができます。新しいポリシーが有効になるまで、最大で 5 分かかります。

コンテナポリシーを編集するには (コンソール)

1. MediaStore コンソール (https://console.aws.amazon.com/mediastore/) を開きます。
2. [Containers (コンテナ)] ページで、コンテナの名前を選択します。
3. [ポリシーの編集] を選択します。異なるアクセス許可を設定する方法の例については、「the section called “コンテナポリシーの例” (p. 16)」を参照してください。
4. 適切な変更を行い、[Save (保存)] を選択します。

コンテナポリシーを編集するには (AWS CLI)

1. コンテナポリシーを定義するファイルを作成します。

```
{
```

```
  "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. AWS CLI で、put-container-policy コマンドを使用します。

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

このコマンドの戻り値はありません。

コンテナポリシーの例

さまざまなユーザーグループ向けに作成されたコンテナポリシーの例を以下に示します。

トピック

- コンテナポリシーの例: デフォルト値 (p. 16)
- コンテナポリシーの例: HTTPS 経由のパブリック読み取りアクセス (p. 17)
- コンテナポリシーの例: HTTP または HTTPS 経由のパブリック読み取りアクセス (p. 17)
- コンテナポリシーの例: クロスアカウント読み取りアクセス (HTTP 対応) (p. 18)
- コンテナポリシーの例: HTTPS 経由のクロスアカウント読み取りアクセス (p. 18)
- コンテナポリシーの例: ロールへのクロスアカウント読み取りアクセス (p. 19)
- コンテナポリシーの例: ロールへのクロスアカウントフルアクセス (p. 19)
- コンテナポリシーの例: フォルダに対する AWS のサービスへのアクセスのポスト (p. 20)
- コンテナポリシーの例: 複数のフォルダに対する AWS のサービスへのアクセスのポスト (p. 21)
- コンテナポリシーの例: 特定の IP アドレスに制限されたアクセス (p. 21)

コンテナポリシーの例: デフォルト値

コンテナを作成すると、AWS Elemental MediaStore は自動的に以下のリソースベースのポリシーをアタッチします。

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

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

ポリシーはサービスに組み込まれているため、作成する必要はありません。ただし、デフォルトポリシーのアクセス許可が、コンテナに使用するアクセス許可と一致しない場合は、コンテナのポリシーを編集できます。 (p. 15)

すべての新しいコンテナに割り当てられるデフォルトポリシーでは、コンテナに対するすべての MediaStore オペレーションへのアクセスが許可されます。このアクセスの条件として、オペレーションでは HTTPS を使用する必要があります。

コンテナポリシーの例: HTTPS 経由のパブリック読み取りアクセス

このポリシー例では、ユーザーが HTTPS リクエストを行ってオブジェクトを取得することができます。これにより、認証済みユーザーや匿名ユーザー (ログインしていないユーザー) など、すべてのユーザーに安全な SSL/TLS 接続を経由した読み取りアクセスが許可されます。ステートメント名は ですPublicReadOverHttps。 任意のオブジェクト (リソースパスの末尾に * を使用して指定) に対する GetObjectおよび DescribeObjectオペレーションへのアクセスが許可されます。このアクセスの条件として、オペレーションでは HTTPS を使用する必要があります。

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

コンテナポリシーの例: HTTP または HTTPS 経由のパブリック読み取りアクセス

このポリシー例では、任意のオブジェクト (リソースパスの末尾に「*」を使用して指定) に対する GetObject オペレーションと DescribeObject オペレーションへのアクセスが許可されます。これにより、すべての認証済みユーザーや匿名ユーザー (ログインしていないユーザー) など、すべてのユーザーに読み取りアクセスが許可されます。

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

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

コンテナポリシーの例: クロスアカウント読み取りアクセス (HTTP 対応)

このポリシー例では、ユーザーが HTTP リクエストを行ってオブジェクトを取得することができます。このアクセスは、クロスアカウントアクセス権限を持つ認証済みユーザーに許可されます。オブジェクトは、SSL/TLS 証明書を持つサーバーでホストされている必要はありません。

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

コンテナポリシーの例: HTTPS 経由のクロスアカウント読み取りアクセス

このポリシー例では、指定した <other acct number> の ルートユーザー ユーザーによって所有されている任意のオブジェクト (リソースパスの末尾に * で指定) に対する `GetObject` オペレーションと `DescribeObject` オペレーションへのアクセスを許可します。このアクセスの条件として、オペレーションでは HTTPS を使用する必要があります。

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

```
      "Condition": {
        "Bool": {
            "aws:SecureTransport": "true"
        }
      }
    }
  ]
}
```

コンテナポリシーの例: ロールへのクロスアカウント読み取りアクセス

このポリシー例では、<owner acct number> によって所有されている任意のオブジェクト (リソースパスの末尾に * で指定) に対する `GetObject` オペレーションと `DescribeObject` オペレーションへのアクセスを許可します。このアクセスは、他のアカウント <other acct number> が <role name> に指定されたロールを引き受けた場合、そのアカウントのすべてのユーザーに許可されます。

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

コンテナポリシーの例: ロールへのクロスアカウントフルアクセス

このポリシー例では、ユーザーが HTTP 経由でログインしていることを条件として、ユーザーにアカウント内のすべてのオブジェクトを更新するクロスアカウントアクセスが許可されます。また、クロスアカウントアクセスでは、指定されたロールを引き受けたアカウントへ HTTP または HTTPS 経由でのオブジェクトの削除、ダウンロード、紹介が許可されます。

- 最初のステートメントは ですCrossAccountRolePostOverHttps。 これにより、任意のオブジェクトに対するPutObjectオペレーションへのアクセスが許可されます<role name>。また、このアクセスは、指定したアカウントが に指定されたロールを引き受けた場合、そのアカウントのすべてのユーザーに許可されます。このアクセスの条件として、オペレーションでは HTTPS を使用する必要があります (この条件は、`PutObject` へのアクセスを提供する場合に必ず含める必要があります)。

  つまり、クロスアカウントアクセスを持つすべてのプリンシパルが `PutObject` にアクセスできますが、アクセス方法は HTTPS 経由に限られます。
- 2 番目のステートメントは ですCrossAccountFullAccessExceptPost。 すべてのオブジェクトを除くすべての オペレーションへのアクセスが許可されます。PutObjectこのアクセスは、指定したアカウントが <role name> に指定されたロールを引き受けた場合、そのアカウントのすべてのユーザーに許可されます。このアクセスでは、オペレーションで HTTPS を使用することが条件として要求されません。

  つまり、クロスアカウントアクセス権限を持つすべてのアカウントが `DeleteObject`、`GetObject` など (ただし、`PutObject` を除く) にアクセスできます。また、アクセスには HTTP または HTTPS を使用できます。

2 番目のステートメントから PutObject を除外しない場合、ステートメントを有効になりません (PutObject を含める場合は、HTTPS を条件として明示的に設定する必要があるためです)。

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

コンテナポリシーの例: フォルダに対する AWS のサービスへのアクセスのポスト

このポリシーでは、AWS Elemental MediaStore のオブジェクトをポストすることを AWS の別のサービスに許可します。任意のオブジェクトに対する PutObject へのアクセスを許可します。AWS の特定サービスに対してこのアクセスが許可されます。このアクセスの条件として、オペレーションでは HTTPS を使用する必要があります (この条件は、PutObject へのアクセスを提供する場合に必ず含める必要があります)。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "MediaStorePostToSpecificPath",
      "Effect": "Allow",
      "Action": "mediastore:PutObject",
      "Principal":{
        "AWS": "<aws service principal>"},
      "Resource": "arn:aws:mediastore:<region>:<owner acct number>:container/<container
 name>/<specific path>/*",
      "Condition": {
        "Bool": {
            "aws:SecureTransport": "true"
        }
      }
    }
  ]
```

```
}
```

コンテナポリシーの例: 複数のフォルダに対する AWS のサービスへのアクセスのポスト

このポリシーは、 2 つの異なるパスへのアクセスの設定方法を示す MediaStorePostToSpecificPath のバリエーションです。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "MediaStorePostToSeveralPaths",
      "Effect": "Allow",
      "Action": "mediastore:PutObject",
      "Principal":{
        "AWS": "<aws service principal>"},
      "Resource": [
        "arn:aws:mediastore:<region>:<owner acct number>:container/<container
 name>/<specific path 1>/*",
        "arn:aws:mediastore:<region>:<owner acct number>:container/<container
 name>/<specific path 2>/*",
      ],
      "Condition": {
        "Bool": {
            "aws:SecureTransport": "true"
        }
      }
    }
  ]
}
```

コンテナポリシーの例: 特定の IP アドレスに制限されたアクセス

このポリシーの例では、指定されたコンテナ内のオブジェクトに対するすべての AWS Elemental MediaStore オペレーションへのアクセスが許可されます。ただし、リクエストは条件で指定された IP アドレス範囲からのリクエストである必要があります。

このステートメントの条件では、198.51.100 を特定します。* の範囲のインターネットプロトコルバージョン 4 (IPv4) IP アドレスが許可されています。ただし、次の例外があります。198.51.100.188.

Condition ブロックでは、IpAddress および NotIpAddress 条件と、aws:SourceIp 条件キーが使用されています。これは AWS 全体を対象とする条件キーです。aws:sourceIpIPv4値には標準の CIDR 表記が使用されます。詳細については、IAM ユーザーガイド の「IP アドレス条件演算子」を参照してください。

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

```
            "Condition": {
                "IpAddress": {
                    "aws:SourceIp": [
                        "192.0.2.0/24",
                        "203.0.113.0/24"
                    ]
                },
                "NotIpAddress": {
                    "aws:SourceIp": "198.51.100.0/24"
                }
            }
        }
    ]
}
```

AWS Elemental MediaStore のクロスオリジンリソース共有 (CORS) ポリシー

クロスオリジンリソース共有 (CORS) は、特定のドメインにロードされたクライアントウェブアプリケーションが異なるドメイン内のリソースと通信する方法を定義します。AWS Elemental MediaStore での CORS のサポートにより、MediaStore で機能豊富なクライアント側ウェブアプリケーションをビルドし、MediaStore リソースに対するクロスオリジンアクセスを選択的に許可できます。

> **Note**
>
> CORS ポリシーのあるコンテナからコンテンツを配信するために Amazon CloudFront を使用している場合、必ず AWS Elemental MediaStore の配信を設定 (CORS をセットアップするためにキャッシュ動作を編集するステップを含む) するようにします。

このセクションでは、CORS の概要を示します。サブトピックでは、 AWS Elemental MediaStore コンソールを使用するか、 MediaStore REST API と AWS を使用してプログラムで CORS SDKs を有効にする方法について説明します。

トピック

- CORS ユースケースのシナリオ (p. 22)
- コンテナへの CORS ポリシーの追加 (p. 23)
- CORS ポリシーの表示 (p. 24)
- CORS ポリシーの編集 (p. 24)
- CORS ポリシーの削除 (p. 25)
- CORS の問題のトラブルシューティング (p. 26)
- CORS ポリシーの例 (p. 26)

CORS ユースケースのシナリオ

CORS のユースケースの例を以下に示します。

- シナリオ 1: という AWS Elemental MediaStore コンテナでライブストリーミング動画を配信するとLiveVideoします。 ユーザーは、 などの特定のオリジンhttp://livevideo.mediastore.ap-southeast-2.amazonaws.comから動画マニフェストエンドポイントをロードwww.example.comします。 認証されていないJavaScriptリクエストGETおよびリクエストを介してこのコンテナから発信される動画にアクセスするには、PUTビデオプレーヤーを使用します。通常、ブラウザはこれらのリクエ

ストを許可JavaScriptしないようにブロックしますが、 からのリクエストを明示的に有効にするようにコンテナに CORS ポリシーを設定できますwww.example.com。

- シナリオ 2: シナリオ 1 と同じライブストリームをMediaStoreコンテナでホストするとします。ただし、任意のオリジンからのリクエストを許可します。ワイルドカード (*) のオリジンを許可し、任意のオリジンのリクエストから動画にアクセスできるように、CORS ポリシーを設定できます。

コンテナへの CORS ポリシーの追加

このセクションでは、クロスオリジンリソース共有 (CORS) 設定を AWS Elemental MediaStore コンテナに追加する方法について説明します。CORS は、特定のドメインにロードされたクライアントウェブアプリケーションが別のドメイン内のリソースと通信できるようにします。

クロスオリジンリクエストを許可するようにコンテナを設定するには、CORS ポリシーをコンテナに追加します。CORS ポリシーでは、コンテナへのアクセスを許可するオリジン、各オリジンでサポートされるオペレーション (HTTP メソッド)、その他のオペレーション固有の情報を識別するルールを定義します。

CORS ポリシーをコンテナに追加すると、コンテナポリシー (p. 14)が (コンテナへのアクセス権限を管理するために) 継続的に適用されます。

CORS ポリシーを更新するには (コンソール)

1. MediaStore コンソール (https://console.aws.amazon.com/mediastore/) を開きます。
2. [Containers (コンテナ)] ページで、CORS ポリシーを作成する対象のコンテナの名前を選択します。

   コンテナの詳細ページが表示されます。
3. [Container CORS policy (コンテナの CORS ポリシー)] セクションで、[Create CORS policy (CORS ポリシーの作成)] を選択します。
4. ポリシーを JSON 形式で挿入し、[Save (保存)] を選択します。

CORS ポリシーを追加するには (AWS CLI)

1. CORS ポリシーを定義するファイルを作成します。

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

2. AWS CLI で、put-cors-policy コマンドを使用します。

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

   このコマンドの戻り値はありません。

CORS ポリシーの表示

クロスオリジンリソース共有 (CORS) は、特定のドメインにロードされたクライアントウェブアプリケーションが異なるドメイン内のリソースと通信する方法を定義します。

CORS ポリシーを表示するには (コンソール)

1. MediaStore コンソール (https://console.aws.amazon.com/mediastore/) を開きます。
2. [Containers (コンテナ)] ページで、CORS ポリシーを表示する対象のコンテナの名前を選択します。

   コンテナの詳細ページの [Container CORS policy (コンテナの CORS ポリシー)] セクションに CORS ポリシーが表示されます。

CORS ポリシーを表示するには (AWS CLI)

- AWS CLI で、`get-cors-policy` コマンドを使用します。

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

戻り値の例を以下に示します。

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

CORS ポリシーの編集

クロスオリジンリソース共有 (CORS) は、特定のドメインにロードされたクライアントウェブアプリケーションが異なるドメイン内のリソースと通信する方法を定義します。

CORS ポリシーを編集するには (コンソール)

1. MediaStore コンソール (https://console.aws.amazon.com/mediastore/) を開きます。
2. [Containers (コンテナ)] ページで、CORS ポリシーを編集する対象のコンテナの名前を選択します。

   コンテナの詳細ページが表示されます。
3. [Container CORS policy (コンテナの CORS ポリシー)] セクションで、[Edit CORS policy (CORS ポリシーの編集)] を選択します。
4. ポリシーを変更し、[Save (保存)] を選択します。

CORS ポリシーを編集するには (AWS CLI)

1. 更新された CORS ポリシーを定義するファイルを作成します。

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

2. AWS CLI で、`put-cors-policy` コマンドを使用します。

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

このコマンドの戻り値はありません。

CORS ポリシーの削除

クロスオリジンリソース共有 (CORS) は、特定のドメインにロードされたクライアントウェブアプリケーションが異なるドメイン内のリソースと通信する方法を定義します。コンテナから CORS ポリシーを削除すると、クロスオリジンリクエストのアクセス許可が削除されます。

CORS ポリシーを削除するには (コンソール)

1. MediaStore コンソール (https://console.aws.amazon.com/mediastore/) を開きます。
2. [Containers (コンテナ)] ページで、CORS ポリシーを削除する対象のコンテナの名前を選択します。

   コンテナの詳細ページが表示されます。
3. [Container CORS policy (コンテナの CORS ポリシー)] セクションで、[Delete CORS policy (CORS ポリシーの削除)] を選択します。
4. [Continue] を選択して確認し、[Save] を選択します。

CORS ポリシーを削除するには (AWS CLI)

- AWS CLI で、`delete-cors-policy` コマンドを使用します。

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

このコマンドの戻り値はありません。

CORS の問題のトラブルシューティング

CORS ポリシーが存在するコンテナにアクセスするときに予期しない動作が発生した場合は、以下のステップに従って問題のトラブルシューティングを行います。

1. CORS ポリシーがコンテナにアタッチされていることを確認します。

   手順については、「the section called “CORS ポリシーの表示” (p. 24)」を参照してください。
2. 任意のツール (ブラウザの開発者コンソールなど) を使用して、完全なリクエストとレスポンスをキャプチャします。コンテナにアタッチされている CORS ポリシーに、リクエストのデータと一致する少なくとも 1 つの CORS ルールが含まれていることを確認します。

   a. リクエストに `Origin` ヘッダーがあることを確認します。

      ヘッダーがないリクエストMediaStoreは、AWS Elemental でクロスオリジンリクエストとして扱われず、CORS レスポンスヘッダーをレスポンスで返送しません。

   b. リクエストの `Origin` ヘッダーが、特定の `AllowedOrigins` の `CORSRule` 要素の少なくとも 1 つと一致していることを確認します。

      `Origin`リクエストヘッダーのスキーム、ホスト、およびポートの値は、 `AllowedOrigins`の と一致する必要がありま`CORSRule`す。 たとえば、 オリジンを許可する`CORSRule`ように `http://www.example.com` を設定した場合、リクエスト内の `https://www.example.com`と`http://www.example.com:80`オリジンの両方が、設定で許可されているオリジンと一致しません。

   c. リクエストのメソッド (またはプリフライトリクエストの場合は `Access-Control-Request-Method` に指定されたメソッド) が、同じ `AllowedMethods` の `CORSRule` 要素の 1 つであることを確認します。

   d. プリフライトリクエストの場合、それに `Access-Control-Request-Headers` ヘッダーが含まれているときに、`CORSRule` に `AllowedHeaders` ヘッダーのすべての値に対する `Access-Control-Request-Headers` エントリが含まれていること。

CORS ポリシーの例

クロスオリジンリソース共有 (CORS) ポリシーの例を以下に示します。

トピック

- CORS ポリシーの例: 任意のドメインのための読み取りアクセス (p. 26)
- CORS ポリシーの例: 特定のドメインへの読み取りアクセス (p. 27)

CORS ポリシーの例: 任意のドメインのための読み取りアクセス

次のポリシーでは、AWS Elemental MediaStore コンテナからコンテンツを取得することをすべてのドメインのウェブページに許可します。リクエストには、送信元ドメインからのすべての HTTP ヘッダーが含まれます。このサービスでは、送信元ドメインからの HTTP GET リクエストと HTTP HEAD リクエストにのみ応答します。結果は、新しい結果セットが配信されるまで 3,000 秒間キャッシュされます。

```
[
  {
    "AllowedHeaders": [
      "*"
    ],
    "AllowedMethods": [
      "GET",
      "HEAD"
    ],
```

