



ユーザーガイド

AWS Elemental MediaStore



AWS Elemental MediaStore: ユーザーガイド

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon の商標とトレードドレスは、Amazon 以外の製品またはサービスとの関連において、顧客に混乱を招いたり、Amazon の名誉または信用を毀損するような方法で使用することはできません。Amazon が所有しない他の商標はすべてそれぞれの所有者に帰属します。所有者は必ずしも Amazon との提携や関連があるわけではありません。また、Amazon の支援を受けているとはかぎりません。

Table of Contents

とは何ですか MediaStore?	1
概念と用語	1
関連サービス	3
アクセス MediaStore	3
料金	4
リージョンとエンドポイント	4
AWS Elemental のセットアップ MediaStore	5
にサインアップする AWS アカウント	5
管理アクセスを持つユーザーを作成する	6
はじめに	8
ステップ 1: AWS Elemental にアクセスする MediaStore	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	10
コンテナ	11
コンテナ名に関するルール	11
コンテナの作成	11
コンテナの詳細の表示	13
コンテナのリストの表示	14
コンテナの削除	15
ポリシー	16
コンテナポリシー	16
コンテナポリシーを表示する	17
コンテナポリシーを編集する	18
コンテナポリシーの例	19
CORS ポリシー	26
ユースケースのシナリオ	26
CORS ポリシーの追加	27
CORS ポリシーの表示	28
CORS ポリシーの編集	29
CORS ポリシーの削除	30
トラブルシューティング	31
CORS ポリシーの例	32
オブジェクトのライフサイクルポリシー	33

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	34
オブジェクトのライフサイクルポリシーを追加する	40
オブジェクトのライフサイクルポリシーを表示する	42
オブジェクトのライフサイクルポリシーを編集する	43
オブジェクトのライフサイクルポリシーを削除する	44
オブジェクトのライフサイクルポリシーの例	45
メトリクスポリシー	50
メトリクスポリシーの追加	50
メトリクスポリシーの表示	51
メトリクスポリシーの編集	51
メトリクスポリシーの例	52
フォルダ	56
フォルダ名に関するルール	56
フォルダの作成	57
フォルダの削除	57
オブジェクト	58
オブジェクトのアップロード	58
リストの表示	60
オブジェクトの詳細の表示	63
オブジェクトのダウンロード	64
オブジェクトの削除	65
1 つのオブジェクトを削除する	65
コンテナを空にする	66
セキュリティ	68
データ保護	69
データ暗号化	70
Identity and Access Management	70
対象者	70
アイデンティティを使用した認証	71
ポリシーを使用したアクセスの管理	75
AWS Elemental と IAM MediaStore の連携方法	77
アイデンティティベースポリシーの例	85
トラブルシューティング	88
ロギングとモニタリング	90
Amazon CloudWatch アラーム	90
AWS CloudTrail ログ	91

AWS Trusted Advisor	91
コンプライアンス検証	91
耐障害性	92
インフラストラクチャセキュリティ	93
サービス間の混乱した代理の防止	93
モニタリングとタグ付け	96
CloudTrail による API コールのログ記録	97
MediaStore内の情報 CloudTrail	97
例: ログファイルエントリ	99
によるモニタリング CloudWatch	100
CloudWatch ログ	101
CloudWatch イベント	111
CloudWatch指標	114
タグ付け	119
AWS Elemental でサポートされているリソース MediaStore	120
タグの命名規則と使用規則	120
タグの管理	120
CDN の使用	122
コンテナへのアクセスを CloudFront に許可する	122
オリジンアクセス制御 (OAC) の使用	123
共有シークレットの使用	123
MediaStore による HTTP キャッシュの操作	126
条件付きリクエスト	126
クォータ	128
関連情報	131
ドキュメント履歴	132
AWS 用語集	136
.....	CXXXVII

AWS Elemental MediaStore とは何ですか？

AWS Elemental MediaStore は、ライブオリジネーションに必要な高いパフォーマンスと即時の一貫性を提供するビデオオリジネーションおよびストレージサービスです。を使用すると MediaStore、動画アセットをコンテナ内のオブジェクトとして管理し、信頼性の高い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 強い一貫性、低レイテンシーの読み取りと書き込み、大量の同時リクエストを処理する機能が必要な場合に、断片化されたビデオファイルを保存するのに最適です。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勧めします。

トピック

- [AWS Elemental MediaStore ルの概念と用語](#)
- [関連サービス](#)
- [AWS Elemental へのアクセス MediaStore](#)
- [AWS Elemental ルの料金 MediaStore](#)
- [AWS Elemental のリージョンとエンドポイント MediaStore](#)