```
    "AllowedOrigins": [
      "*"
    ],
    "MaxAgeSeconds": 3000
  }
]
```

CORS ポリシーの例: 特定のドメインへの読み取りアクセス

以下のポリシーでは、AWS Elemental MediaStore コンテナからコンテンツを取得することを `https://www.example.com` のウェブページに許可します。リクエストには `https://www.example.com` からのすべての HTTP ヘッダーが含まれ、サービスは からの HTTP GET リクエストと HTTP HEAD リクエストにのみ応答`https://www.example.com`します。 結果は、新しい結果セットが配信されるまで 3,000 秒間キャッシュされます。

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

AWS Elemental MediaStore のオブジェクトのライフサイクルポリシー

オブジェクトのコンテナ内保存期間を管理する、オブジェクトのライフサイクルポリシーをコンテナごとに作成することができます。オブジェクトが指定された有効期限に達すると、AWS Elemental MediaStore によりオブジェクトが削除されます。不要になったオブジェクトを削除して、ストレージコストを削減することもできます。

オブジェクトが特定の経過時間に達した後、MediaStore がオブジェクトを低頻度アクセス (IA) ストレージクラスに移動するように指定することもできます。IA ストレージクラスに保存されているオブジェクトは、標準のストレージクラスに保存されているオブジェクトとはストレージおよび取得の速度が異なります。詳細については、「MediaStore 料金表」を参照してください。

オブジェクトのライフサイクルポリシーには、サブフォルダごとにオブジェクトの保持期間を決定するというルールが含まれています (オブジェクトのライフサイクルポリシーをオブジェクトに個別に割り当てることはできません)。1 つのコンテナに対してオブジェクトのライフサイクルポリシーを 1 つ割り当てることができますが、オブジェクトのライフサイクルポリシーごとに最大 10 個のルールを追加できます。詳細については、オブジェクトのライフサイクルポリシーのコンポーネント (p. 28) を参照してください。

トピック

- オブジェクトのライフサイクルポリシーのコンポーネント (p. 28)
- コンテナにオブジェクトのライフサイクルポリシーを追加する (p. 32)
- オブジェクトのライフサイクルポリシーを表示する (p. 33)

- オブジェクトのライフサイクルポリシーを編集する (p. 34)
- オブジェクトのライフサイクルポリシーを削除する (p. 35)
- オブジェクトのライフサイクルポリシーの例 (p. 35)

オブジェクトのライフサイクルポリシーのコンポーネント

オブジェクトのライフサイクルポリシーでは、AWS Elemental MediaStore コンテナにオブジェクトを保持する期間を管理します。各オブジェクトのライフサイクルポリシーには 1 つ以上のルールがあり、オブジェクトの保持期間を決定します。ルールは、1 つのフォルダ、複数フォルダ、コンテナ全体に適用されます。

1 つのコンテナに対して 1 つのオブジェクトのライフサイクルポリシーをアタッチすることができ、各オブジェクトのライフサイクルポリシーには最大 10 個のルールを含めることができます。オブジェクトのライフサイクルポリシーを個々のオブジェクトに割り当てることはできません。

オブジェクトのライフサイクルポリシーのルール

次の 3 種類のルールを作成できます。

- 一時データ (p. 28)
- DELETE Object (p. 29)
- ライフサイクル移行 (p. 30)

一時データ

一時的なデータルールで、数秒以内に有効期限が切れるようにオブジェクトを設定します。このタイプのルールは、ポリシーが有効になった後にコンテナに追加されたオブジェクトにのみ適用されます。MediaStore がコンテナに新しいポリシーを適用するまでに、最大 20 分かかります。

一時データのルールの例は次のようになります。

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

一時データのルールには以下の 3 つの部分があります。

- `path`: 常に に設定されます`wildcard`。 このパートを使用して、削除するオブジェクトを定義します。1 つ以上のワイルドカードを使用できます。これはアスタリスク (*) で表されます。各ワイルドカードは、0 個以上の文字の任意の組み合わせを表します。たとえば、`"path": [ {"wildcard": "Football/index*.m3u8"} ]`, は、`Football` フォルダのパターン `index*.m3u8` (例: index.m3u8、index1.m3us8、および index123456.m3u8) に一致するすべてのファイルに適用されます。1 つのルールに最大 10 個の パスを含めることができます。
- `seconds_since_create`: 常に に設定されます`numeric`。 1～300秒の値を指定できます。演算子は、より大きい (>) または以上 (>=) に設定することもできます。
- `action`: 常に に設定されます`EXPIRE`。

一時データルールの場合 (オブジェクトは数秒以内に期限切れになる)、オブジェクトの期限切れとオブジェクトの削除の間に遅延はありません。

> Note
>
> 一時データルールの対象となるオブジェクトは、`list-items` レスポンスには含まれていません。

DELETE Object

オブジェクト削除ルールは、オブジェクトが数日以内に期限切れになるように設定します。このタイプのルールは、ポリシーが作成される前にオブジェクトがコンテナに追加された場合でも、コンテナ内のすべてのオブジェクトに適用されます。MediaStore がコンテナに新しいポリシーを適用するまでに最大 20 分かかりますが、オブジェクトがコンテナからクリアされるまでには最大 24 時間かかる場合があります。

オブジェクトの削除に関する 2 つのルールの例は以下のようになります。

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

オブジェクトの削除ルールには以下の 3 つの部分があります。

- `path`: `prefix`または に設定します`wildcard`。 同じルール`prefix`で `wildcard` と を組み合わせることはできません。両方を使用する場合は、上記の例に示すように、`prefix` に 1 つのルールを作成し、`wildcard` に別のルールを作成する必要があります。
  - `prefix` - 特定のフォルダ内のすべてのオブジェクトを削除する場合は、パスを `prefix` に設定します。パラメータが空 (`"path": [ { "prefix": "" } ],`) の場合は、現在のコンテナ内に保存されているすべてのオブジェクトがターゲットになります。1 つのルールに最大 10 個の `prefix` パスを含めることができます。
  - `wildcard` - ファイル名やファイルタイプに基づいて特定のオブジェクトを削除する場合は、パスを `wildcard` に設定します。1 つ以上のワイルドカードを使用できます。これはアスタリスク (*) で表されます。各ワイルドカードは、0 個以上の文字の任意の組み合わせを表します。たとえば、`"path": [ {"wildcard": "Football/*.ts"} ],` は、`Football` フォルダ内で `*.ts` のパターンに一致するすべてのファイル (filename.ts、filename1.ts、filename123456.ts など) に適用されます。1 つのルールに最大 10 個の `wildcard` パスを含めることができます。
- `days_since_create`: 常に に設定されます`numeric`。 1 ~ 36,500の値を指定できます。演算子は、より大きい (>) または以上 (>=) に設定することもできます。
- `action`: 常に に設定されます`EXPIRE`。

オブジェクト削除ルールの場合 (オブジェクトは数日以内に期限切れになる)、オブジェクトの期限切れとオブジェクトの削除の間にはわずかな遅延が生じます。ただし、オブジェクトの有効期限が切れるとすぐに請求が変更されます。たとえば、ライフサイクルルールで 10 `days_since_create` を指定した場合、

オブジェクトがまだ削除されていない場合でも、オブジェクトの作成から 10 日が経過すると、アカウントでそのオブジェクトには課金されなくなります。

ライフサイクル移行

ライフサイクル移行ルールは、オブジェクトが一定の期間 (日単位) に達した後に低頻度アクセス (IA) ストレージクラスに移動されるよう設定します。IA ストレージクラスに保存されているオブジェクトは、標準のストレージクラスに保存されているオブジェクトとはストレージおよび取得の速度が異なります。詳細については、「MediaStore 料金表」を参照してください。

オブジェクトを IA ストレージクラスに移動すると、標準のストレージクラスに戻すことはできません。

ライフサイクル移行ルールは、ポリシーの作成前にコンテナに追加されていたオブジェクトも含め、コンテナ内のすべてのオブジェクトに適用されます。MediaStore がコンテナに新しいポリシーを適用するまでに最大 20 分かかりますが、オブジェクトがコンテナからクリアされるまでには最大 24 時間かかる場合があります。

ライフサイクル移行ルールの例は次のようになります。

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

ライフサイクル移行ルールには、以下の 3 つの部分があります。

- path: prefixまたは に設定しますwildcard。 同じルールprefixで wildcard と を組み合わせることはできません。両方を使用する場合は、prefix に 1 つのルールを作成し、wildcard には別のルールを作成する必要があります。
  - prefix - 特定のフォルダ内のすべてのオブジェクトを IA ストレージクラスに移行する場合は、パスを prefix に設定します。パラメータが空 ("path": [{ "prefix": "" }],) の場合は、現在のコンテナ内に保存されているすべてのオブジェクトがターゲットになります。1 つのルールに最大 10 個の prefix パスを含めることができます。
  - wildcard - ファイル名やファイルタイプに基づいて IA ストレージクラスに特定のオブジェクトを移行する場合は、パスを wildcard に設定します。1 つ以上のワイルドカードを使用できます。これはアスタリスク (*) で表されます。各ワイルドカードは、0 個以上の文字の任意の組み合わせを表します。たとえば、"path": [{"wildcard": "Football/*.ts"}], は、Football フォルダ内で *.ts のパターンに一致するすべてのファイル (filename.ts、filename1.ts、filename123456.ts など) に適用されます。1 つのルールに最大 10 個の wildcard パスを含めることができます。
- days_since_create: 常に に設定されます"numeric": [">=" , 30]。
- action: 常に に設定されますARCHIVE。

Example

という名前のコンテナに 4 つのサブフォルダLiveEventsがあるとします。Football、 Baseball 、Basketball 、および AwardsShow。 LiveEventsフォルダに割り当てられたオブジェクトのライフサイクルポリシーは次のようになります。

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

前述のポリシーでは、以下を指定します。

- 最初のルールでは、 `LiveEvents/Football` フォルダと `LiveEvents/Baseball` フォルダに 28 日以上保存されているオブジェクトを削除するように AWS Elemental MediaStore に指示します。
- 2 つ目のルールでは、`LiveEvents/AwardsShow` フォルダに 15 日以上保存されているオブジェクトが削除されるようにサービスに指示します。
- 3 つ目のルールでは、`LiveEvents` コンテナ内のどこかに 40 日以上保存されているオブジェクトが削除されるように指定します。このルールは、 `LiveEvents` コンテナに直接保存されているオブジェクト、コンテナの 4 つのサブフォルダのいずれかに保存されているオブジェクトに適用されます。
- 4 番目のルールでは、`Football` フォルダ内でパターン `*.ts` に一致するオブジェクトを 20 日経過したら削除するようにサービスに指示します。
- 5 番目のルールでは、`Football` フォルダ内でパターン `index*.m3u8` に一致するオブジェクトを 15 秒経過したら削除するようにサービスに指示します。MediaStore によって、これらのファイルはコンテナに配置されてから 16 秒後に削除されます。
- 6 番目のルールは、30 日経過後に `Program` フォルダ内のオブジェクトを IA ストレージクラスに移動するようにサービスに指示します。

オブジェクトライフサイクルポリシーの例の詳細については、「オブジェクトのライフサイクルポリシーの例 (p. 35)」を参照してください。

コンテナにオブジェクトのライフサイクルポリシーを追加する

オブジェクトのライフサイクルポリシーでは、ユーザーがコンテナにオブジェクトを保存する期間を指定することができます。有効期限を設定し、有効期限が切れたら、AWS Elemental MediaStore によってオブジェクトが削除されます。サービスがコンテナに新しいポリシーを適用するまでに、最大 20 分かかります。

ライフサイクルポリシーを作成する方法については、「オブジェクトのライフサイクルポリシーのコンポーネント (p. 28)」を参照してください。

> Note
>
> オブジェクト削除ルールの場合 (オブジェクトは数日以内に期限切れになる)、オブジェクトの期限切れとオブジェクトの削除の間にはわずかな遅延が生じます。ただし、オブジェクトの有効期限が切れるとすぐに請求が変更されます。たとえば、ライフサイクルルールで 10 `days_since_create` を指定した場合、オブジェクトがまだ削除されていない場合でも、オブジェクトの作成から 10 日が経過すると、アカウントでそのオブジェクトには課金されなくなります。

オブジェクトのライフサイクルポリシーを追加するには (コンソール)

1. MediaStore コンソール (https://console.aws.amazon.com/mediastore/) を開きます。
2. [Containers (コンテナ)] ページで、オブジェクトのライフサイクルポリシーを作成する対象のコンテナの名前を選択します。

   コンテナの詳細ページが表示されます。
3. [オブジェクトのライフサイクルポリシー] セクションで、[オブジェクトのライフサイクルポリシーの作成] を選択します。
4. ポリシーを JSON 形式で挿入し、[Save (保存)] を選択します。

オブジェクトのライフサイクルポリシーを追加するには (AWS CLI)

1. オブジェクトのライフサイクルポリシーを定義するファイルを作成します。

```
{
```

```
    "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. AWS CLI で、put-lifecycle-policy コマンドを使用します。

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

このコマンドの戻り値はありません。指定されたポリシーがコンテナにアタッチされます。

オブジェクトのライフサイクルポリシーを表示する

オブジェクトのライフサイクルポリシーでは、オブジェクトをコンテナに保存する期間を指定します。

オブジェクトのライフサイクルポリシーを表示するには (コンソール)

1. MediaStore コンソール (https://console.aws.amazon.com/mediastore/) を開きます。
2. [Containers (コンテナ)] ページで、オブジェクトのライフサイクルポリシーを表示する対象のコンテナの名前を選択します。

   [オブジェクトのライフサイクルポリシー] セクションのオブジェクトのライフサイクルポリシーによって、コンテナの詳細ページが表示されます。

オブジェクトのライフサイクルポリシーを表示するには (AWS CLI)

- AWS CLI で、get-lifecycle-policy コマンドを使用します。

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

戻り値の例を以下に示します。

```
{
    "LifecyclePolicy": "{
        "rules": [
```

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

オブジェクトのライフサイクルポリシーを編集する

既存のオブジェクトのライフサイクルポリシーを編集することはできません。ただし、代わりのポリシーをアップロードして、既存のポリシーを変更することができます。サービスが、更新されたポリシーをコンテナに適用するまでに、最大 20 分かかります。

オブジェクトのライフサイクルポリシーを編集するには (コンソール)

1. MediaStore コンソール (https://console.aws.amazon.com/mediastore/) を開きます。
2. [Containers (コンテナ)] ページで、オブジェクトのライフサイクルポリシーを編集する対象のコンテナの名前を選択します。

   コンテナの詳細ページが表示されます。
3. [オブジェクトのライフサイクルポリシー] セクションで、[オブジェクトのライフサイクルポリシーの編集] を選択します。
4. ポリシーを変更し、[Save (保存)] を選択します。

オブジェクトのライフサイクルポリシーを編集するには (AWS CLI)

1. 更新するオブジェクトのライフサイクルポリシーを定義するファイルを作成します。

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

2. AWS CLI で、`put-lifecycle-policy` コマンドを使用します。

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

このコマンドの戻り値はありません。このサービスでは、指定されたポリシーを以前のポリシーと置き換えてコンテナにアタッチします。

オブジェクトのライフサイクルポリシーを削除する

オブジェクトライフサイクルポリシーを削除すると、サービスがコンテナに変更を適用するまで最大 20 分かかります。

オブジェクトのライフサイクルポリシーを削除するには (コンソール)

1. MediaStore コンソール (https://console.aws.amazon.com/mediastore/) を開きます。
2. [Containers (コンテナ)] ページで、CORS ポリシーを削除する対象のコンテナの名前を選択します。

   コンテナの詳細ページが表示されます。
3. [オブジェクトのライフサイクルポリシー] セクションで、[ライフサイクルポリシーの削除] を選択します。
4. [Continue] を選択して確認し、[Save] を選択します。

オブジェクトのライフサイクルポリシーを削除するには (AWS CLI)

- AWS CLI で、`delete-lifecycle-policy` コマンドを使用します。

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

このコマンドの戻り値はありません。

オブジェクトのライフサイクルポリシーの例

次の例は、オブジェクトのライフサイクルポリシーを示しています。

トピック

- オブジェクトのライフサイクルポリシーの例: 数秒以内に期限切れにする (p. 35)
- オブジェクトのライフサイクルポリシーの例: 数日以内に期限切れにする (p. 36)
- オブジェクトのライフサイクルポリシーの例: 低頻度アクセスストレージクラスへの移行 (p. 36)
- オブジェクトのライフサイクルポリシーの例: 複数のルール (p. 37)
- オブジェクトのライフサイクルポリシーの例: コンテナを空にする (p. 38)

オブジェクトのライフサイクルポリシーの例: 数秒以内に期限切れにする

次のポリシーでは、MediaStore が次の条件すべてに一致するオブジェクトを削除するように指定します。

- オブジェクトは、ポリシーが有効になった後にコンテナに追加されます。
- オブジェクトは、`Football` フォルダに保存されます。
- オブジェクトのファイル拡張子は `m3u8` です。

- このオブジェクトはコンテナに 20 秒以上保持されています。

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

オブジェクトのライフサイクルポリシーの例: 数日以内に期限切れにする

次のポリシーでは、MediaStore が次の条件すべてに一致するオブジェクトを削除するように指定します。

- オブジェクトは、`Program` フォルダに保存されます
- オブジェクトのファイル拡張子は `ts` です
- オブジェクトがコンテナに 5 日以上保持されています

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

オブジェクトのライフサイクルポリシーの例: 低頻度アクセスストレージクラスへの移行

次のポリシーでは、30 日経過したときに、MediaStore がオブジェクトを低頻度アクセス (IA) ストレージクラスに移動するように指定します。IA ストレージクラスに保存されているオブジェクトは、標準のストレージクラスに保存されているオブジェクトとはストレージおよび取得の速度が異なります。

`days_since_create` フィールドは `"numeric": [">=" ,30]` に設定する必要があります。

```
{
    "rules": [
        {
```

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

オブジェクトのライフサイクルポリシーの例: 複数のルール

次のポリシーは、MediaStore が次のことを行うことを指定します。

- AwardsShow フォルダに保存されているオブジェクトを、30 日後に低頻度アクセス (IA) ストレージクラスに移動します
- ファイル拡張子が m3u8 で、Football フォルダに保存されて 20 秒経過したオブジェクトを削除します
- April フォルダに保存されて 10 日間経過したオブジェクトを削除します
- ファイル拡張子が ts で、Program フォルダに保存されて 5 日間経過したオブジェクトを削除します

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

オブジェクトのライフサイクルポリシーの例: コンテナを空にする

次のオブジェクトライフサイクルポリシーは、コンテナに追加されてから 1 日後に、MediaStore がコンテナ内のすべてのオブジェクト (フォルダやサブフォルダを含む) を削除するように指定します。このポリシーが適用される前にコンテナにオブジェクトが保持されている場合、 はポリシーが有効になってから 1 日後にオブジェクトMediaStoreを削除します。サービスがコンテナに新しいポリシーを適用するまでに、最大 20 分かかります。

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

AWS Elemental MediaStore のメトリクスポリシー

コンテナごとに、メトリクスポリシーを追加して、AWS Elemental MediaStore がメトリクスを Amazon CloudWatch に送信できるようにすることができます。新しいポリシーが有効になるまで、最大 20 分かかります。各 MediaStore メトリクスの説明については、「MediaStore のメトリクス (p. 84)」を参照してください。

メトリクスポリシーには、次のものが含まれます。

- コンテナレベルでメトリクスを有効または無効にする設定。
- オブジェクトレベルでメトリクスを有効にする 0 から 5 までのルールの任意の場所。ポリシーにルールが含まれている場合は、各ルールに次の両方を含める必要があります。
  - グループに含めるオブジェクトを定義するオブジェクトグループ。定義にはパスまたはファイル名を使用できますが、900 文字を超えることはできません。有効な文字は、a ~ z、A ~ Z、0 ~ 9、_ (アンダースコア)、= (等しい)、: (コロン)、. (ピリオド)、- (ハイフン)、~ (チルダ)、/ (スラッシュ)、* (アスタリスク) です。ワイルドカード (*) も使用できます。
  - オブジェクトグループを参照できるオブジェクトグループ名。名前は 30 文字を超えることはできません。有効な文字は、a ~ z、A ~ Z、0 ~ 9、_ (下線) です。

オブジェクトが複数のルールに一致する場合、CloudWatch は一致するルールごとにデータポイントを表示します。たとえば、オブジェクトが `rule1` と `rule2` いう名前の 2 つのルールに一致する場合、CloudWatch はこれらのルールの 2 つのデータポイントを表示します。最初のルールには、`ObjectGroupName=rule1` のディメンションがあり、2 番目のルールは `ObjectGroupName=rule2` のディメンションがあります。

トピック

- メトリクスポリシーの追加 (p. 39)
- メトリクスポリシーの表示 (p. 39)
- メトリクスポリシーの編集 (p. 39)
- メトリクスポリシーの例 (p. 42)

メトリクスポリシーの追加

メトリクスポリシーには、AWS Elemental MediaStore が Amazon CloudWatch に送信するメトリクスを指定するルールが含まれています。メトリクスポリシーの例については、「メトリクスポリシーの例 (p. 42)」を参照してください。

メトリクスポリシーを追加するには (コンソール)

1. MediaStore コンソール (https://console.aws.amazon.com/mediastore/) を開きます。
2. [Containers (コンテナ)] ページで、メトリクスポリシーを追加する対象のコンテナの名前を選択します。

   コンテナの詳細ページが表示されます。
3. [メトリクスポリシー] セクションで、[Create metric policy (メトリクスポリシーの作成)] を選択します。
4. ポリシーを JSON 形式で挿入し、[Save (保存)] を選択します。

メトリクスポリシーの表示

コンソールまたは AWS CLI を使用して、コンテナのメトリクスポリシーを表示できます。

メトリクスポリシーを表示するには (コンソール)

1. MediaStore コンソール (https://console.aws.amazon.com/mediastore/) を開きます。
2. [Containers (コンテナ)] ページで、コンテナの名前を選択します。

   コンテナの詳細ページが表示されます。ポリシーが [メトリクスポリシー] セクションに表示されます。

メトリクスポリシーの編集

メトリクスポリシーには、AWS Elemental MediaStore が Amazon CloudWatch に送信するメトリクスを指定するルールが含まれています。既存のメトリクスポリシーを編集する場合、新しいポリシーが有効になるまでに最大 20 分かかります。メトリクスポリシーの例については、「

さまざまなユースケース向けに作成されたメトリクスポリシーの例を以下に示します。

トピック

- メトリクスポリシーの例: コンテナレベルのメトリクス (p. 42)

- メトリクスポリシーの例: パスレベルのメトリクス (p. 42)

- メトリクスポリシーの例: コンテナレベルおよびパスレベルのメトリクス (p. 43)
- メトリクスポリシーの例: ワイルドカードを使用したパスレベルのメトリクス (p. 43)
- メトリクスポリシーの例: ルールが重複するパスレベルのメトリクス (p. 44)

メトリクスポリシーの例: コンテナレベルのメトリクス

このサンプルポリシーは、AWS Elemental MediaStore がコンテナレベルで Amazon CloudWatch にメトリクスを送信する必要があることを示します。たとえば、これには、コンテナに対して行われた `Put` リクエストの数をカウントする `RequestCount` メトリクスが含まれます。または、これを `DISABLED` に設定することもできます。

このポリシーにはルールがないため、MediaStore はパスレベルでメトリクスを送信しません。たとえば、このコンテナ内の特定のフォルダに対して行われた `Put` リクエストの数を確認することはできません。

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

メトリクスポリシーの例: パスレベルのメトリクス

このサンプルポリシーは、AWS Elemental MediaStore がコンテナレベルで Amazon CloudWatch にメトリクスを送信しないことを示します。また、MediaStore は、`baseball/saturday` および `football/saturday` の 2 つの特定のフォルダ内のオブジェクトのメトリクスを送信する必要があります。MediaStore リクエストのメトリクスは次のとおりです。

- `baseball/saturday` フォルダへのリクエストには `ObjectGroupName=baseballGroup` の CloudWatch ディメンションが含まれます。

- `football/saturday` フォルダへのリクエストにはディメンション `ObjectGroupName=footballGroup` が含まれます。

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

- baseball/saturday フォルダへのリクエストには ObjectGroupName=baseballGroup の CloudWatch ディメンションが含まれます。

- football/saturday フォルダへのリクエストには CloudWatch ディメンション ObjectGroupName=footballGroup が含まれます。

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

メトリクスポリシーの例: ワイルドカードを使用したパスレベルのメトリクス

このサンプルポリシーは、AWS Elemental MediaStore がコンテナレベルで Amazon CloudWatch にメトリクスを送信する必要があることを示します。また、MediaStore は、ファイル名に基づいてオブジェクトのメトリクスも送信する必要があります。ワイルドカードは、オブジェクトがコンテナ内のどこにでも保存され、.m3u8 拡張子で終わる限り、任意のファイル名を持つことができることを示します。

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

メトリクスポリシーの例: ルールが重複するパスレベルのメトリクス

```
    {
      "ObjectGroup": "sports/football/saturday",
      "ObjectGroupName": "footballGroup1"
    },
    {
      "ObjectGroup": "sports/football",
      "ObjectGroupName": "footballGroup2"
    }
  ]
}
```

(p. 42)」を参照してください。

メトリクスポリシーを編集するには (コンソール)

1. MediaStore コンソール (https://console.aws.amazon.com/mediastore/) を開きます。
2. [Containers (コンテナ)] ページで、コンテナの名前を選択します。
3. [メトリクスポリシー] セクションで、[メトリクスポリシーの編集] を選択します。
4. 適切な変更を行い、[Save (保存)] を選択します。

メトリクスポリシーの例

さまざまなユースケース向けに作成されたメトリクスポリシーの例を以下に示します。

トピック

- メトリクスポリシーの例: コンテナレベルのメトリクス (p. 42)
- メトリクスポリシーの例: パスレベルのメトリクス (p. 42)
- メトリクスポリシーの例: コンテナレベルおよびパスレベルのメトリクス (p. 43)
- メトリクスポリシーの例: ワイルドカードを使用したパスレベルのメトリクス (p. 43)
- メトリクスポリシーの例: ルールが重複するパスレベルのメトリクス (p. 44)

メトリクスポリシーの例: コンテナレベルのメトリクス

このサンプルポリシーは、AWS Elemental MediaStore がコンテナレベルで Amazon CloudWatch にメトリクスを送信する必要があることを示します。たとえば、これには、コンテナに対して行われた `Put` リクエストの数をカウントする `RequestCount` メトリクスが含まれます。または、これを `DISABLED` に設定することもできます。

このポリシーにはルールがないため、MediaStore はパスレベルでメトリクスを送信しません。たとえば、このコンテナ内の特定のフォルダに対して行われた `Put` リクエストの数を確認することはできません。

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

メトリクスポリシーの例: パスレベルのメトリクス

このサンプルポリシーは、AWS Elemental MediaStore がコンテナレベルで Amazon CloudWatch にメトリクスを送信しないことを示します。また、MediaStore は、`baseball/saturday` および `football/saturday` の 2 つの特定のフォルダ内のオブジェクトのメトリクスを送信する必要があります。MediaStore リクエストのメトリクスは次のとおりです。

- `football/saturday` フォルダへのリクエストにはディメンション `ObjectGroupName=footballGroup` が含まれます。

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

メトリクスポリシーの例: コンテナレベルおよびパスレベルのメトリクス

このサンプルポリシーは、AWS Elemental MediaStore がコンテナレベルで Amazon CloudWatch にメトリクスを送信する必要があることを示します。また、MediaStore は、`baseball/saturday` および `football/saturday` の 2 つの特定のフォルダ内のオブジェクトのメトリクスを送信する必要があります。MediaStore リクエストのメトリクスは次のとおりです。

- `baseball/saturday` フォルダへのリクエストには `ObjectGroupName=baseballGroup` の CloudWatch ディメンションが含まれます。
- `football/saturday` フォルダへのリクエストには CloudWatch ディメンション `ObjectGroupName=footballGroup` が含まれます。

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

メトリクスポリシーの例: ワイルドカードを使用したパスレベルのメトリクス

このサンプルポリシーは、AWS Elemental MediaStore がコンテナレベルで Amazon CloudWatch にメトリクスを送信する必要があることを示します。また、MediaStore は、ファイル名に基づいてオブジェクトのメトリクスも送信する必要があります。ワイルドカードは、オブジェクトがコンテナ内のどこにでも保存され、`.m3u8` 拡張子で終わる限り、任意のファイル名を持つことができることを示します。

```
{
   "ContainerLevelMetrics": "ENABLED",
   "MetricPolicyRules": [
```

```
    {
      "ObjectGroup": "*.m3u8",
      "ObjectGroupName": "index"
    }
  ]
}
```

メトリクスポリシーの例: ルールが重複するパスレベルのメトリクス

このサンプルポリシーは、AWS Elemental MediaStore がコンテナレベルで Amazon CloudWatch にメトリクスを送信する必要があることを示します。さらに、MediaStore は、`sports/football/saturday` と `sports/football` の 2 つのフォルダのメトリクスを送信する必要があります。

`sports/football/saturday` フォルダへの MediaStore リクエストのメトリクスには、`ObjectGroupName=footballGroup1` の CloudWatch ディメンションが含まれます。`sports/football` フォルダに保存されているオブジェクトは両方のルールに一致するため、CloudWatch では、これらのオブジェクトの 2 つのデータポイントが表示されます。1 つは `ObjectGroupName=footballGroup1` のディメンションを含むデータポイントで、もう 1 つは `ObjectGroupName=footballGroup2` のディメンションを含むデータポイントです。

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

AWS Elemental MediaStore のフォルダ

フォルダはコンテナ内の区分です。フォルダを使用してコンテナを分割します。ファイルシステムでサブフォルダを作成してフォルダを分割するのと同じです。最大 10 レベルのフォルダを作成できます (コンテナ自体は含みません)。

フォルダは省略可能です。オブジェクトは、フォルダを使わずに直接コンテナにアップロードできます。ただし、フォルダを使用するとオブジェクトを整理しやすくなります。

オブジェクトをフォルダにアップロードするには、フォルダへのパスを指定します。フォルダが既に存在する場合、AWS Elemental MediaStore はそのフォルダにオブジェクトを保存します。フォルダが存在しない場合は、フォルダが自動的に作成されて、そのフォルダにオブジェクトが保存されます。

たとえば、`movies` というコンテナがあり、`mlaw.ts` というファイルをアップロードするとします。アップロードパスは `premium/canada` です。AWS Elemental MediaStore によってオブジェクトが premium フォルダの下の canada サブフォルダに保存されます。どちらのフォルダも存在しない場合は、`premium` フォルダと `canada` サブフォルダの両方が自動的に作成され、オブジェクトが `canada` サブフォルダに保存されます。パスを指定せずにコンテナ `movies` のみを指定すると、オブジェクトはコンテナに直接保存されます。

フォルダから最後のオブジェクトを削除すると、AWS Elemental MediaStore によってフォルダが自動的に削除されます。そのフォルダの上位にあるすべての空のフォルダも削除されます。たとえば、premium という名前のフォルダに、ファイルを一切含まず、 `canada`. [`canada` サブフォルダに次の名前が付けられたファイルが 1 つ含まれています: `mlaw.ts`. ファイルを削除すると、 `mlaw.ts`、サービスは両方の `premium` および `canada` フォルダ。この自動削除はフォルダにのみ適用されます。空のコンテナは削除されません。

トピック

- フォルダ名に関するルール (p. 45)
- フォルダの作成 (p. 46)
- フォルダの削除 (p. 46)

フォルダ名に関するルール

フォルダの名前を選択する際は、次の点に注意してください。

- 名前に使用できる文字は、大文字(A～Z)、小文字(a～z)、数字(0～9)、ピリオド(.)、ハイフン(-)、チルダ(～)、アンダースコア(_)、等号(=)、コロン(:)のみです。
- 名前は少なくとも 1 文字の長さにする必要があります。空のフォルダ名(`folder1//folder3/`) は使用できません。
- フォルダ名では大文字と小文字が区別されます。たとえば、`myFolder` フォルダと `myfolder` フォルダは一意の名前であるため、同じコンテナまたはフォルダ内で使用できます。
- 名前は親コンテナまたは親フォルダ内でのみ一意である必要があります。たとえば、 `myfolder` 2つの異なるコンテナで `movies/myfolder` および `sports/myfolder`.
- 名前は親コンテナと同じ名前にすることができます。
- フォルダを作成した後に名前を変更することはできません。

フォルダの作成

オブジェクトをアップロードするときにフォルダを作成できます。オブジェクトをフォルダにアップロードするには、フォルダへのパスを指定します。フォルダが既に存在する場合、AWS Elemental MediaStore はそのフォルダにオブジェクトを保存します。フォルダが存在しない場合は、フォルダが自動的に作成されて、そのフォルダにオブジェクトが保存されます。

詳細については、the section called “オブジェクトのアップロード” (p. 47) を参照してください。

フォルダの削除

フォルダが空の場合にのみフォルダを削除できます。オブジェクトが含まれているフォルダは削除できません。

フォルダから最後のオブジェクトを削除すると、AWS Elemental MediaStore によってフォルダが自動的に削除されます。そのフォルダの上位にあるすべての空のフォルダも削除されます。たとえば、 `premium` はファイルを含みませんが、サブフォルダの名前は1つ `canada`. [`canada` サブフォルダに次の名前が付けられたファイルが 1 つ含まれています: `mlaw.ts`. ファイルを削除すると、 `mlaw.ts`、サービスは両方の `premium` および `canada` フォルダ。この自動削除はフォルダにのみ適用されます。空のコンテナは削除されません。

詳細については、オブジェクトの削除 (p. 52) を参照してください。

AWS Elemental MediaStore のオブジェクト

AWS Elemental MediaStore のアセットは、オブジェクトと呼ばれます。オブジェクトは、コンテナまたはコンテナ内のフォルダにアップロードできます。

MediaStore では、オブジェクトのアップロード、ダウンロード、削除を行うことができます。

- アップロード - オブジェクトをコンテナまたはフォルダに追加します。これは、オブジェクトの作成とは異なります。オブジェクトを MediaStore にアップロードするには、事前にローカルで作成する必要があります。
- ダウンロード - オブジェクトを MediaStore から別の場所にコピーします。オブジェクトは MediaStore からは削除されません。
- 削除 - オブジェクトを MediaStore から完全に削除します。オブジェクトを個別に削除するか、オブジェクトライフサイクルポリシーを追加 (p. 32)して、指定された期間が経過するとコンテナ内のオブジェクトが自動的に削除されるように設定することができます。

MediaStore は、すべてのファイルタイプをサポートしています。

トピック

- オブジェクトのアップロード (p. 47)
- オブジェクトのリストの表示 (p. 48)
- オブジェクトの詳細の表示 (p. 50)
- オブジェクトのダウンロード (p. 50)
- オブジェクトの削除 (p. 51)

オブジェクトのアップロード

オブジェクトをコンテナ、またはコンテナ内のフォルダにアップロードできます。オブジェクトをフォルダにアップロードするには、フォルダへのパスを指定します。フォルダが既に存在する場合、AWS Elemental MediaStore はそのフォルダにオブジェクトを保存します。フォルダが存在しない場合は、フォルダが自動的に作成されて、そのフォルダにオブジェクトが保存されます。フォルダの詳細については、「AWS Elemental MediaStore のフォルダ (p. 45)」を参照してください。

オブジェクトをアップロードするには、MediaStore コンソールまたは AWS CLI を使用できます。

MediaStore では、オブジェクトのチャンク転送をサポートしており、アップロード中のオブジェクトをダウンロード可能できるため、レイテンシーが低減されます。この機能を使用するには、オブジェクトのアップロード可用性を次のように設定します。 `streaming`. このヘッダーの値は、 API を使用してオブジェクトをアップロード. リクエストでこのヘッダーを指定しない場合は、MediaStore によって、オブジェクトのアップロードの可用性にデフォルト値の `standard` が割り当てられます。

オブジェクトのサイズは、標準アップロードの可用性に対しては 25 MB を、ストリーミングアップロードの可用性に対しては 10 MB を超えることはできません。

> Note
>
> オブジェクト ファイル名には、文字、数字、ピリオド (.)、アンダースコア (_)、チルダ (~)、ハイフン (-)、等号 (=)、およびコロン (:) のみを使用できます。

オブジェクトをアップロードするには (コンソール)

1. MediaStore コンソール (https://console.aws.amazon.com/mediastore/) を開きます。
2. [Containers (コンテナ)] ページで、コンテナの名前を選択します。コンテナの詳細パネルが表示されます。
3. [Upload object (オブジェクトのアップロード)] を選択します。
4. [Target path (ターゲットのパス)] に、フォルダへのパスを入力します たとえば、 `premium/canada`. 指定したパス内のフォルダがまだ存在しない場合、サービスによって自動的に作成されます。
5. [オブジェクト] セクションで、[参照] を選択します。
6. 適切なフォルダに移動し、アップロードする 1 つのオブジェクトを選択します。
7. [Open (開く)]、[Upload (アップロード)] の順に選択します。

   Note

   選択したフォルダ内に同じ名前のファイルが既に存在する場合、元のファイルはアップロードしたファイルで置き換えられます。

オブジェクトをアップロードするには (AWS CLI)

- AWS CLI で、`put-object` コマンドを使用します。また、次のパラメータのどれでも含めることができます。`content-type`、 `cache-control` (呼び出し元がオブジェクトのキャッシュ動作を制御できるようにするため) `path` (コンテナー内のフォルダーにオブジェクトを配置するため)。

  Note

  オブジェクトをアップロードした後、`content-type`、`cache-control`、または `path` を編集することはできません。

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

戻り値の例を以下に示します。

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

オブジェクトのリストの表示

AWS Elemental MediaStore コンソールでは、最上位のコンテナまたはフォルダに保存されている項目 (オブジェクトとフォルダ) を表示できます。現在のコンテナまたはフォルダのサブフォルダに保存されている項目は表示されません。コンテナ内のオブジェクトとフォルダのリストは、AWS CLI を使用して表示できます。コンテナ内のフォルダやサブフォルダの数には関係ありません。

特定のコンテナに含まれているオブジェクトのリストを表示するには (コンソール)

1. MediaStore コンソール (https://console.aws.amazon.com/mediastore/) を開きます。
2. [Containers (コンテナ)] ページで、表示するフォルダが含まれているコンテナの名前を選択します。
3. リストからフォルダの名前を選択します。

詳細ページに、フォルダに保存されているすべてのフォルダとオブジェクトが表示されます。

特定のフォルダに含まれているオブジェクトのリストを表示するには (コンソール)

1. MediaStore コンソール (https://console.aws.amazon.com/mediastore/) を開きます。
2. [Containers (コンテナ)] ページで、表示するフォルダが含まれているコンテナの名前を選択します。

   詳細ページに、コンテナに保存されているすべてのフォルダとオブジェクトが表示されます。

特定のコンテナに含まれているオブジェクトとフォルダのリストを表示するには (AWS CLI)

- AWS CLI で、`list-items` コマンドを使用します。

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

戻り値の例を以下に示します。

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

> **Note**
>
> `seconds_since_create` ルールの対象なるオブジェクトは、`list-items` レスポンスには含まれていません。

特定のフォルダに含まれているオブジェクトとフォルダのリストを表示するには (AWS CLI)

- AWS CLI で、`list-items` コマンドを使用します。リクエストの最後にフォルダ名を指定します。

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

戻り値の例を以下に示します。

```
{
    "Items": [
        {
            "Type": "FOLDER",
            "Name": "folder_1"
        },
```

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

Note

`seconds_since_create` ルールの対象なるオブジェクトは、`list-items` レスポンスには含まれていません。

オブジェクトの詳細の表示

オブジェクトをアップロードした後、 AWS Elemental MediaStore 変更日、コンテンツの長さ、 ETag (エンティティタグ)、およびコンテンツタイプ。オブジェクトのメタデータの使用方法については、「MediaStore による HTTP キャッシュの操作 (p. 90)」を参照してください。

オブジェクトの詳細を表示するには (コンソール)

1. MediaStore コンソール (https://console.aws.amazon.com/mediastore/) を開きます。
2. [Containers (コンテナ)] ページで、表示するオブジェクトが含まれているコンテナの名前を選択します。
3. 表示するオブジェクトがフォルダ内にある場合は、オブジェクトが表示されるまで繰り返しフォルダ名を選択します。
4. オブジェクトの名前を選択します。

   詳細ページにオブジェクトに関する情報が表示されます。

オブジェクトの詳細を表示するには (AWS CLI)

- AWS CLI で、`describe-object` コマンドを使用します。

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

戻り値の例を以下に示します。

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

オブジェクトのダウンロード

オブジェクトをダウンロードするには、コンソールを使用できます。オブジェクトまたはオブジェクトの一部をダウンロードするには、AWS CLI を使用できます。

オブジェクトをダウンロードするには (コンソール)

1. MediaStore コンソール (https://console.aws.amazon.com/mediastore/) を開きます。
2. [Containers (コンテナ)] ページで、ダウンロードするオブジェクトが含まれているコンテナの名前を選択します。
3. ダウンロードするオブジェクトがフォルダ内にある場合は、オブジェクトが表示されるまで繰り返しフォルダ名を選択します。
4. オブジェクトの名前を選択します。
5. [オブジェクト] の詳細ページで、[Download (ダウンロード)] を選択します。

オブジェクトをダウンロードするには (AWS CLI)

- AWS CLI で、`get-object` コマンドを使用します。

```
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
```

戻り値の例を以下に示します。

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

オブジェクトの一部をダウンロードするには (AWS CLI)

- AWS CLI で、`get-object` コマンドを使用し、範囲を指定します。

```
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
```

戻り値の例を以下に示します。

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

オブジェクトの削除

AWS Elemental MediaStore には、コンテナからオブジェクトを削除するためのさまざまなオプションがあります。

- 個々のオブジェクトを削除します (p. 52)。料金が適用されます。

- コンテナを空にして (p. 52)、コンテナ内のすべてのオブジェクトを一度に削除します。このプロセスでは API コールが使用されるため、通常の API 料金が適用されます。
- オブジェクトライフサイクルポリシーを追加して (p. 32)特定の期間に達したときにオブジェクトを削除します。料金が適用されます。

オブジェクトの削除

コンソールまたは AWS CLI を使用してオブジェクトを個別に削除することができます。または、オブジェクトライフサイクルポリシーを追加して (p. 32)、コンテナ内の特定の期間に達した後にオブジェクトを自動的に削除したり、コンテナを空にして (p. 52)そのコンテナ 内のすべてのオブジェクトを削除できます。

> Note
>
> フォルダ内の唯一のオブジェクトを削除すると、AWS Elemental MediaStore は自動的にそのフォルダを削除し、さらにそのフォルダの上位にある空のフォルダも削除します。たとえば、`premium` はファイルを含みませんが、サブフォルダの名前は1つ `canada`. [`canada` サブフォルダに次の名前が付けられたファイルが 1 つ含まれています: `mlaw.ts`. ファイルを削除すると、`mlaw.ts`、サービスは両方の `premium` および `canada` フォルダ。

オブジェクトを削除するには (コンソール)

1. MediaStore コンソール (https://console.aws.amazon.com/mediastore/) を開きます。
2. [Containers (コンテナ)] ページで、削除したいオブジェクトが含まれているコンテナの名前を選択します。
3. 削除するオブジェクトがフォルダ内にある場合は、オブジェクトが表示されるまで繰り返しフォルダ名を選択します。
4. オブジェクト名の左にあるオプションを選択します。
5. [Delete (削除)] を選択します。

オブジェクトを削除するには (AWS CLI)

- AWS CLI で、`delete-object` コマンドを使用します。

  例:

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

  このコマンドの戻り値はありません。

コンテナを空にする

コンテナを空にすると、コンテナ内に保存されているすべてのオブジェクトを削除できます。または、オブジェクトライフサイクルポリシー (p. 38)を追加して、コンテナ内の特定の期間に達した後にオブジェクトを自動的に削除したり、オブジェクトを個別に削除できます (p. 52)。

コンテナを空にするには (コンソール)

1. MediaStore コンソール (https://console.aws.amazon.com/mediastore/) を開きます。
2. [コンテナ] ページで、空にするコンテナのオプションを選択します。
3. [Empty container (コンテナを空にする)] を選択します。確認メッセージが表示されます。

4. フィールドに **`empty`** を入力し、[Empty (空にする)] を選択します。

AWS Elemental MediaStore のセキュリティ

AWS では、クラウドのセキュリティが最優先事項です。AWS のお客様は、セキュリティを最も重視する組織の要件を満たすように構築されたデータセンターとネットワークアーキテクチャーから利点を得られます。

セキュリティは、AWS とお客様の間の共有責任です。共有責任モデルでは、これをクラウドのセキュリティおよびクラウド内のセキュリティとして説明しています。

- クラウドのセキュリティ – AWS は、AWS クラウド内で AWS サービスを実行するインフラストラクチャを保護する責任を担います。また、AWS は、使用するサービスを安全に提供します。AWS コンプライアンスプログラムの一環として、サードパーティーの監査が定期的にセキュリティの有効性をテストおよび検証しています。AWS Elemental MediaStore に適用するコンプライアンスプログラムの詳細については、「コンプライアンスプログラムによる AWS 対象範囲内のサービス」を参照してください。
- クラウド内のセキュリティ – お客様の責任はお客様が使用する AWS のサービスによって決まります。また、お客様は、お客様のデータの機密性、企業の要件、および適用可能な法律および規制などの他の要因についても責任を担います。

このドキュメントでは、MediaStore を使用する際に責任共有モデルを適用する方法について説明します。以下のトピックでは、セキュリティおよびコンプライアンスの目的を達成するように MediaStore を設定する方法について説明します。また、MediaStore リソースのモニタリングや保護に役立つ他の AWS のサービスの使用方法についても説明します。

トピック

- AWS Elemental MediaStore でのデータ保護 (p. 54)
- AWS Elemental MediaStore での Identity and Access Management (p. 55)
- AWS Elemental MediaStore でのログ記録とモニタリング (p. 70)
- AWS Elemental MediaStore のコンプライアンス検証 (p. 70)
- AWS Elemental MediaStore での耐障害性 (p. 71)
- AWS Elemental MediaStore でのインフラストラクチャセキュリティ (p. 71)

AWS Elemental MediaStore でのデータ保護

AWS 責任共有モデルは、AWS Elemental MediaStore のデータ保護に適用されます。このモデルで説明したように、AWS は、すべての AWS クラウドを実行するグローバルインフラストラクチャを保護します。お客様は、このインフラストラクチャでホストされているコンテンツに対する管理を維持する責任があります。このコンテンツには、使用する AWS サービスのセキュリティ設定および管理タスクが含まれます。データプライバシーの詳細については、「データプライバシーのよくある質問」を参照してください。 欧州でのデータ保護の詳細については、AWS セキュリティブログの「AWS の責任共有モデルと GDPR」のブログ記事を参照してください。

データ保護の目的で、AWS アカウントの認証情報を保護し、個々のユーザーアカウントを AWS Identity and Access Management (IAM) で設定することをお勧めします。この方法により、それぞれの職務を遂行するために必要なアクセス許可のみを各ユーザーに付与できます。また、以下の方法でデータを保護することをお勧めします。

- 各アカウントで多要素認証 (MFA) を使用します。
- SSL/TLS を使用して AWS リソースと通信します。TLS 1.2 以降が推奨されています。
- AWS CloudTrail で API とユーザーアクティビティログをセットアップします。
- AWS 暗号化ソリューションを、AWS サービス内のすべてのデフォルトのセキュリティ管理と一緒に使用します。
- Amazon Macie などの高度なマネージドセキュリティサービスを使用します。これにより、Amazon S3 に保存される個人データの検出と保護が支援されます。
- コマンドラインインターフェイスまたは API を使用して AWS にアクセスするときに FIPS 140-2 検証済みの暗号化モジュールが必要な場合は、FIPS エンドポイントを使用します。使用可能な FIPS エンドポイントの詳細については、「連邦情報処理規格 (FIPS) 140-2」を参照してください。

顧客のアカウント番号などの機密の識別情報は、[名前] フィールドなどの自由形式のフィールドに配置しないことを強くお勧めします。これは、コンソール、API、AWS CLI、または AWS で MediaStore または他の AWS サービスを使用する場合も同様です。MediaStore または他のサービスに入力したデータはすべて、診断ログの内容として取得される可能性があります。外部サーバーへの URL を指定するときは、そのサーバーへのリクエストを検証するための認証情報を URL に含めないでください。

は、業界標準の AES-256 アルゴリズムを使用して、保管時のコンテナとオブジェクトをMediaStore暗号化します。を使用して、以下の方法でデータを保護することをお勧めします。MediaStore

- コンテナポリシーを作成して、そのコンテナ内のすべてのフォルダとオブジェクトへのアクセス権限を制御します。詳細については、the section called “コンテナポリシー” (p. 14) を参照してください。
- クロスオリジンリソース共有 (CORS) ポリシーを作成して、MediaStoreクロスオリジンアクセスをリソースに選択的に許可します。CORS を使用すると、特定のドメインにロードされたクライアントウェブアプリケーションが、異なるドメイン内のリソースと通信できるようになります。詳細については、the section called “CORS ポリシー” (p. 22) を参照してください。

AWS Elemental MediaStore での Identity and Access Management

AWS Identity and Access Management (IAM) は、管理者が AWS リソースへのアクセスを安全にコントロールするために役立つ AWS のサービスです。IAM 管理者は、MediaStore リソースを使用するために認証 (サインイン) および承認 (アクセス許可を持つ) される者を制御します。IAM は、追加料金なしで使用できる AWS のサービスです。

このセクションでは、MediaStore を使用するための設定手順に関する背景情報や追加情報を提供し ます。「セットアップ (p. 4)」を参照してください。

Audience

AWS Identity and Access Management (IAM) の使用方法は、MediaStore で行う作業によって異なります。

サービスユーザー – ジョブを実行するために MediaStore サービスを使用する場合は、管理者が必要なアクセス許可と認証情報を用意します。作業を実行するためにさらに多くの MediaStore 機能を使用するとき、追加のアクセス許可が必要になる場合があります。アクセスの管理方法を理解すると、管理者から適切なアクセス許可をリクエストするのに役に立ちます。MediaStore の機能にアクセスできない場合は、「AWS Elemental MediaStore Identity and Access のトラブルシューティング (p. 68)」を参照してください。

サービス管理者 – 社内の MediaStore リソースを担当している場合は、おそらく MediaStore へのフルアクセスがあります。従業員がどの MediaStore 機能とリソースアクセスする必要があるかを決定するのは

管理者の仕事です。その後で、サービスユーザーのアクセス許可を変更するために、IAM 管理者にリクエストを送信する必要があります。IAM の基本概念については、このページの情報を確認します。お客様の会社で MediaStore の IAM を利用する方法の詳細については、「AWS Elemental MediaStore と IAM の連携 (p. 60)」を参照して ください。

IAM 管理者 – IAM 管理者は、MediaStore へのアクセスを管理するポリシーの作成方法の詳細について確認する場合があります。IAM で使用できる MediaStore アイデンティティベースのポリシーの例を表示するには、「AWS Elemental MediaStore アイデンティティベースのポリシーの例 (p. 63)」を参照して ください。

アイデンティティを使用した認証

認証は、アイデンティティ認証情報を使用して AWS にサインインする方法です。AWS マネジメントコンソール を使用したサインインの詳細については、IAM ユーザーガイド の「IAM ユーザーまたは ルートユーザー としての AWS マネジメントコンソール へのログイン」を参照してください。

AWS アカウントのルートユーザー、IAM ユーザーとして、または IAM ロールを引き受けて、認証されている (AWS にサインインしている) 必要があります。会社のシングルサインオン認証を使用することも、Google や Facebook を使用してサインインすることもできます。このような場合、管理者は以前に IAM ロールを使用して ID フェデレーションを設定しました。他の会社の認証情報を使用して AWS にアクセスした場合、ロールを間接的に割り当てられています。

AWS マネジメントコンソール に直接サインインするには、ルートユーザー E メールまたは IAM ユーザー名とパスワードを使用します。ルートユーザー または IAM ユーザーのアクセスキーを使用して AWS にプログラム的にアクセスできます。AWS では、SDK とコマンドラインツールを提供し、お客様の認証情報を使用して、リクエストに暗号で署名できます。AWS ツールを使用しない場合は、リクエストに自分で署名する必要があります。これには、インバウンド API リクエストを認証するためのプロトコル、署名バージョン 4 を使用します。リクエストの認証の詳細については、AWS General Referenceの「署名バージョン 4 の署名プロセス」を参照してください。

使用する認証方法を問わず、追加のセキュリティ情報の提供を要求される場合もあります。たとえば、AWS では多要素認証 (MFA) を使用してアカウントのセキュリティを高めることを推奨しています。詳細については、IAM ユーザーガイドの「AWS での多要素認証 (MFA) の使用」を参照してください。

AWS アカウントのルートユーザー

AWS アカウントを初めて作成する場合は、このアカウントのすべての AWS サービスとリソースに対して完全なアクセス権限を持つシングルサインインアイデンティティで始めます。このアイデンティティは AWS アカウント ルートユーザー と呼ばれ、アカウントの作成に使用した E メールアドレスとパスワードでのサインインによりアクセスします。強くお勧めしているのは、日常的なタスクには、それが管理者タスクであっても、ルートユーザーを使用しないことです。代わりに、最初の IAM ユーザーを作成するためだけに ルートユーザー を使用するというベストプラクティスに従います。その後、ルートユーザー認証情報を安全な場所に保管し、それらを使用して少数のアカウントおよびサービス管理タスクのみを実行します。

IAM ユーザーとグループ

IAM ユーザーは、単一のユーザーまたはアプリケーションに特定のアクセス許可がある AWS アカウント内のアイデンティティです。IAM ユーザーは、ユーザー名とパスワード、アクセスキーのセットなど、長期的な認証情報を持つことができます。アクセスキーを生成する方法については、IAM ユーザーガイドの「IAM ユーザーのアクセスキーの管理」を参照してください。IAM ユーザーにアクセスキーを生成するとき、必ずキーペアを表示して安全に保存してください。後になって、シークレットアクセスキーを回復することはできません。新しいアクセスキーペアを生成する必要があります。

IAM グループは、IAM ユーザーのコレクションを指定するアイデンティティです。グループとしてサインインすることはできません。グループを使用して、一度に複数のユーザーに対してアクセス許可を指定で

きます。多数の組のユーザーがある場合、グループを使用すると管理が容易になります。たとえば、IAM Admin という名前のグループを設定して、そのグループに IAM リソースを管理するアクセス許可を与えることができます。

ユーザーは、ロールとは異なります。ユーザーは 1 人の特定の人またはアプリケーションに一意に関連付けられますが、ロールはそれを必要とする任意の人が引き受けるようになっています。ユーザーには永続的な長期の認証情報がありますが、ロールでは一時的な認証情報が利用できます。詳細については、IAM ユーザーガイドの「IAM ユーザーの作成が適している場合 (ロールではなく)」を参照してください。

IAM ロール

IAM ロールは、特定のアクセス許可を持つ、AWS アカウント内のアイデンティティです。これは IAM ユーザーに似ていますが、特定のユーザーに関連付けられていません。ロールを切り替えて、AWS マネジメントコンソール で IAM ロールを一時的に引き受けることができます。ロールを引き受けるには、AWS CLI または AWS API オペレーションを呼び出すか、カスタム URL を使用します。ロールを使用する方法の詳細については、IAM ユーザーガイドの「IAM ロールの使用」を参照してください。

IAM ロールと一時的な認証情報は、次の状況で役立ちます。

- 一時的な IAM ユーザーアクセス許可 – IAM ユーザーは、特定のタスクに対して複数の異なるアクセス許可を一時的に IAM ロールで引き受けることができます。
- フェデレーティッドユーザーアクセス – IAM ユーザーを作成する代わりに、AWS Directory Service、エンタープライズユーザーディレクトリ、またはウェブ ID プロバイダーに既存のアイデンティティを使用できます。このようなユーザーはフェデレーティッドユーザーと呼ばれます。AWS では、ID プロバイダーを通じてアクセスがリクエストされたとき、フェデレーティッドユーザーにロールを割り当てます。フェデレーティッドユーザーの詳細については、IAM ユーザーガイド の「フェデレーティッドユーザーとロール」を参照してください。
- クロスアカウントアクセス – IAM ロールを使用して、自分のアカウントのリソースにアクセスすることを別のアカウントの信頼済みプリンシパルに許可できます。ロールは、クロスアカウントアクセスを許可する主な方法です。ただし、一部の AWS のサービスでは、(ロールをプロキシとして使用する代わりに) リソースにポリシーを直接アタッチできます。クロスアカウントアクセスでのロールとリソースベースのポリシーの違いの詳細については、IAM ユーザーガイドの「IAM ロールとリソースベースのポリシーとの相違点」を参照してください。
- サービス間アクセス – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.
  - プリンシパルアクセス許可 – When you use an IAM user or role to perform actions in AWS, you are considered a principal. Policies grant permissions to a principal. When you use some services, you might perform an action that then triggers another action in a different service. In this case, you must have permissions to perform both actions. To see whether an action requires additional dependent actions in a policy, see Actions, Resources, and Condition Keys for AWS Elemental MediaStore in the Service Authorization Reference.
  - サービスロール – サービスロールは、サービスがお客様に代わってアクションを実行するために引き受ける IAM ロールです。サービスロールは、お客様のアカウント内のみでアクセスを提供します。他のアカウントのサービスへのアクセス権を付与するためにサービスロールを使用することはできません。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細については、IAM ユーザーガイドの「AWS サービスにアクセス許可を委任するロールの作成」を参照してください。
  - サービスにリンクされたロール – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your IAM account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.
- Amazon EC2で実行されているアプリケーション – IAM ロールを使用して、EC2 インスタンスで実行され、AWS CLI または AWS API リクエストを作成しているアプリケーションの一時的な認証情報を管理できます。これは、EC2 インスタンス内でのアクセスキーの保存に推奨されます。AWS ロールを EC2

インスタンスに割り当て、そのすべてのアプリケーションで使用できるようにするには、インスタンスにアタッチされたインスタンスプロファイルを作成します。インスタンスプロファイルにはロールが含まれ、EC2 インスタンスで実行されるプログラムは一時認証情報を取得することができます。詳細については、IAM ユーザーガイド の「Amazon EC2 インスタンスで実行されるアプリケーションに IAM ロールを使用してアクセス許可を付与する」を参照してください。

IAM ロールを使用するか IAM ユーザーを使用するかについては、IAM ユーザーガイド の「IAM ロールの作成が適している場合 (ユーザーではなく)」を参照してください。

ポリシーを使用したアクセスの管理

AWS でアクセスをコントロールするには、ポリシーを作成して IAM アイデンティティや AWS リソースにアタッチします。ポリシーは AWS のオブジェクトであり、ID やリソースに関連付けて、これらのアクセス許可を定義します。ルートユーザー または IAM ユーザーとしてサインインすることも、IAM ロールを引き受けることもできます。リクエストを行うと、AWS は関連する ID ベースまたはリソースベースのポリシーを評価します。ポリシーでのアクセス許可により、リクエストが許可されるか拒否されるかが決まります。大半のポリシーは JSON ドキュメントとして AWS に保存されます。JSON ポリシードキュメントの構造と内容の詳細については、IAM ユーザーガイドの「JSON ポリシーの概要」を参照してください。

Administrators can use AWS JSON policies to specify who has access to what. That is, which principal can perform actions on what resources, and under what conditions.

すべての IAM エンティティ (ユーザーまたはロール) は、アクセス許可のない状態からスタートします。言い換えると、デフォルト設定では、ユーザーは何もできず、自分のパスワードを変更することすらできません。何かを実行するアクセス許可をユーザーに付与するには、管理者がユーザーにアクセス許可ポリシーをアタッチする必要があります。また、管理者は、必要なアクセス許可があるグループにユーザーを追加できます。管理者がグループにアクセス許可を付与すると、そのグループ内のすべてのユーザーにこれらのアクセス許可が付与されます。

IAM ポリシーは、オペレーションの実行方法を問わず、アクションのアクセス許可を定義します。たとえば、`iam:GetRole` アクションを許可するポリシーがあるとします。このポリシーがあるユーザーは、AWS マネジメントコンソール、AWS CLI、または AWS API からロールの情報を取得できます。

アイデンティティベースのポリシー

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Creating IAM policies in the IAM ユーザーガイド.

アイデンティティベースのポリシーは、さらにインラインポリシーまたは管理ポリシーに分類できます。インラインポリシーは、単一のユーザー、グループ、またはロールに直接埋め込まれています。管理ポリシーは、AWS アカウント内の複数のユーザー、グループ、およびロールにアタッチできるスタンドアロンポリシーです。管理ポリシーには、AWS 管理ポリシーとカスタマー管理ポリシーが含まれます。管理ポリシーまたはインラインポリシーのいずれかを選択する方法については、IAM ユーザーガイドの「管理ポリシーとインラインポリシーの比較」を参照してください。

リソースベースのポリシー

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM role trust policies and Amazon S3 bucket policies. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

リソースベースのポリシーは、そのサービス内にあるインラインポリシーです。リソースベースのポリシーで IAM の AWS 管理ポリシーを使用することはできません。

アクセスコントロールリスト (ACL)

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

ACL をサポートするサービスの例としては、Amazon S3、AWS WAF、Amazon VPC などがあります。ACL の詳細については、Amazon Simple Storage Service 開発者ガイドの「アクセスコントロールリスト (ACL) の概要」を参照してください。

その他のポリシータイプ

AWS では、別のあまり一般的ではないポリシータイプもサポートしています。これらのポリシータイプでは、より一般的なポリシータイプで付与された最大のアクセス許可を設定できます。

- アクセス許可の境界 – アクセス許可の境界は、アイデンティティベースのポリシーが IAM エンティティ (IAM ユーザーまたはロール) に付与できるアクセス許可の上限を設定する高度な機能です。エンティティのアクセス許可の境界を設定できます。結果として得られるアクセス許可は、エンティティの ID ベースのポリシーとそのアクセス許可の境界の共通部分です。`Principal` フィールドでユーザーまたはロールを指定するリソースベースのポリシーは、アクセス許可の境界では制限されません。これらのポリシーのいずれかを明示的に拒否した場合、その許可は無効になります。アクセス許可の境界の詳細については、IAM ユーザーガイドの「IAM エンティティのアクセス許可の境界」を参照してください。
- サービスコントロールポリシー (SCP) – SCP は、AWS Organizations で 組織や組織単位 (OU) に最大権限を指定する JSON ポリシーです。AWS Organizations は、お客様のビジネスが所有する複数の AWS アカウントをグループ化し、一元的に管理するサービスです。組織内のすべての機能を有効にすると、サービス制御ポリシー (SCP) を一部またはすべてのアカウントに適用できます。SCP はメンバーアカウントのエンティティに対するアクセス許可を制限します (各 AWS アカウントのルートユーザー など)。Organizations および SCP の詳細については、AWS Organizations ユーザーガイドの「SCP の動作」を参照してください。
- セッションポリシー – セッションポリシーは、ロールまたはフェデレーティッドユーザーの一時セッションをプログラムで作成する際にパラメータとして渡す高度なポリシーです。結果として得られるセッションのアクセス許可は、ユーザーまたはロールの ID ベースのポリシーとセッションポリシーの共通部分です。また、リソースベースのポリシーからアクセス許可が派生する場合もあります。これらのポリシーのいずれかを明示的に拒否した場合、その許可は無効になります。詳細については、IAM ユーザーガイドの「セッションポリシー」を参照してください。

複数のポリシータイプ

1 つのリクエストに複数のタイプのポリシーが適用されると、結果として作成されるアクセス許可を理解するのがさらに複雑になります。複数のポリシータイプが関連する場合にリクエストを許可するかどうかを AWS で決定する方法の詳細については、IAM ユーザーガイドの「ポリシーの評価ロジック」を参照してください。

詳細はこちら

MediaStore の Identity and Access Management に関する詳細については、以下のページに進んでください。

- MediaStore と IAM の連携 (p. 60)
- アイデンティティベースのポリシーの例 (p. 63)
- リソースベースのポリシーの例 (p. 65)
- トラブルシューティング (p. 68)

AWS Elemental MediaStore と IAM の連携

IAM を使用して、MediaStore へのアクセスを管理するには、MediaStore で使用できる IAM の機能を理解しておく必要があります。IAM と連携する方法およびその他のMediaStoreサービスの概要については、の「 と連携するAWSサービスAWSIAM」を参照してくださいIAM ユーザーガイド。

MediaStore アイデンティティベースのポリシー

IAM アイデンティティベースのポリシーでは、許可または拒否されたアクションやリソースを指定でき、さらにアクションが許可または拒否された条件を指定できます。MediaStore は、特定のアクション、リソース、条件キーをサポートします。JSON ポリシーで使用するすべての要素については、の「IAMJSON ポリシーエレメントのリファレンス」を参照してくださいIAM ユーザーガイド。

Actions

Administrators can use AWS JSON policies to specify who has access to what. That is, which principal can perform actions on what resources, and under what conditions.

JSON ポリシーの `Action` 要素は、ポリシー内のアクセスを許可または拒否するために使用できるアクションを記述します。ポリシーアクションの名前は通常、関連する AWS API オペレーションと同じです。ただし、一致する API オペレーションを持たないアクセス許可のみのアクションなど、いくつかの例外があります。また、ポリシーに複数のアクションが必要なオペレーションもあります。これらの追加アクションは、従属アクションと呼ばれます。

このアクションは、関連付けられたオペレーションを実行するためのアクセス許可を付与するポリシーで使用されます。

のポリシーアクションMediaStoreは、アクションの前に次のプレフィックスを使用します。`mediastore:`。 たとえば、MediaStore`CreateContainer`API オペレーションを使用して新しいコンテナを作成する権限を付与するには、ポリシーに`mediastore:CreateContainer`アクションを含めます。ポリシーステートメントには、`Action` または `NotAction` 要素を含める必要があります。MediaStore は、このサービスで実行できるタスクを説明する独自の一連のアクションを定義します。

単一のステートメントに複数のアクションを指定するには、次のようにコンマで区切ります。

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

ワイルドカード (*) を使用して複数のアクションを指定できます。たとえば、`Describe` という単語で始まるすべてのアクションを指定するには、以下のアクションを含めます。

```
"Action": "mediastore:Describe*"
```

MediaStore のアクションを一覧表示するには、IAM ユーザーガイドの「AWS Elemental MediaStore で定義したアクション」を参照してください。

Resources

Administrators can use AWS JSON policies to specify who has access to what. That is, which principal can perform actions on what resources, and under what conditions.

`Resource` JSON ポリシー要素は、アクションが適用されるオブジェクトを指定します。ステートメントには、`Resource` または `NotResource` エレメントを含める必要があります。ベストプラクティスとして、リソースは Amazon リソースネーム (ARN) を使用して指定します。これは、リソース レベルのアクセス許可と呼ばれる特定のリソースタイプをサポートするアクションに対して実行できます。

オペレーションのリスト化など、リソースレベルのアクセス許可をサポートしないアクションの場合は、ワイルドカード (*) を使用して、ステートメントがすべてのリソースに適用されることを示します。

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

MediaStore コンテナリソースには次の ARN が含まれています。

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

ARNsの形式の詳細については、「Amazon リソースネーム (ARN)」とAWS「サービスの名前空間」を参照してください。

たとえば、ステートメントで `AwardsShow` コンテナを指定するには、次の ARN を使用します。

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

特定のアカウントに属するすべてのインスタンスを指定するには、ワイルドカード (*) を使用します。

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

リソースの作成など、一部の MediaStore アクションは、特定のリソースで実行できません。このような場合は、ワイルドカード (*) を使用する必要があります。

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

MediaStoreリソースタイプおよびその ARNs のリストを表示するには、の を参照してくださいAWS Elemental MediaStore で定義したリソースIAM ユーザーガイド。どのアクションで、各リソースの ARN を指定することができるかについては、「AWS Elemental MediaStore で定義したアクション」を参照してください。

条件キー

MediaStore にはサービス固有条件キーがありませんが、いくつかのグローバル条件キーの使用がサポートされています。すべてのAWSグローバル条件キーを確認するには、 の「 AWS グローバル条件コンテキストキー」を参照してくださいIAM ユーザーガイド。

Examples

MediaStore アイデンティティベースのポリシーの例については、「AWS Elemental MediaStore アイデンティティベースのポリシーの例 (p. 63)」を参照してください。

MediaStore リソースベースのポリシー

リソースベースのポリシーは、指定されたプリンシパルが MediaStore リソースに対して実行できるアクションおよび実行条件を指定する JSON ポリシードキュメントです。MediaStore は、MediaStore コンテナのリソースベースのアクセス許可ポリシーをサポートしています。リソースベースのポリシーでは、リソースごとに他のアカウントに使用許可を付与することができます。リソースベースのポリシーを使用して、MediaStore コンテナへのアクセスを AWS サービスに許可することもできます。

クロスアカウントアクセスを有効にするには、全体のアカウント、または別のアカウントの IAM エンティティを、リソースベースのポリシーのプリンシパルとして指定します。リソースベースのポリシーにクロスアカウントのプリンシパルを追加しても、信頼関係は半分しか確立されない点に注意してください。プリンシパルとリソースが異なる AWS アカウントにある場合は、リソースにアクセスするためのアクセス許可をプリンシパルエンティティにも付与する必要があります。アクセス許可は、アイデンティティベースのポリシーをエンティティにアタッチすることで付与します。ただし、リソースベースのポリシーで、

同じアカウントのプリンシパルへのアクセス権が付与されている場合は、アイデンティティベースのポリシーをさらに付与する必要はありません。詳細については、次のガイドの「IAMロールとリソースベースのポリシーとの相違点」を参照してくださいIAM ユーザーガイド。

Note

MediaStoreは、コンテナでアクションを実行できるプリンシパルエンティティ (アカウント、ユーザー、ロール、フェデレーティッドユーザー) を定義するコンテナポリシーもサポートしています。詳細については、コンテナポリシー (p. 14) を参照してください。

Examples

MediaStore のリソースベースのポリシーの例については、「the section called “リソースベースのポリシーの例” (p. 65)」を参照してください。

MediaStore タグに基づいた承認

タグを MediaStore リソースにアタッチすることも、MediaStore へのリクエストでタグを渡すこともできます。タグに基づいてアクセスを制御するには、`mediastore:ResourceTag/key-name`、`aws:RequestTag/key-name`、`aws:TagKeys` 条件キーを使用して、ポリシーの条件要素でタグ情報を提供します。リソースにタグを付けるには、 API を使用する必要があります。MediaStoreリソースにタグを割り当てる方法の詳細については、次のTagResourceガイドを参照してくださいAWS Elemental MediaStore API リファレンス。

リソースのタグに基づいてリソースへのアクセスを制限するためのアイデンティティベースのポリシーの例については、「MediaStoreリソースに対するタグに基づいてアクションを許可または拒否する (p. 67)」を参照してください。

MediaStore の IAM ロール

IAM ロールは、特定のアクセス許可を持つ、AWS アカウント内のエンティティです。

MediaStore を使用した一時的な認証情報の使用

一時的な認証情報を使用して、フェデレーションでサインイン、IAM ロールを引き受ける、またはクロスアカウントロールを引き受けることができます。AWS STSAssumeRoleや などの API オペレーションを呼び出して、一時的なセキュリティ認証情報を取得しますGetFederationToken。

MediaStore では、一時認証情報の使用をサポートしています。

サービスにリンクされたロール

サービスにリンクされたロールによって、AWS サービスが他のサービスのリソースにアクセスして自動的にアクションを完了できます。サービスにリンクされたロールは、IAM アカウント内に表示され、サービスによって所有されます。IAM 管理者は、サービスにリンクされたロールのアクセス許可を表示できますが、編集することはできません。

MediaStore ではサービスにリンクされたロールをサポートしていません。

サービスロール

この機能では、サービスのロールをユーザーに代わって引き受けることをサービスに許可します。このロールにより、サービスはお客様に代わって他のサービスのリソースにアクセスし、アクションを実行できます。サービスロールは、IAM アカウントに表示され、サービスによって所有されます。つまり、IAM 管理者は、このロールのアクセス許可を変更できます。ただし、これを行うことにより、サービスの機能が損なわれる場合があります。

MediaStore ではサービスロールがサポートされています。

AWS Elemental MediaStore アイデンティティベースのポリシーの例

デフォルトでは、IAM ユーザーおよびロールには、MediaStore リソースを作成または変更するアクセス許可はありません。また、AWS マネジメントコンソール、AWS CLI、あるいは AWS API を使用してタスクを実行することもできません。IAM 管理者は、ユーザーとロールに必要な、指定されたリソースで特定の API オペレーションを実行するアクセス許可をユーザーとロールに付与する IAM ポリシーを作成する必要があります。続いて、管理者はそれらのアクセス許可が必要な IAM ユーザーまたはグループにそのポリシーをアタッチします。

JSON ポリシードキュメントのこれらの例を使用して、IAMアイデンティティベースのポリシーを作成する方法については、 の「JSON タブのポリシーの作成」を参照してくださいIAM ユーザーガイド。

ポリシーのベストプラクティス

アイデンティティベースのポリシーは非常に強力です。アカウント内で、MediaStore リソースを作成、アクセス、または削除できるかどうかを決定します。これらのアクションを実行すると、AWS アカウントに追加料金が発生する可能性があります。アイデンティティベースのポリシーを作成または編集するときは、以下のガイドラインと推奨事項に従います。

- AWS 管理ポリシーの使用を開始する – MediaStore の使用をすばやく開始するには、AWS 管理ポリシーを使用して、従業員に必要なアクセス許可を付与します。これらのポリシーはアカウントですでに有効になっており、AWS によって管理および更新されています。詳細については、『IAM ユーザーガイド』の「AWS 管理ポリシーを使用したアクセス許可の使用開始」を参照してください。
- 最小権限を付与する – カスタムポリシーを作成するときは、タスクの実行に必要なアクセス許可のみを付与します。最小限のアクセス権限から開始し、必要に応じて追加のアクセス権限を付与します。この方法は、寛容なアクセス権限で始め、後でそれらを強化しようとするよりも安全です。詳細については、「最小権限を付与する」 (IAM ユーザーガイド) を参照してください。
- 機密性の高いオペレーションのために MFA を有効にする – 追加のセキュリティとして、機密性の高いリソースや API オペレーションにアクセスする際に Multi-Factor Authentication (MFA) を使用することを IAM ユーザーに要求します。詳細については、『IAM ユーザーガイド』の「AWS のデバイスに多要素認証 (MFA) を使用」を参照してください。
- 追加のセキュリティとしてポリシー条件を使用する – 実行可能な範囲内で、アイデンティティベースのポリシーでリソースへのアクセスを許可する条件を定義します。たとえば、要求が発生しなければならない許容 IP アドレスの範囲を指定するための条件を記述できます。指定された日付または時間範囲内でのみリクエストを許可する条件を書くことも、SSL や MFA の使用を要求することもできます。ポリシー要素の詳細については、『IAM ユーザーガイド』の「IAM JSON ポリシー要素: 条件」を参照してください。

MediaStore コンソールを使用する

AWS Elemental MediaStore コンソールにアクセスするには、一連の最小限のアクセス許可が必要です。これらのアクセス許可により、AWS アカウントの MediaStore リソースの詳細をリストおよび表示できます。最小限必要なアクセス許可よりも制限されたアイデンティティベースのポリシーを作成すると、そのポリシーをアタッチしたエンティティ (IAM ユーザーまたはロール) に対してはコンソールが意図したとおりに機能しません。

これらのエンティティが MediaStore コンソールを使用できるように、エンティティに次の AWS 管理対象ポリシーもアタッチします。詳細については、次のガイドの「ユーザーへのアクセス許可の追加」を参照してくださいIAM ユーザーガイド。

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

```
        {
            "Action": [
                "mediastore:*"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "Bool": {
                    "aws:SecureTransport": "true"
                }
            }
        }
    ]
}
```

AWS CLI または AWS API のみを呼び出すユーザーには、最小限のコンソールアクセス許可を付与する必要はありません。代わりに、実行しようとしている API オペレーションに一致するアクションのみへのアクセスが許可されます。

自分のアクセス許可の表示をユーザーに許可する

この例では、ユーザー ID にアタッチされたインラインおよび管理ポリシーの表示を IAM ユーザーに許可するポリシーを作成する方法を示します。このポリシーには、コンソールで、または AWS CLI か AWS API を使用してプログラム的に、このアクションを完了するアクセス許可が含まれています。

```
{
    "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": "*"
        }
    ]
}
```

MediaStore コンテナへのアクセス

この例では、 IAM AWSコンテナの 1 つ、 へのアクセスを アカウントの MediaStoreユーザーに許可`AwardsShow`します。 また、オブジェクトの管理や表示をユーザーに許可する必要もあります。

このポリシーには以下の 3 つのステートメントがあります。

- ListContainersInConsole は、このアカウントのすべてのコンテナの一覧を表示するためのアクセス許可を付与します。
- ReadContainerMetadata は、AwardsShow コンテナに関連付けられているメタデータを表示するためのアクセス許可を付与します。これには、コンテナに割り当てられている、コンテナへのアクセスを管理するポリシーに加えて、コンテナに格納されているオブジェクトのライフサイクルを管理するポリシーが含まれます。
- ManageContainerContents は、AwardsShow コンテナに格納されているオブジェクトを管理および表示するためのアクセス許可を付与します。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"ListContainersInConsole",
         "Effect":"Allow",
         "Action":[
            "mediastore:ListContainers"
         ],
         "Resource":"*"
      },
      {
         "Sid":"ReadContainerMetadata",
         "Effect":"Allow",
         "Action":[
            "mediastore:DescribeContainer",
            "mediastore:GetContainerPolicy",
            "mediastore:GetCorsPolicy",
            "mediastore:GetLifecyclePolicy"
         ],
         "Resource":"arn:aws:mediastore:*:111122223333:container/AwardsShow"
      },
      {
         "Sid":"ManageContainerContents",
         "Effect":"Allow",
         "Action":[
            "mediastore:ListItems",
            "mediastore:GetObject",
            "mediastore:PutObject",
            "mediastore:DescribeObject",
            "mediastore:DeleteObject"
         ],
         "Resource":"arn:aws:mediastore:*:111122223333:container/AwardsShow/*",
         "Condition": {
                "Bool": {
                    "aws:SecureTransport": "true"
                }
            }
      }
   ]
}
```

AWS Elemental MediaStore リソースベースのポリシーの例

AWS Elemental MediaStore コンソールにアクセスするには、AWS アカウントの MediaStore リソースに関する詳細を表示して確認するための最小限のアクセス許可が必要です。このセクションのIAMポリ

シーは、 のリソースに対する特定のアクションを許可するポリシーの例を示していますAWS Elemental MediaStore。

Note

MediaStoreは、コンテナでアクションを実行できるプリンシパルエンティティ (アカウント、ユーザー、ロール、フェデレーティッドユーザー) を定義するコンテナポリシーもサポートしています。詳細については、コンテナポリシー (p. 14) を参照してください。

すべてのMediaStoreリソースへの読み取りアクセスを許可する

AWS Elemental MediaStoreコンソールにアクセスするには、 MediaStore アカウントのAWSリソースに対して実行できるアクションを定義するポリシーが必要です。以下のIAMポリシーでは、以下のアクセス権限が付与されます。

- `mediastore:List*`および `mediastore:Describe*` アクションのセクションでは、 で作成したすべてのリソースへの読み取り専用アクセスが許可MediaStoreされます。
- `cloudwatch:GetMetricData`アクションのセクションでは、 サービスが からメトリクスを取得できますAmazon CloudWatch。ポリシーのこの部分は必須です。
- `iam:PassRole`アクションのセクションでは、 が にIAMロールを渡すのを許可します。これにより、は、サービスに代わってロールを引き受けるために と通信できるようになります。MediaStoreIAMこれにより、サービスは後でロールを引き受け、ユーザーに代わってアクションを実行できます。ポリシーのこの部分は必須です。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "mediastore:List*",
                "mediastore:Describe*"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "cloudwatch:GetMetricData"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "iam:PassRole"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

すべてのMediaStoreリソースですべてのアクションを許可する

のすべてのユーザーには、 MediaStore リソースに対するアクセス権限を定義するポリシーが必要です。MediaStore以下のIAMポリシーでは、以下のアクセス権限が付与されます。

- `mediastore:*`アクションのセクションでは、 で作成するすべてのリソースに対するすべてのアクションを許可しますMediaStore。

- cloudwatch:GetMetricDataアクションのセクションでは、 サービスが からメトリクスを取得できますAmazon CloudWatch。ポリシーのこの部分は必須です。
- iam:PassRoleアクションのセクションでは、 が にIAMロールを渡すのを許可します。これにより、は、サービスに代わってロールを引き受けるために、 と通信できるようになります。MediaStoreIAM これにより、サービスは後でロールを引き受け、ユーザーに代わってアクションを実行できます。ポリシーのこの部分は必須です。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "mediastore:*"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "cloudwatch:GetMetricData"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "iam:PassRole"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

MediaStoreリソースに対するタグに基づいてアクションを許可または拒否する

リソースのタグに基づいてアクセスを許可するには、タグベースのアクセスポリシーを使用します。以下のIAMポリシーでは、以下のアクセス権限が付与されます。

- キーPutObjectと値でタグ付けされたすべてのリソースでDeleteObjectアクションcompanyを許可するITW
- キーPutObjectと値でタグ付けされたすべてのリソースでDeleteObjectアクションenvironmentを許可するITW-prod

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Sid": "AllowPutDeleteForITW",
        "Effect": "Allow",
        "Action": [
            "mediastore:PutObject",
            "mediastore:DeleteObject"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:ResourceTag/company": "ITW",
```

```
                "aws:ResourceTag/environment": "ITW-prod"
            }
        }
    }
}
```

これらの状況でアクセスを拒否するポリシーを作成するには、ポリシーのアクセス許可行を次のように変更します。

```
"Effect": "Deny",
```

AWS Elemental MediaStore Identity and Access のトラブルシューティング

以下の情報は、MediaStore と IAM の使用に伴って発生する可能性がある一般的な問題の診断や修復に役立ちます。

MediaStore でアクションを実行する権限がない

AWS マネジメントコンソール から、アクションを実行する権限がないと通知された場合は、管理者に問い合わせてサポートを依頼してください。お客様のユーザー名とパスワードを発行したのが、担当の管理者です。

以下の例のエラーは、`mateojackson` IAM ユーザーがコンソールを使用して、コンテナに関する詳細を表示しようとしているが、`mediastore:GetContainer` アクセス許可がない場合に発生します。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 mediastore:GetContainer on resource: exampleContainer
```

この場合、Mateo は管理者に依頼し、`mediastore:GetContainer` アクションを使用して `exampleContainer` リソースにアクセスできるようにポリシーを更新してもらいます。

次のことを実行する権限がない: iam:PassRole

`iam:PassRole` アクションを実行する権限がないというエラーが表示された場合、管理者に問い合わせ、サポートを依頼する必要があります。お客様のユーザー名とパスワードを発行したのが、担当の管理者です。MediaStore にロールを渡すことができるようにポリシーを更新するよう、管理者に依頼します。

一部の AWS サービスでは、新しいサービスロールまたはサービスにリンクされたロールを作成せずに、既存のロールをサービスに渡すことができます。そのためには、サービスにロールを渡すアクセス許可が必要です。

以下の例のエラーは、`marymajor` という IAM ユーザーがコンソールを使用して MediaStore でアクションを実行しようする場合に発生します。ただし、アクションでは、サービスロールによって付与されたアクセス許可がサービスにある必要があります。メアリーには、ロールをサービスに渡すアクセス許可がありません。

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

この場合、メアリーは担当の管理者に `iam:PassRole` アクションを実行できるようにポリシーの更新を依頼します。

アクセスキーを表示する場合

IAM ユーザーアクセスキーを作成した後は、いつでもアクセスキー ID を表示できます。ただし、シークレットアクセスキーをもう一度表示することはできません。シークレットアクセスキーを紛失した場合は、新しいキーペアを作成する必要があります。

アクセスキーは、アクセスキー ID (例: `AKIAIOSFODNN7EXAMPLE`) とシークレットアクセスキー (例: `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`) の 2 つの部分から構成されます。ユーザー名とパスワードと同様に、リクエストを認証するために、アクセスキー ID とシークレットアクセスキーの両方を使用する必要があります。ユーザー名とパスワードと同様に、アクセスキーをしっかり管理してください。

> Important
>
> 正規ユーザー ID を確認するためであっても、アクセスキーをサードパーティーに提供しないでください。提供すると、第三者がアカウントへの永続的アクセスを取得する場合があります。

アクセスキーペアを作成する場合、アクセスキー ID とシークレットアクセスキーを安全な場所に保存するように求めるプロンプトが表示されます。このシークレットアクセスキーは、作成時にのみ使用できます。シークレットアクセスキーを紛失した場合、新しいアクセスキーを IAM ユーザーに追加する必要があります。最大 2 つのアクセスキーを持つことができます。すでに 2 つある場合は、新しいキーペアを作成する前に、いずれかを削除する必要があります。手順を確認するには、IAM ユーザーガイドの「アクセスキーの管理」を参照してください。

管理者として MediaStore へのアクセスを他のユーザーに許可したい

MediaStore へのアクセスを他のユーザーに許可するには、アクセスを必要とする人またはアプリケーションの IAM エンティティ (ユーザーまたはロール) を作成する必要があります。ユーザーは、このエンティティの認証情報を使用して AWS にアクセスします。次に、MediaStore の適切なアクセス許可を付与するポリシーを、そのエンティティにアタッチする必要があります。

すぐに開始するには、IAM ユーザーガイドの「IAM が委任した最初のユーザーおよびグループの作成」を参照してください 。

自分の AWS アカウント以外のユーザーに MediaStore リソースへのアクセスを許可したい

他のアカウントのユーザーや組織外のユーザーが、リソースへのアクセスに使用できるロールを作成できます。ロールを引き受けるように信頼されたユーザーを指定することができます。リソースベースのポリシーまたはアクセスコントロールリスト (ACL) をサポートするサービスの場合、それらのポリシーを使用して、リソースへのアクセスを付与できます。

詳細については、以下を参照してください。

- MediaStore でこれらの機能がサポートされるかどうかを確認するには、「AWS Elemental MediaStore と IAM の連携 (p. 60)」を参照してください。
- すべての所有してる AWS アカウントのリソースに対するアクセスを許可する方法については、IAM ユーザーガイドの「所有している別の AWS アカウントへのアクセス権を IAM ユーザーに提供」を参照してください。
- サードパーティーの AWS アカウントに対して自分のリソースへのアクセスを許可する方法については、IAM ユーザーガイド の「サードパーティーが所有する AWS アカウントへのアクセスを許可する」を参照してください。
- ID フェデレーションを介してアクセスを許可する方法については、IAM ユーザーガイドの「外部で認証されたユーザー (ID フェデレーション) へのアクセスの許可」を参照してください。

- クロスアカウントアクセスでのロールとリソースベースのポリシーの使用の違いについては、IAM ユーザーガイドの「IAM ロールとリソースベースのポリシーとの相違点」を参照してください。

AWS Elemental MediaStore でのログ記録とモニタリング

このセクションでは、セキュリティ上の目的で AWS Elemental MediaStore 内でログ記録およびモニタリングを行うためのオプションについての概要を説明します。MediaStore でのログ記録およびモニタリングの詳細については、「AWS Elemental MediaStore でのモニタリングとタグ付け (p. 72)」を参照してください。

モニタリングは、AWS Elemental MediaStore および AWS ソリューションの信頼性、可用性、パフォーマンスを維持するうえで重要な要素です。マルチポイント障害が発生した場合は、その障害をより簡単にデバッグできるように、AWS ソリューションのすべての部分からモニタリングデータを収集する必要があります。AWS には、MediaStore リソースをモニタリングし、潜在的なインシデントに対応するための複数のツールが用意されています。

Amazon CloudWatch アラーム

CloudWatch アラームを使用して、指定した期間中、1 つのメトリクスをモニタリングします。メトリクスが指定されたしきい値を超える場合、Amazon SNS トピックあるいは AWS Auto Scaling ポリシーに通知が送信されます。CloudWatch アラームは特定の状態にあるため、アクションを呼び出しません。その代わり、状態が変更され、指定期間にわたって維持される必要があります。詳細については、CloudWatch によるモニタリング (p. 75) を参照してください。

AWS CloudTrail ログ

CloudTrail は、AWS Elemental MediaStore のユーザー、ロール、または AWS のサービスによって実行されたアクションのレコードを提供します。CloudTrail によって収集された情報を使用して、MediaStore に対して行われたリクエスト、リクエスト元の IP アドレス、リクエスト者、リクエスト日時などの詳細を確認できます。詳細については、CloudTrail による API コールのログ記録 (p. 72) を参照してください。

AWS Trusted Advisor

Trusted Advisor は、AWS の数十万のお客様にサービスを提供することにより得られた運用実績から学んだベストプラクティスを活用しています。Trusted Advisor はお客様の AWS 環境を検査し、システムの可用性とパフォーマンスを向上させたりセキュリティギャップを埋める機会がある場合には、推奨事項を作成します。AWS のすべてのお客様は、5 つの Trusted Advisor チェックにアクセスできます。ビジネスまたはエンタープライズサポートプランをご利用のお客様は、すべての Trusted Advisor チェックを表示できます。

詳細については、AWS Trusted Advisor を参照してください。

AWS Elemental MediaStore のコンプライアンス検証

AWS Elemental MediaStore は AWS コンプライアンスプログラムの対象範囲ではありません。

特定のコンプライアンスプログラムの範囲内の AWS サービスのリストについては、「コンプライアンスプログラムによる AWS 対象範囲内のサービス」を参照してください。一般的な情報については、「AWS コンプライアンスプログラム」を参照してください。

サードパーティーの監査レポートをダウンロードするには、AWS Artifact を使用します。詳細については、「AWS Artifact のレポートのダウンロード」を参照してください。

MediaStore を使用する際のお客様のコンプライアンス責任は、お客様のデータの機密性や貴社のコンプライアンス目的、適用可能な法律および規制によって決定されます。AWS では以下のリソースを提供しています。

- セキュリティおよびコンプライアンスのクイックスタートガイド – これらのデプロイメントガイドでは、アーキテクチャー上の考慮事項について説明し、セキュリティとコンプライアンスに重点を置いたベースライン環境を AWS にデプロイするための手順を説明します。
- HIPAA のセキュリティとコンプライアンスに関するホワイトペーパーを作成する – このホワイトペーパーでは、企業が AWS を使用して HIPAA 準拠のアプリケーションを作成する方法について説明します。
- AWS コンプライアンスのリソース – このワークブックとガイドのコレクションは、お客様の業界や場所に適用される場合があります。
- AWS Config – この AWS サービスでは、自社プラクティス、業界ガイドライン、および規制に対するリソースの設定の準拠状態を評価します。
- AWS Security Hub – この AWS サービスでは、AWS 内のセキュリティ状態を包括的に表示しており、セキュリティ業界の標準およびベストプラクティスへのコンプライアンスを確認するのに役立ちます。

AWS Elemental MediaStore での耐障害性

AWSグローバルインフラストラクチャはAWS、リージョンとアベイラビリティーゾーンを中心として構築されます。AWSリージョンには、低レイテンシー、高いスループット、そして高度の冗長ネットワークで接続されている複数の物理的に独立・隔離されたアベイラビリティーゾーンがあります。アベイラビリティーゾーンでは、アベイラビリティーゾーン間で中断することなく自動的にフェイルオーバーするアプリケーションとデータベースを設計および運用することができます。アベイラビリティーゾーンは、従来の単一または複数のデータセンターインフラストラクチャよりも可用性、耐障害性、および拡張性が優れています。

AWSのリージョンやアベイラビリティーゾーンの詳細については、AWSグローバルインフラストラクチャを参照してください。

AWS Elemental MediaStore でのインフラストラクチャセキュリティ

マネージド型サービスとしての AWS Elemental MediaStoreは、 に説明されているAWSグローバルネットワークセキュリティの手順で保護されていますAmazon Web Services。セキュリティプロセスの概要ホワイトペーパー。

AWS が公開している API コールを使用して、ネットワーク経由で MediaStore にアクセスします。クライアントで Transport Layer Security (TLS) 1.0 以降がサポートされている必要があります。TLS 1.2 以降が推奨されています。また、Ephemeral Diffie-Hellman (DHE) や Elliptic Curve Ephemeral Diffie-Hellman (ECDHE) などの Perfect Forward Secrecy (PFS) を使用した暗号スイートもクライアントでサポートされている必要があります。これらのモードは、Java 7 以降など、最近のほとんどのシステムでサポートされています。

また、リクエストは、アクセスキー ID と、IAM プリンシパルに関連付けられているシークレットのアクセスキーを使用して署名する必要があります。または、AWS Security Token Service (AWS STS) を使用して、一時的なセキュリティ認証情報を生成し、リクエストに署名することもできます。

AWS Elemental MediaStore でのモニタリングとタグ付け

モニタリングは、AWS Elemental MediaStore およびその他の AWS ソリューションの信頼性、可用性、およびパフォーマンスを維持する上で重要な部分です。AWS には、MediaStore を監視したり、問題が発生したときに報告したり、必要に応じて自動アクションを実行したりするために以下のモニタリングツールが用意されています。

- AWS CloudTrail は、AWS アカウントにより、またはそのアカウントに代わって行われた、API 呼び出しおよび関連イベントを取得し、指定した Amazon S3 バケットにログファイルを配信します。AWS を呼び出したユーザーとアカウント、呼び出し元のソース IP アドレス、および呼び出しの発生日時を特定できます。詳細については、AWS CloudTrail User Guide を参照してください。
- Amazon CloudWatch は、AWS のリソースおよび AWS で実行しているアプリケーションをリアルタイムでモニタリングします。メトリクスの収集と追跡、カスタマイズしたダッシュボードの作成、および指定したメトリクスが指定したしきい値に達したときに通知またはアクションを実行するアラームの設定を行うことができます。たとえば、CloudWatch で Amazon EC2 インスタンスの CPU 使用率などのメトリクスを追跡し、必要に応じて新しいインスタンスを自動的に起動することができます。詳細については、Amazon CloudWatch ユーザーガイド を参照してください。
- Amazon CloudWatch Events は、AWS リソースの変更を示すシステムイベントをほぼリアルタイムのストリームとして提供します。CloudWatch イベント で自動イベント駆動型コンピューティングを有効にすると、特定のイベントを監視するルールを記述し、これらのイベントが発生したときに AWS の他のサービスで自動アクションをトリガーできます。詳細については、Amazon CloudWatch Events ユーザーガイド を参照してください。
- Amazon CloudWatch Logs を使用して、Amazon EC2 インスタンス、CloudTrail、その他のソースのログファイルを監視、保存し、それらのファイルにアクセスできます。CloudWatch Logs は、ログファイル内の情報を監視し、特定のしきい値が満たされたときに通知します。また、耐久性の高いストレージにログデータをアーカイブすることもできます。詳細については、Amazon CloudWatch Logs User Guide を参照してください。

また、MediaStore コンテナにメタデータをタグ形式で割り当てることもできます。各タグは、お客様が定義するキーと値で構成されるラベルです。タグを使用することで、リソースの管理、検索、フィルタリングを行うことができます。タグを使用して、AWS マネジメントコンソールでの AWS リソースの整理、すべての AWS リソース間での利用状況レポートおよび請求レポートの作成、インフラストラクチャの自動化アクティビティ中のリソースのフィルタリングを行うことができます。

トピック

- AWS CloudTrail による AWS Elemental MediaStore API コールのログ記録 (p. 72)
- Amazon CloudWatch による AWS Elemental MediaStore のモニタリング (p. 75)
- AWS Elemental MediaStore リソースのタグ付け (p. 86)

AWS CloudTrail による AWS Elemental MediaStore API コールのログ記録

AWS Elemental MediaStore は AWS CloudTrail と統合されています。このサービスは、MediaStore 内でユーザーやロール、または AWS のサービスによって実行されたアクションを記録するサービスです。CloudTrail は、MediaStore コンソールとコードからの MediaStore API コールなど、MediaStore の

API コールのみをイベントとしてキャプチャします。証跡を作成する場合は、Amazon S3 バケットへの CloudTrail イベント (MediaStore のイベントなど) の継続的デリバリーを有効にすることができます。証跡を設定しない場合でも、CloudTrail コンソールの [Event history (イベント履歴)] で最新のイベントを表示できます。CloudTrail によって収集された情報を使用すると、MediaStore に対してどのようなリクエストが行われたかを判断することができます。リクエストの作成元の IP アドレス、リクエストの実行者、リクエストの実行日時などの詳細が得られます。

CloudTrail の詳細 (設定して有効にする方法など) については、AWS CloudTrail User Guide を参照してください。

トピック

- CloudTrail 内の AWS Elemental MediaStore 情報 (p. 73)
- 例: AWS Elemental MediaStore ログファイルのエントリ (p. 74)

CloudTrail 内の AWS Elemental MediaStore 情報

CloudTrail は、アカウント作成時に AWS アカウントで有効になります。AWS Elemental MediaStore でサポートされるイベントアクティビティが発生すると、そのアクティビティは CloudTrail イベントとして AWS のサービスの他のイベントと共に [Event history (イベント履歴)] に記録されます。最近のイベントは、AWS アカウントで表示、検索、ダウンロードできます。詳細については、「CloudTrail イベント履歴でのイベントの表示」を参照してください。

MediaStoreのイベントなど、AWS アカウントのイベントの継続的な記録については、証跡を作成します。証跡により、CloudTrail はログファイルを Amazon S3 バケットに配信できます。デフォルトでは、コンソールで証跡を作成するときに、証跡がすべての AWS リージョンに適用されます。証跡では、AWS パーティションのすべてのリージョンからのイベントがログに記録され、指定した Amazon S3 バケットにログファイルが配信されます。さらに、より詳細な分析と CloudTrail ログで収集されたデータに基づいた行動のためにその他の AWS サービスを設定できます。詳細については、以下のトピックを参照してください。

- 証跡の作成に関する概要
- CloudTrail でサポートされるサービスと統合
- の Amazon SNS 通知の設定 CloudTrail
- 複数のリージョンから CloudTrail ログファイルを受け取ると複数のアカウントから CloudTrail ログファイルを受け取る

AWS Elemental MediaStore は CloudTrail ログファイルのイベントとして次のオペレーションを記録します。

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

各イベントまたはログエントリには、リクエストの生成者に関する情報が含まれます。この ID 情報は以下のことを確認するのに役立ちます。

- リクエストが、ルートと IAM ユーザー認証情報のどちらを使用して送信されたか
- リクエストが、ロールとフェデレーティッドユーザーのどちらの一時的なセキュリティ認証情報を使用して送信されたか
- リクエストが、別の AWS サービスによって送信されたかどうか

詳細については、「CloudTrail userIdentity 要素」を参照してください。

例: AWS Elemental MediaStore ログファイルのエントリ

証跡は、指定した Amazon S3 バケットにイベントをログファイルとして配信できる設定です。CloudTrail ログファイルには、1 つ以上のログエントリが含まれます。イベントは任意の送信元からの単一のリクエストを表し、リクエストされたアクション、アクションの日時、リクエストのパラメーターなどに関する情報が含まれます。CloudTrail ログファイルは、パブリック API 呼び出しの順序付けられたスタックトレースではないため、特定の順序では表示されません。

以下の例は、`CreateContainer` 操作を表す CloudTrail ログエントリを示しています。


```
{
    "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"
}
```

Amazon CloudWatch による AWS Elemental MediaStore のモニタリング

CloudWatch を使用して AWS Elemental MediaStore をモニタリングすることで、生データを収集し、ほぼリアルタイムの読み取り可能なメトリクスに加工できます。これらの統計は 15 か月間保持されるため、履歴情報にアクセスしてウェブアプリケーションやサービスの動作をより的確に把握できます。また、特定のしきい値を監視するアラームを設定し、これらのしきい値に達したときに通知を送信したりアクションを実行したりできます。詳細については、Amazon CloudWatch ユーザーガイド を参照してください。

AWS には、MediaStore を監視して異常を検出した場合にレポートし、必要に応じて自動的に対処するために、以下のモニタリングツールが用意されています。

- Amazon CloudWatch Logs を使用すると、AWS Elemental MediaStore などの AWS サービスのログファイルをモニタリング、保存し、このログファイルにアクセスすることができます。CloudWatch Logs を使用すると、ログデータを使用してアプリケーションやシステムをモニタリングできます。たとえば、CloudWatch Logs はアプリケーションログに記録されたエラーの数をトラッキングし、エラー率が指定のしきい値を超えたときに管理者に通知を送ることができます。お客様のログデータが CloudWatch Logs によるモニタリングに使用されるので、コードの変更は不要です。たとえば、特定のリテラル用語 (「ValidationException」など) がアプリケーションログに含まれていないかを監視したり、特定の期間中に行われた `PutObject` リクエストの数をカウントしたりすることができます。検索した語句が見つかると、CloudWatch Logs は指定された CloudWatch メトリクスにデータをレポートします。ログデータは、転送時や保管時に暗号化されます。
- Amazon CloudWatch Events は、MediaStore オブジェクトなどの AWS リソースの変更を示すシステムイベントを配信します。イベント (`DeleteObject` リクエストなど) に一致するルールを設定し、1 つ以上のターゲット関数またはストリームにイベントを振り分けることができます。オペレーションの変更は、変更時に CloudWatch イベント で即座に認識されます。さらに、CloudWatch イベント はこれらのオペレーションの変更に応答し、必要に応じて、応答メッセージを環境に送り、機能をアクティブ化し、変更を行い、状態情報を収集することによって、修正アクションを実行します。

CloudWatch Logs

アクセスのログ記録には、コンテナ内でオブジェクトに対して行われたリクエストの詳細が記録されます。アクセスログは、セキュリティやアクセスの監査などの多くのアプリケーションに役立ちます。また、顧客基盤について知り、MediaStore の請求を理解することにも役立ちます。CloudWatch Logs は、次のように分類されます。

- ログストリームは、同じソースを共有する一連のログイベントです。
- ロググループは、保持、モニタリング、アクセス制御について同じ設定を共有するログストリームのグループです。コンテナでアクセスのログ記録を有効にすると、 MediaStoreは などの名前でロググループを作成します`/aws/mediastore/MyContainerName`。 ロググループを定義して、各グループに入れるストリームを指定することができます。1 つのロググループに属することができるログストリームの数にクォータはありません。

デフォルトでは、ログは無制限に保持され、失効しません。ロググループごとに保持ポリシーを調整し、無制限の保持期間を維持するか、1 日間 ~ 10 年間の保持期間を選択することができます。

Amazon CloudWatch のアクセス許可のセットアップ

AWS Identity and Access Management (IAM) を使用して、Amazon CloudWatch へのアクセスを AWS Elemental MediaStore に許可するロールを作成します。以下の手順は、お客様のアカウント用に発行される CloudWatch Logs に対して実行する必要があります。CloudWatch によってお客様のアカウントのメトリクスが自動的に発行されます。

MediaStore に CloudWatch へのアクセスを許可するには

1. IAM コンソール (https://console.aws.amazon.com/iam/) を開きます。
2. IAM コンソールのナビゲーションペインで、[ポリシー]、[ポリシーの作成] の順に選択します。
3. [JSON] タブを選択し、以下のポリシーを貼り付けます。

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

   このポリシーは、AWS アカウント内の任意のリージョンに含まれている任意のコンテナに対して、MediaStore がロググループとログストリームを作成できるようにします。
4. [ポリシーの確認] を選択します。
5. [ポリシーの確認] ページで、[名前] に「**MediaStoreAccessLogsPolicy**」と入力し、[ポリシーの作成] を選択します。
6. IAM コンソールのナビゲーションペインで、[ロール]、[ロールの作成] の順に選択します。
7. [別の AWS アカウント] ロールタイプを選択します。
8. [アカウント ID] に AWS アカウント ID を入力します。
9. [Next (次へ)] を選択します。アクセス許可。
10. 検索ボックスに「**MediaStoreAccessLogsPolicy**」と入力します。
11. 新しいポリシーの横にあるチェックボックスをオンにし、[次へ] を選択します。タグ.
12. [Next (次へ)] を選択します。新しいユーザーをプレビューするために確認します。
13. [ロール名] に「**MediaStoreAccessLogs**」と入力し、[ロールの作成] を選択します。
14. 確認メッセージで、作成したロールの名前 (**MediaStoreAccessLogs**) を選択します。
15. ロールの [概要] ページで、[Trust relationship (信頼関係)] タブを選択します。
16. [Edit trust relationship] を選択します。
17. ポリシードキュメントで、プリンシパルを MediaStore サービスに変更します。以下のようになります。

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

   ポリシー全体は以下のようになります。

```
{
```

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

18. [Update Trust Policy] を選択します。

コンテナのアクセスログ記録の有効化

デフォルトでは、AWS Elemental MediaStore によってアクセスログは収集されません。コンテナでアクセスのログ記録を有効にすると、MediaStore ではそのコンテナに保存されているオブジェクトのアクセスログを Amazon CloudWatch に配信します。アクセスログには、コンテナに保存されているすべてのオブジェクトに対して行われたリクエストの詳細が記録されます。この情報には、リクエストタイプ、リクエストで指定したリソース、リクエストを処理した日時などが含まれます。

> Important
>
> MediaStore コンテナでアクセスのログ記録を有効にしても追加料金はかかりません。ただし、サービスが配信するいずれのログファイルの格納に対しても通常の料金がかかります (ログファイルはいつでも削除できます)。AWS のログファイルの配信に対してはデータ転送料金はかかりませんが、ログファイルへのアクセスに対しては通常のデータ転送料金がかかります。

アクセスのログ記録を有効にするには (AWS CLI)

- AWS CLI で、`start-access-logging` コマンドを使用します。

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

このコマンドの戻り値はありません。

コンテナのアクセスログ記録の無効化

コンテナでアクセスのログ記録を無効にすると、AWS Elemental MediaStore が Amazon CloudWatch へのアクセスログの送信を停止します。これらのアクセスログは保存されないため、取り出せなくなります。

アクセスのログ記録を無効にするには (AWS CLI)

- AWS CLI で、`stop-access-logging` コマンドを使用します。

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

このコマンドの戻り値はありません。

AWS Elemental MediaStore でのアクセスのログ記録のトラブルシューティング

AWS Elemental MediaStore のアクセスログが Amazon CloudWatch に表示されない場合は、考えられる原因と解決策について以下の表を参照してください。

Note

トラブルシューティングプロセスに役立つ AWS CloudTrail ログを有効にしてください。

症状	考えられる原因	解決策
CloudTrail ログが有効になっていても、CloudTrail イベントが全く表示されない。	IAM ロールが存在しないか、正しくない名前、アクセス許可、または信頼ポリシーが含まれています。	正しい名前、アクセス許可、信頼ポリシーを使用してロールを作成します。「the section called “CloudWatch のアクセス許可のセットアップ” (p. 75)」を参照してください。
`DescribeContainer`API リクエストを送信したが、`AccessLoggingEnabled`パラメータに の値が含まれていることがレスポンスに示されている`False`。 さらに、 CloudTrail ロールが `MediaStoreAccessLogs` 、`DescribeLogGroup` 、 、または`CreateLogGroup`の呼び出しに成功した`DescribeLogStream`ことを示すイベントが表示されません。`CreateLogStream`	IAM ロールが存在しないか、正しくない名前、アクセス許可、または信頼ポリシーが含まれています。	正しい名前、アクセス許可、信頼ポリシーを使用してロールを作成します。「the section called “CloudWatch のアクセス許可のセットアップ” (p. 75)」を参照してください。
	アクセスのログ記録がコンテナで有効になっていません。	コンテナのアクセスログを有効にします。「the section called “アクセスログ記録の有効化” (p. 77)」を参照してください。

CloudTrail コンソールに、`MediaStoreAccessLogs` ロールに関連するアクセス拒否エラーを含むイベントが表示されている。CloudTrail イベントには、次のような行が含まれている場合があります。
 	
`"eventSource": "logs.amazonaws.com",`
 	
`"errorCode": "AccessDenied",`
 	
`"errorMessage": "User: arn:aws:sts::111122223333:assumed-role/MediaStoreAccessLogs/MediaStoreAccessLogsSession is not authorized to perform: logs:DescribeLogGroups on resource: arn:aws:logs:us-west-2:111122223333:log-group::log-stream:",`
	IAM ロールに AWS Elemental MediaStore に対する正しいアクセス許可がありません 。	適切なアクセス許可と信頼ポリシーを持つように IAM ロールを更新します。「the section called “CloudWatch のアクセス許可のセットアップ” (p. 75)」を参照してください。

1 つのコンテナまたは複数のコンテナのログが表示されない。	お客様のアカウントが、1 アカウント、1 リージョンあたりのロググループに対する CloudWatch のクォータを超えている可能性があります。Amazon CloudWatch Logs User Guide でロググループ	CloudWatch コンソールで、アカウントがロググループの CloudWatch のクォータに達しているかどうかを判断します。必要に応じて、クォータの引き上げをリクエストします。

症状	考えられる原因	解決策
	のクォータを参照してください。	
CloudWatch に、予期しているすべてのログではなく、一部のログしか表示されない。	お客様のアカウントが、1 秒、1 アカウント、1 リージョンあたりの CloudWatch のクォータを超えている可能性があります。Amazon CloudWatch Logs User Guide で `PutLogEvents` のクォータを参照してください。	1 秒、1 アカウント、1 リージョンあたりの CloudWatch トランザクションのクォータの引き上げをリクエストします。

アクセスログの形式

アクセスログファイルは、一連の JSON 形式のログレコードで構成されており、各ログレコードは 1 つのリクエストを表します。ログ内のフィールドの順序は変わることがあります。2 つのログレコードで構成されるログの例を次に示します。

```
{
  "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
}
```

次のリストは、ログレコードのフィールドについて説明しています。

AWSAccountId

リクエストを行うために使用されたアカウントの AWS アカウント ID。

BytesReceived

MediaStore サーバーが受信するリクエストボディのバイト数。

BytesSent

MediaStore サーバーが送信するレスポンス本文のバイト数。この値は、多くの場合、サーバーレスポンスに含まれている `Content-Length` ヘッダーの値と同じです。

ContainerName

リクエストを受信したコンテナの名前。

ErrorCode

MediaStore のエラーコード (`InternalServerError` など)。エラーが発生しなかった場合は、`-` の文字が表示されます。ステータスコードが 200 (接続が閉じられているか、サーバーがレスポンスのストリーミングを開始した後にエラーが発生したことを示す) であってもエラーコードが表示される場合があります。

ExpiresAt

オブジェクトの有効期限の日時。この値は、コンテナに適用されるライフサイクルポリシーtransient data ruleで によって設定された有効期限に基づきます。この値は `ISO-8601` の日時で、リクエストに対応したホストのシステムクロックに基づいています。ライフサイクルポリシーにオブジェクトに適用される一時的なデータルールがない場合、またはコンテナに適用されるライフサイクルポリシーがない場合、このフィールドの値は ですnull。 このフィールドは、次のオペレーションにのみ適用されます。`PutObject`、 `GetObject` 、 `DescribeObject` 、および `DeleteObject`。

HTTPStatus

レスポンスの HTTP ステータスの数値。

オペレーション

`PutObject` や `ListItems` などの、実行されたオペレーション。

パス

オブジェクトが保存されているコンテナ内のパス。オペレーションがパスのパラメータを使用しない場合は、`-` の文字が表示されます。

ReceivedTime

リクエストを受け取った時刻。この値は `ISO-8601` の日時で、リクエストに対応したホストのシステムクロックに基づいています。

リクエスタ

リクエストを行うために使用されたアカウントのユーザーの Amazon リソースネーム (ARN)。認証されていないリクエストの場合、この値は になりますanonymous。 認証が完了する前にリクエストが失敗した場合は、このフィールドがログに表示されない可能性があります。このようなリクエストでは、`ErrorCode` で認可の問題を特定できる場合があります。

RequestID

各リクエストを一意に識別するために AWS Elemental MediaStore で生成される文字列。

ソース

呼び出しを行った AWS サービスのリクエスタまたはサービスプリンシパルの表面上のインターネットアドレス。中間プロキシやファイアウォールにより、リクエストを作成したマシンのアドレスが不明確になる場合、値は null に設定されます。

TotalTime

サーバーから見た、リクエストの転送中の時間数 (ミリ秒単位)。これは、サービスがリクエストを受信してから、レスポンスの最終バイトが送信されるまでの時間を計測した値です。この値は、サーバーの観点から計測されます。クライアント側の観点で計測された値は、ネットワークレイテンシーの影響を受けるためです。

TurnAroundTime

MediaStore でリクエストの処理に要した時間数 (ミリ秒単位)。これは、リクエストの最終バイトが受信されてから、レスポンスの先頭バイトが送信されるまでの時間を計測した値です。

ログのフィールドの順序は変わることがあります。

ログ記録ステータスの変更が有効になるまでの期間

コンテナのログ記録ステータスの変更がログファイルの配信に反映されるまでには時間がかかります。たとえば、コンテナ A のログ記録を有効にした場合、その後数時間に行われるリクエストは記録されることもあれば、されないこともあります。コンテナ B のログ記録を無効にした場合、その後数時間はログが引き続き配信されることもあれば、されないこともあります。いずれの場合も、最終的には新しい設定が有効になるため、追加の操作は不要です。

ベストエフォート型のサーバーログ配信

アクセスログレコードの配信は、ベストエフォートで行われます。コンテナがログ記録用に適切に設定されている場合、そのコンテナへのほとんどのリクエストについてログレコードが配信されます。ほとんどのログレコードは、記録された時間から数時間以内に配信されますが、配信間隔は短くなる場合もあります。

アクセスのログ記録の完全性や適時性は保証されません。リクエストのログレコードが、リクエストが実際に処理されてからかなり後に配信されたり、配信すらされないこともあり得ます。アクセスログの目的は、コンテナに対するトラフィックの特性を理解することです。ログレコードが失われることはまれですが、すべてのリクエストが完全に報告されるとは限りません。

アクセスのログ記録機能はベストエフォート型であるため、AWS ポータルで利用できる使用状況レポート (AWS マネジメントコンソール でレポートされる請求およびコスト管理レポート) には、アクセスログに記録されていないアクセスリクエストが含まれる場合があります。

アクセスログの形式のプログラミングに関する考慮事項

新しいフィールドを追加することで、アクセスログの形式を随時拡張する場合があります。アクセスログを解析するコードは、追加のフィールドを理解できなくても処理するよう作成する必要があります。

CloudWatch イベント

Amazon CloudWatch Events を使用すると、AWS サービスを自動化して、アプリケーションの可用性の問題やリソースの変更などのシステムイベントに自動的に対応できます。AWS サービスからのイベントは、ほぼリアルタイムに CloudWatch イベント に提供されます。簡単なルールを記述して、注目するイベントと、イベントがルールに一致した場合に自動的に実行するアクションを指定できます。

ファイルをコンテナにアップロードするか、コンテナから削除すると、CloudWatch サービスで 2 つのイベントが連続して発生します。

1. the section called “オブジェクト状態の変更イベント” (p. 82)
2. the section called “コンテナ状態の変更イベント” (p. 83)

これらのイベントにサブスクライブする方法については、Amazon CloudWatch を参照してください。

自動的にトリガーできるオペレーションには、以下が含まれます。

- AWS Lambda 関数の呼び出し
- Amazon EC2 Run Command の呼び出し
- Amazon Kinesis Data Streams へのイベントの中継
- AWS Step Functions ステートマシンのアクティブ化
- Amazon SNS トピックまたは AWS SMS キューの通知

AWS Elemental MediaStore で CloudWatch イベント を使用する例をいくつか以下に示します。

- コンテナが作成されるたびに Lambda 関数をアクティブ化する
- オブジェクトが削除されたときに Amazon SNS トピックに通知する

詳細については、Amazon CloudWatch Events ユーザーガイド を参照してください。

トピック

- AWS Elemental MediaStore オブジェクト状態の変更イベント (p. 82)
- AWS Elemental MediaStore コンテナ状態の変更イベント (p. 83)

AWS Elemental MediaStore オブジェクト状態の変更イベント

このイベントは、オブジェクトの状態が変わると (オブジェクトがアップロードまたは削除されると)、発行されます。このイベントにサブスクライブする方法については、Amazon CloudWatch を参照してください。

オブジェクトの更新

```
{
  "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"
  }
}
```

オブジェクトの削除

```
{
  "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"
  }
}
```

AWS Elemental MediaStore コンテナ状態の変更イベント

このイベントは、コンテナの状態が変わると (コンテナが追加または削除されると)、発行されます。このイベントにサブスクライブする方法については、Amazon CloudWatch を参照してください。

コンテナの作成

```
{
  "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"
  }
}
```

コンテナの削除

```
{
  "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"
  }
}
```

Amazon CloudWatch メトリクスによる AWS Elemental MediaStore のモニタリング

CloudWatch を使用して AWS Elemental MediaStore をモニタリングすることで、生データを収集し、ほぼリアルタイムの読み取り可能なメトリクスに加工できます。これらの統計は 15 か月間保持されるため、履歴情報にアクセスしてウェブアプリケーションやサービスの動作をより的確に把握できます。また、特定のしきい値を監視するアラームを設定し、これらのしきい値に達したときに通知を送信したりアクションを実行したりできます。詳細については、Amazon CloudWatch ユーザーガイド を参照してください。

AWS Elemental MediaStore では、`BytesDownloaded` を確認して、メトリクスが特定のしきい値に達したときに、自分自身に E メールを送信することができます。

CloudWatch コンソールを使用してメトリクスを表示するには

メトリクスはまずサービスの名前空間ごとにグループ化され、次に各名前空間内のさまざまなディメンションの組み合わせごとにグループ化されます。

1. AWS マネジメントコンソール にサインインした後、https://console.aws.amazon.com/cloudwatch/ にある CloudWatch コンソールを開きます。
2. ナビゲーションペインで、[Metrics (メトリクス)] を選択します。
3. [All metrics (すべてのメトリクス)] で、[AWS/MediaStore] 名前空間を選択します。
4. メトリクスディメンションを選択して、メトリクスを表示します。たとえば、`Request metrics by container` を選択して、コンテナに送信されたさまざまなタイプのリクエストのメトリクスを表示します。

AWS CLI を使ってメトリクスを表示するには

- コマンドプロンプトで、次のコマンドを使用します。

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

AWS Elemental MediaStore のメトリクス

次の表に、AWS Elemental MediaStore が CloudWatch に送信するメトリクスを示します。

> **Note**
>
> メトリクスを表示するには、MediaStore が Amazon CloudWatch にメトリクスを送信することを許可するメトリクスポリシーを追加する (p.)必要があります。ポリシーをコンテナに追加する必要があります。

メトリクス	説明
`RequestCount`	MediaStore コンテナに対して行われた HTTP リクエストの総数。オペレーションタイプで分割されます (`Put`、`Get`、`Delete`、`Describe`、`List`)。 単位: カウント 有効なディメンション: • コンテナ名 • オブジェクトグループ名

メトリクス	説明
	• リクエストタイプ 有効な統計: 合計
4xxErrorCount	4xx エラーを発生させた、MediaStore に対して行われた HTTP リクエストの数。 単位: カウント 有効なディメンション: • コンテナ名 • オブジェクトグループ名 • リクエストタイプ 有効な統計: 合計
5xxErrorCount	5xx エラーヲ発生させた MediaStore に対して行われた HTTP リクエストの数。 単位: カウント 有効なディメンション: • コンテナ名 • オブジェクトグループ名 • リクエストタイプ 有効な統計: 合計
BytesUploaded	MediaStore コンテナに対して行われたリクエストに対してアップロードされたバイト数。リクエストには本文が含まれます。 単位: バイト 有効なディメンション: • コンテナ名 • オブジェクトグループ名 有効な統計: Average (リクエストあたりのバイト数)、Sum (期間あたりのバイト数)、Sample Count、Min (P0.0 と同じ)、Max (p100 と同じ)、p0.0 ~ p99.9 のパーセンタイル

メトリクス	説明
BytesDownloaded	MediaStore コンテナに対する、レスポンスに本文が含まれるリクエストに対してダウンロードしたバイト数。 単位: バイト 有効なディメンション: • コンテナ名 • オブジェクトグループ名 有効な統計: Average (リクエストあたりのバイト数)、Sum (期間あたりのバイト数)、Sample Count、Min (P0.0 と同じ)、Max (p100 と同じ)、p0.0 ~ p99.9 のパーセンタイル
TotalTime	サーバーから見た、リクエストの転送中の時間数 (ミリ秒単位)。この値は、MediaStore がリクエストを受信してから、レスポンスの最終バイトが送信するまでの時間を計測した値です。この値は、サーバーの観点から計測されます。クライアント側の観点で計測された値は、ネットワークレイテンシーの影響を受けるためです。 単位: Milliseconds 有効なディメンション: • コンテナ名 • オブジェクトグループ名 • リクエストタイプ 有効な統計: Average、Min (P0.0 と同じ)、Max (p100 と同じ)、p0.0 と p100 の間のパーセンタイル
TurnaroundTime	MediaStore でリクエストの処理に要した時間数 (ミリ秒単位)。この値は、MediaStore がリクエストの最終バイトを受信してから、レスポンスの先頭バイトが送信されるまでの時間を計測した値です。 単位: Milliseconds 有効なディメンション: • コンテナ名 • オブジェクトグループ名 • リクエストタイプ 有効な統計: Average、Min (P0.0 と同じ)、Max (p100 と同じ)、p0.0 と p100 の間のパーセンタイル

AWS Elemental MediaStore リソースのタグ付け

タグとは、ユーザーまたは AWS が AWS リソースに割り当てるカスタム属性ラベルです。各タグは 2 つの部分で構成されます。

- タグキー (例: `CostCenter`、`Environment`、または `Project`)。タグキーでは、大文字と小文字が区別されます。
- タグ値として知られるオプションのフィールド (例: `111122223333` または `Production`)。タグ値を省略すると、空の文字列を使用した場合と同じになります。タグキーとタグ値は大文字と小文字が区別されます。

タグは、以下のことに役立ちます。

- AWS リソースの特定と整理。多くの AWS のサービスではタグ付けがサポートされるため、さまざまなサービスからリソースに同じタグを割り当てて、リソースの関連を示すことができます。たとえば、AWS Elemental MediaLive 入力に割り当てるのと同じタグを AWS Elemental MediaStore *container* に割り当てることができます。
- AWS のコストの追跡。これらのタグは、AWS Billing and Cost Management ダッシュボードで有効化します。AWS はタグを使用してコストを分類し、月単位のコスト配分レポートを提供します。詳細については、「AWS Billing and Cost Management ユーザーガイド」の「コスト配分タグの使用」を参照してください。

以下のセクションでは、AWS Elemental MediaStore のタグの詳細について説明しています。

AWS Elemental MediaStore でサポートされているリソース

AWS Elemental MediaStore の次のリソースがタグ付けをサポートしています:

- *container*

タグを追加および管理する方法については、「タグを管理する (p. 88)」を参照してください。

AWS Elemental MediaStore は AWS Identity and Access Management (IAM) のタグベースのアクセスコントロール機能をサポートしていません。

タグの命名規則と使用規則

AWS Elemental MediaStore リソースでのタグの使用には、次の基本的な命名規則と使用規則が適用されます。

- 各リソースには、最大 50 個のタグを設定できます。
- タグキーは、リソースごとにそれぞれ一意である必要があります。また、各タグキーに設定できる値は 1 つのみです。
- タグキーの最大長は UTF-8 で 128 Unicode 文字です。
- タグ値の最大長は UTF-8 で 256 Unicode 文字です。
- 使用できる文字は、UTF-8 で表現可能な文字、数字、スペース、および . : + = @ _ / - (ハイフン) です。Amazon EC2 リソースには任意の文字を使用できます。
- タグのキーと値は大文字と小文字が区別されます。ベストプラクティスとして、タグを大文字にするための戦略を決定し、その戦略をすべてのリソースタイプにわたって一貫して実装します。たとえば、`Costcenter`、`costcenter`、`CostCenter` のいずれを使用するかを決定し、すべてのタグに同じ規則を使用します。大文字と小文字の扱いについて、同様のタグに整合性のない規則を使用することは避けてください。
- プレフィックス `aws:` はタグで使用することはできません。AWS 用に予約されています。このプレフィックスが含まれるタグのキーや値を編集したり削除したりすることはできません。このプレフィックスの付いたタグは、リソースあたりのタグ数のクォータにカウントされません。

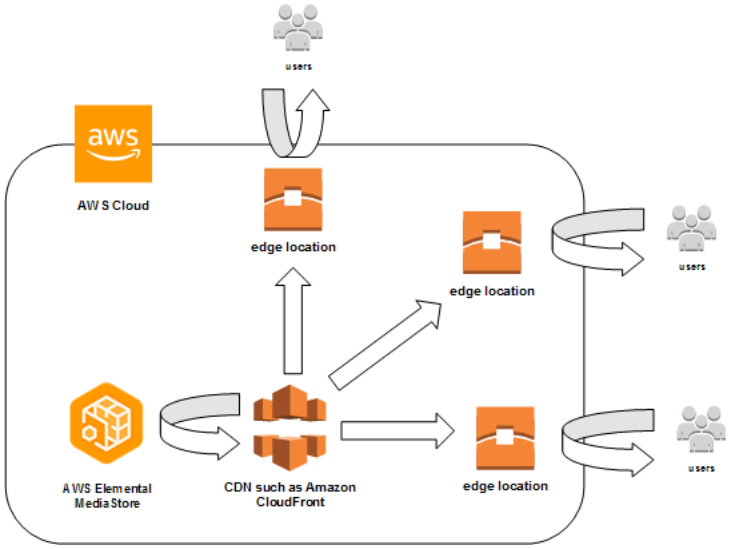
タグを管理する

タグは、リソースの `Key` および `Value` プロパティで構成されています。AWS CLI または MediaStore API を使用して、これらのプロパティの値を追加、編集、または削除できます。タグの使用については、AWS Elemental MediaStore API リファレンスの以下のセクションを参照してください。

- CreateContainer
- ListTagsForResource
- リソース
- TagResource
- UntagResource

コンテンツ配信ネットワーク (CDN) の使用

Amazon CloudFront などのコンテンツ配信ネットワーク (CDN) を使用して AWS Elemental MediaStore に保存したコンテンツを配信できます。CDN は、グローバルに分散されたサーバーのセットであり、動画などのコンテンツをキャッシュします。ユーザーがコンテンツをリクエストすると、CDN はそのリクエストを最もレイテンシーが低いエッジロケーションにルーティングします。コンテンツがこのエッジロケーションにキャッシュ済みである場合、CDN はコンテンツを直ちに配信します。コンテンツがこのエッジロケーションに現在存在しない場合、CDN は、オリジン (MediaStore コンテナなど) からそのコンテンツを取得してユーザーに配信します。



トピック

- AWS Elemental MediaStore コンテナへのアクセスを Amazon CloudFront に許可する (p. 89)
- AWS Elemental MediaStore による HTTP キャッシュの操作 (p. 90)

AWS Elemental MediaStore コンテナへのアクセスを Amazon CloudFront に許可する

Amazon CloudFront を使用して AWS Elemental MediaStore のコンテナに保存したコンテンツを配信できます。開始するには、CloudFront に読み取りアクセス権以上を付与するポリシーをコンテナにアタッチします。

コンテナへのアクセスを CloudFront に許可するには (コンソール)

1. MediaStore コンソール (https://console.aws.amazon.com/mediastore/) を開きます。
2. [Containers (コンテナ)] ページで、コンテナの名前を選択します。

コンテナの詳細ページが表示されます。

3. [Container policy (コンテナポリシー)] セクションで、Amazon CloudFront に読み取りアクセス権限以上を付与するポリシーをアタッチします。

   Note

   HTTPS 経由のパブリック読み取りアクセス (p. 17)のポリシーの例がこれらの要件に一致するのは、このポリシーが、HTTPS を介してドメインにリクエストを送信するすべての送信元からの `GetObject` コマンドと `DescribeObject` コマンドを許可するためです。

4. [Container CORS policy (コンテナの CORS ポリシー)] セクションで、適切なアクセスレベルを許可するポリシーを割り当てます。

   Note

   CORS ポリシー (p. 22)は、ブラウザベースのプレーヤーにアクセスを許可する場合にのみ必要です。

5. 以下の詳細を書き留めます。

   - コンテナに割り当てられたデータエンドポイント。この情報は [Containers (コンテナ)] ページの [Info (情報)] セクションにあります。CloudFront では、データエンドポイントはオリジンドメイン名と呼ばれます。
   - オブジェクトが保存されているコンテナのフォルダ構造。CloudFront の場合、これはオリジンパスと呼ばれます。ただし、この設定は省略可能です。オリジンパスの詳細については、Amazon CloudFront 開発者ガイドを参照してください。

6. CloudFront でディストリビューションを作成し、AWS Elemental MediaStore のコンテンツを配信するように設定します。前のステップで収集した情報が必要です。

AWS Elemental MediaStore による HTTP キャッシュの操作

AWS Elemental MediaStore は、Amazon CloudFront などのコンテンツ配信ネットワーク (CDN) によって正しく効率的にキャッシュできるようにオブジェクトを保存します。エンドユーザーまたは CDN が MediaStore からオブジェクトを取得すると、サービスはオブジェクトのキャッシュ動作に影響する HTTP ヘッダーを返します (HTTP 1.1 キャッシュ動作の標準については、RFC2616 セクション 13 を参照してください)。 これらのヘッダーは次のとおりです。

- **`ETag`** (カスタマイズ不可) - エンティティタグヘッダーは、MediaStore が送信するレスポンスの一意の識別子です。標準に準拠した CDN およびウェブブラウザは、このタグをキーとして使用し、オブジェクトをキャッシュします。MediaStore は、アップロード時に各オブジェクトに対して `ETag` を自動的に生成します。オブジェクトの詳細を表示 (p. 50)して、ETag 値を決定できます。
- **`Last-Modified`** (カスタマイズ不可) - このヘッダーの値は、オブジェクトが変更された日時を示します。MediaStore は、オブジェクトのアップロード時にこの値を自動的に生成します。
- **`Cache-Control`** (カスタマイズ可能) - このヘッダーの値は、変更されたかどうかを CDN が確認するまでオブジェクトをキャッシュする時間を制御します。CLI (p. 47) または API を使用して MediaStore コンテナにオブジェクトをアップロードするときに、このヘッダーを任意の値に設定できます。有効な値の完全なセットについては、HTTP/1.1 のドキュメントに記載されています。オブジェクトをアップロードするときにこの値を設定しない場合、MediaStore はオブジェクトの取得時にこのヘッダーを返しません。

  Cache-Control ヘッダーの一般的ユースケースは、オブジェクトをキャッシュする期間の指定です。たとえば、エンコーダーによって頻繁に上書きされるビデオマニフェストファイルがあるとします。`max-age` を 10 に設定すると、オブジェクトを 10 秒間キャッシュする必要があることを指定できます。ま

たは、上書きされないビデオセグメントが保存されているとします。このオブジェクトの `max-age` を 31536000 に設定して、約 1 年間キャッシュできます。

条件付きリクエスト

MediaStore への条件付きリクエスト

MediaStore は、条件付きリクエスト (RFC7232 で説明されている `If-Modified-Since` や `If-None-Match` などのリクエストヘッダーを使用) と無条件リクエストに同じように応答します。つまり、MediaStore が有効な `GetObject` リクエストを受信すると、クライアントがすでにオブジェクトを持っている場合でも、サービスは常にオブジェクトを返します。

CDN への条件付きリクエスト

MediaStore に代わってコンテンツを提供する CDN は、RFC7232 セクション 4.1 で説明されているように、`304 Not Modified` を返すことによって条件付きリクエストを処理できます。これは、リクエスタが条件付きリクエストに一致するオブジェクトをすでに持っているため、オブジェクトのコンテンツ全体を転送する必要がないことを示します。

CDN (および HTTP/1.1 に準拠したその他のキャッシュ) は、オリジンサーバーによって転送される `ETag` および `Cache-Control` ヘッダーに基づいてこれらの決定を行います。CDN が MediaStore オリジンサーバーに繰り返し取得されたオブジェクトの更新をクエリする頻度を制御するには、MediaStore にアップロードするときにそれらのオブジェクトの `Cache-Control` ヘッダーを設定します。

AWS Elemental MediaStore のクォータ

サービスクォータコンソールには、AWS Elemental MediaStore のクォータに関する情報が表示されます。デフォルトのクォータの表示に加えて、Service Quotas コンソールを使用して、調整可能なクォータの引き上げをリクエストできます。

以下の表では、AWS Elemental MediaStore のクォータ (旧称は制限) について説明しています。クォータは、AWS アカウントのサービスリソースまたはオペレーションの最大数です。

> Note
>
> アカウント内の個々のコンテナにクォータを割り当てるには、AWS サポートまたはアカウントマネージャーにお問い合わせください。このオプションを使用すると、アカウントレベルの制限をコンテナ間で分割し、1 つのコンテナでクォータ全体を使い切ることを防止できます。

リソースまたはオペレーション	デフォルトのクォータ	コメント
コンテナ	100	このアカウントで作成できるコンテナの最大数
DeleteObject	1 秒あたり 100 件のトランザクション (TPS)	1 秒あたりに可能なオペレーションリクエストクエストは調整されます。 クォータの引き上げをリクエストできます。
DescribeObject	1,000 TPS	1 秒あたりに可能なオペレーションリクエストクエストは調整されます。 クォータの引き上げをリクエストできます。
フォルダレベル	10	コンテナ内に作成できるフォルダレベルの最大レベルを超えて入れ子にしない限り、無制限のできます。
フォルダ	無制限	コンテナ内で 10 レベルを超えて入れ子にしないのフォルダを作成できます。
GetObject–標準アップロードの可用性	1,000 TPS	1 秒あたりに可能なオペレーションリクエストクエストは調整されます。 クォータの引き上げをリクエストできます。
GetObject–ストリーミングアップロードの可用性	25 TPS	1 秒あたりに可能なオペレーションリクエストクエストは調整されます。 クォータの引き上げをリクエストできます。
ListItems	5 TPS	1 秒あたりに可能なオペレーションリクエストクエストは調整されます。 クォータの引き上げをリクエストできます。
Object Size	25 MB	1 つのオブジェクトの最大ファイルサイズ。

リソースまたはオペレーション	デフォルトのクォータ	コメント
オブジェクト	無制限	アカウントのフォルダまたはコンテナに必要な アップロードできます。
PutObject–標準アップロードの可用性	100 TPS	1 秒あたりに可能なオペレーションリクエスト クエストは調整されます。 クォータの引き上げをリクエストできます。リ ストされた TPS と平均オブジェクトサイズを指
PutObject–ストリーミングアップロードの可用性	10 TPS	1 秒あたりに可能なオペレーションリクエスト クエストは調整されます。 クォータの引き上げをリクエストできます。リ ストされた TPS と平均オブジェクトサイズを指
オブジェクトのライフサイクルポリシーのルール	10	オブジェクトのライフサイクルポリシーに含め ルの最大数。
メトリクスポリシーのルール	5	メトリクスポリシーに含めることができるルー クォータの引き上げをリクエストできます。

AWS Elemental MediaStore 関連情報

AWS Elemental MediaStore を利用する際に役立つ関連リソースを次の表にまとめました。

- クラスとワークショップ – AWS に関するスキルを磨き、実践的経験を積むために役立つ、職務別の特別コースとセルフペースラボへのリンクです。
- AWS 開発者用ツール – AWS アプリケーションの開発と管理のための開発者ツール、SDK、IDE ツールキット、およびコマンドラインツールへのリンクです。
- AWS ホワイトペーパー – アーキテクチャー、セキュリティ、エコノミクスなどのトピックをカバーし、AWS のソリューションアーキテクトや他の技術エキスパートによって書かれた、技術的な AWS ホワイトペーパーの包括的なリストへのリンクです。
- AWS サポートセンター – AWS サポートケースを作成および管理するためのハブです。フォーラム、技術上のよくある質問、サービス状態ステータス、AWS Trusted Advisor などの便利なリソースへのリンクも含まれています。
- AWS サポート – 1 対 1 での迅速な対応を行うサポートチャネルである AWS サポートに関する情報のメインウェブページです。AWS サポートは、クラウドでのアプリケーションの構築および実行を支援します。
- お問い合わせ – AWS 請求、アカウント、イベント、不正使用、およびその他の問題に関する問い合わせ先です。
- AWS サイトの利用規約 – 当社の著作権、商標、お客様のアカウント、ライセンス、サイトへのアクセス、およびその他のトピックに関する詳細情報です。

ユーザーガイドのドキュメント履歴

次の表は、今回のAWS Elemental MediaStore のリリースの内容をまとめたものです。このドキュメントの更新に関するお知らせについては、RSS フィードをご購読ください。

update-history-change	update-history-description	update-history-date
タグベースのアクセスコントロール (p. 62)	リソースに割り当てたリソースベースのタグに対して、アクセス許可を設定できるようになりました。	November 6, 2020
ExpiresAt field (p. 79)	アクセスログに、コンテナのライフサイクルポリシーの一時データルールに基づいて、オブジェクトの有効期限を示す `ExpiresAt` フィールドが含まれるようになりました。	July 16, 2020
ライフサイクル移行ルール (p. 28)	オブジェクトライフサイクルポリシーにライフサイクル移行ルールを追加して、オブジェクトが一定の期間に達した後に低頻度アクセス (IA) ストレージクラスに移動されるように設定できるようになりました。	April 20, 2020
コンテナを空にする (p. 52)	コンテナ内のすべてのオブジェクトを一度に削除できるようになりました。	April 7, 2020
Amazon CloudWatch メトリクスのサポート (p. 38)	メトリクスポリシーを設定して、MediaStore が CloudWatch に送信するメトリクスを指定できます。	March 30, 2020
オブジェクトの削除ルールのワイルドカード (p. 28)	オブジェクトのライフサイクルポリシーで、オブジェクトの削除ルールにワイルドカードを使用できるようになりました。これにより、ファイル名や拡張子に基づいて特定の日数後にサービスによって削除されるファイルを指定できます。	December 20, 2019
オブジェクトのライフサイクルポリシー (p. 28)	有効期限を秒単位で示すルールをオブジェクトのライフサイクルポリシーに追加できるようになりました。	September 13, 2019
AWS CloudFormation サポート (p. 10)	AWS CloudFormation テンプレートを使用して、コンテナを自動的に作成できるようになりました。AWS CloudFormation テンプレートは 5 つの API アクションのデータを管理し、コンテナの	May 17, 2019

	作成、アクセスのログ記録の設定を追加、デフォルトのコンテナポリシーの更新、 Cross-Origin Resource Sharing(CORS) ポリシーの追加、およびライフサイクルポリシーオブジェクトを追加します。	
ストリーミングアップロードの可用性に対するクォータ (p. 92)	ストリーミングアップロードの可用性に従うオブジェクト (チャンク転送されるオブジェクト) の場合、`PutObject` オペレーションは 10 TPS を、`GetObject` オペレーションは 25 TPS を超えることはできません。	April 8, 2019
オブジェクトのチャンク転送 (p. 47)	オブジェクトのチャンク転送のサポートが追加されました。この機能を使用すると、オブジェクトが完全にアップロードされる前にダウンロードできるように設定することができます。	April 5, 2019
アクセスのログ記録 (p. 75)	AWS Elemental MediaStore では、アクセスのログ記録をサポートするようになりました。これにより、コンテナ内でオブジェクトに対して行われたリクエストの詳細が記録されます。	February 25, 2019
オブジェクトのライフサイクルポリシー (p. 27)	現在のコンテナ内のオブジェクトの有効期限を管理する、オブジェクトのライフサイクルポリシーのサポートが追加されました。	December 12, 2018
オブジェクトサイズのクォータ (引き上げ後) (p. 92)	オブジェクトサイズのクォータは 25 MB になりました。	October 10, 2018
オブジェクトサイズのクォータ (引き上げ後) (p. 92)	オブジェクトサイズのクォータは 20 MB になりました。	September 6, 2018
AWS CloudTrail 統合 (p. 72)	CloudTrail 統合コンテンツは、CloudTrail サービスの最近の変更に合わせて更新されました。	July 12, 2018
CDN コラボレーション (p. 89)	Amazon CloudFront などのコンテンツ配信ネットワーク (CDN) で AWS Elemental MediaStore を使用する方法に関する情報を追加しました。	April 14, 2018
CORS 設定 (p. 22)	AWS Elemental MediaStore でクロスオリジンリソース共有 (CORS) がサポートされました。CORS では、特定のドメインにロードされたクライアントウェブアプリケーションが別のドメインのリソースと通信できます。	February 7, 2018

新しいサービスとガイド (p. 1)	これは、動画の配信およびストレージサービスである AWS Elemental MediaStore と AWS Elemental MediaStore ユーザーガイドの最初のリリースです。	November 27, 2017

Note

- AWS Media Services は、サービスの中断や障害が死亡、人身傷害、物的損害、または環境破壊につながるような、フェイルセーフ機能を要する用途または状況 (人命救助活動、航空機の航行や通信システム、航空管制、生命維持装置など) での使用を意図または目的としたものではありません。

AWS の用語集

最新の AWS の用語については、『AWS General Reference』の「AWS の用語集」を参照してください。

「翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。」