AWS Elemental MediaStore ルの概念と用語

ARN

[Amazon リソースネーム](#)。

[Body] (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。

コンテナ

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

MediaStore サービスへのエン트리ポイント。HTTPS ルート URL として提供されます。

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。

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 メディアコンテンツ配信の配信ポイントとなるため、オリジネーションサービスとみなされます。

[Path] (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](#)。

リソース

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 リソースネーム (ARN) が割り当てられています。では MediaStore、これがリソースとその 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、データや動画を視聴者に安全に配信するグローバルなコンテンツ配信ネットワーク (CDN) サービスです。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。AWS 必要なすべてのリソース (MediaStore コンテナなど) を記述したテンプレートを作成し、AWS CloudFormation それらのリソースのプロビジョニングと設定を自動的に行います。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、AWS マネジメントコンソールによる呼び出しを含む、アカウントの CloudTrail API への呼び出し AWS CLI、およびその他のサービスを監視できるサービスです。詳細については、[AWS CloudTrail ユーザーガイド](#)を参照してください。
- Amazon CloudWatch は、AWS AWS クラウドリソースとユーザーが実行するアプリケーションを監視するサービスです。CloudWatch イベントを使用すると、コンテナやオブジェクトのステータスの変化を追跡できます MediaStore。詳細については、[Amazon CloudWatch のドキュメントを参照してください](#)。
- AWS Identity and Access Management (IAM) は、AWS リソースへのユーザーアクセスを安全に管理するウェブサービスです。IAM を使用して、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、「[AWS Elemental のセットアップ MediaStore](#)」を参照してください。
- Amazon Simple Storage Service (Amazon S3) はオブジェクトストレージであり、任意の場所の任意の量のデータを保存および取得するように構築されています。詳細については、[Amazon S3 のドキュメント](#)参照してください。

AWS Elemental へのアクセス MediaStore

MediaStore 以下のいずれかの方法でアクセスできます。

- 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するタスクを実行する方法について説明します MediaStore。MediaStore コンソールを使用してアクセスするには:


```
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。CLI MediaStore エンドポイントを使用してアクセスするには:

```
aws mediastore
```

- MediaStore API — SDK に対応していないプログラミング言語を使用している場合、API アクションと [AWS Elemental MediaStore API リクエストの作成方法については、API リファレンスをご覧ください](#)。REST API MediaStore エンドポイントを使用してアクセスするには:

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

- AWS SDKs – AWS が SDK を提供してい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 にアクセスできます。SDK では、認証を簡素化し、開発環境と容易に統合して、MediaStore のコマンドに簡単にアクセスできます。詳細については、[Tools for Amazon Web Services](#) 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 ユーザーガイドを参照してください](#)。

AWS Elemental への料金 MediaStore

AWS 他 の製品と同様に、使用に関する契約や最低契約はありません。MediaStore 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 MediaStore 基本料金](#)」を参照してください。

AWS Elemental のリージョンとエンドポイント MediaStore

アプリケーションのデータレイテンシーを減らすため、MediaStore リクエストを行うためのリージョンのエンドポイントを用意しています。

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

利用可能な AWS リージョンの全リストを確認するには、『AWS ジェネラルリファレンス』の [「AWS Elemental MediaStore エンドポイントとクォータ」](#)を参照してください。MediaStore

AWS Elemental のセットアップ MediaStore

このセクションでは、AWS Elemental にアクセスするようにユーザーを設定するために必要な手順について説明します MediaStore。の ID とアクセスの管理に関する背景と追加情報については MediaStore、「」を参照してください [AWS Elemental の Identity and Access Management MediaStore](#)。

AWS Elemental の使用を開始するには MediaStore、次の手順を実行します。

トピック

- [にサインアップする AWS アカウント](#)
- [管理アクセスを持つユーザーを作成する](#)

にサインアップする AWS アカウント

がない場合は AWS アカウント、次の手順を実行して作成します。

にサインアップするには AWS アカウント

1. <https://portal.aws.amazon.com/billing/signup> を開きます。
2. オンラインの手順に従います。

サインアップ手順の一環として、通話呼び出しを受け取り、電話キーパッドで検証コードを入力するように求められます。

にサインアップすると AWS アカウント、AWS アカウントのルートユーザーが作成されます。ルートユーザーには、アカウントのすべての AWS のサービス とリソースへのアクセス権があります。セキュリティのベストプラクティスとして、ユーザーに管理アクセスを割り当て、ルートユーザーのみを使用して [ルートユーザーアクセスが必要なタスク](#) を実行してください。

AWS サインアッププロセスが完了すると、 から確認メールが送信されます。 <https://aws.amazon.com/> の [マイアカウント] を選んで、いつでもアカウントの現在のアクティビティを表示し、アカウントを管理できます。

管理アクセスを持つユーザーを作成する

にサインアップしたら AWS アカウント、 を保護し AWS アカウントのルートユーザー、 を有効にして AWS IAM Identity Center、 日常的なタスクにルートユーザーを使用しないように管理ユーザーを作成します。

のセキュリティ保護 AWS アカウントのルートユーザー

1. ルートユーザーを選択し、 AWS アカウント E メールアドレスを入力して、 アカウント所有者 [AWS Management Console](#) として 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。

ルートユーザーを使用してサインインする方法については、 AWS サインイン ユーザーガイドの「[ルートユーザーとしてサインインする](#)」を参照してください。

2. ルートユーザーの多要素認証 (MFA) を有効にします。

手順については、「IAM [ユーザーガイド](#)」の AWS アカウント「[ルートユーザーの仮想 MFA デバイスを有効にする \(コンソール\)](#)」を参照してください。

管理アクセスを持つユーザーを作成する

1. IAM アイデンティティセンターを有効にします。

手順については、「AWS IAM Identity Center ユーザーガイド」の「[AWS IAM Identity Centerの有効化](#)」を参照してください。

2. IAM アイデンティティセンターで、ユーザーに管理アクセスを付与します。

を ID ソース IAM アイデンティティセンターディレクトリとして使用する方法のチュートリアルについては、「[ユーザーガイド](#)」の「[デフォルトでユーザーアクセス IAM アイデンティティセンターディレクトリを設定するAWS IAM Identity Center](#)」を参照してください。

管理アクセス権を持つユーザーとしてサインインする

- IAM アイデンティティセンターのユーザーとしてサインインするには、IAM アイデンティティセンターのユーザーの作成時に E メールアドレスに送信されたサインイン URL を使用します。

IAM Identity Center ユーザーを使用してサインインする方法については、「AWS サインインユーザーガイド」の AWS「[アクセスポータルにサインインする](#)」を参照してください。

追加のユーザーにアクセス権を割り当てる

1. IAM アイデンティティセンターで、最小特権のアクセス許可を適用するというベストプラクティスに従ったアクセス許可セットを作成します。

手順については、「AWS IAM Identity Center ユーザーガイド」の「[権限設定を作成する](#)」を参照してください。

2. グループにユーザーを割り当て、そのグループにシングルサインオンアクセス権を割り当てます。

手順については、「AWS IAM Identity Center ユーザーガイド」の「[グループの参加](#)」を参照してください。

AWS Elemental Medmental MediaStore

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について説明します。

トピック

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

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

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

AWS Elemental にアクセスするには MediaStore

- AWS Management Consoleにログインし、<https://console.aws.amazon.com/mediastore/MediaStore> でコンソールを開きます。

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、「[AWS Elemental のセットアップ MediaStore](#)」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ッ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](#)」を参照してください。
3. [コンテナの作成] を選択します。AWS Elemental MediaStore によって、MediaStore 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。例えば、premium/canada。パスのフォルダがまだ存在しない場合は、AWS Elemental MediaStore によって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 のコンテナMediaStore

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ップロードしたときにのみ料金が発生します。の料金の詳細については、「」を参照してください。[AWS ElementalMediaStore料金](#)。

トピックス

- [コンテナ名に関するルール](#)
- [コンテナの作成](#)
- [コンテナの詳細の表示](#)
- [コンテナのリストの表示](#)
- [コンテナの削除](#)

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。

i 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. を開くMediaStoreコンソール<https://console.aws.amazon.com/mediastore/>。
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](#)」を参照してください。
4. 選択コンテナの作成。AWS ElementalMediaStoreは、コンテナのリストに新しいコンテナを追加します。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. を開く MediaStore コンソール <https://console.aws.amazon.com/mediastore/>。
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スペ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](#)」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ます。

```
{
  "Container": {
    "CreationTime": 1563558086.0,
    "AccessLoggingEnabled": false,
    "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleContainer",
    "Status": "ACTIVE",
    "Name": "ExampleContainer",
    "Endpoint": "https://aaabbbcccddee.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)

- を開く MediaStore コンソール <https://console.aws.amazon.com/mediastore/>。

[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ます。

```
{
  "Containers": [
    {
      "CreationTime": 1505317931.0,
      "Endpoint": "https://aaabbbcccddee.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://ffffggghhhiiijj.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. を開く MediaStore コンソール <https://console.aws.amazon.com/mediastore/>。
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. [Delete] (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 エレメンタルのポリシー-MediaStore

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

- [コンテナポリシー](#)-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ユーザーがすべての実行を許可するデフォルトポリシーを設定します。MediaStoreコンテナに対する操作。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](#)-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、異なるドメイン内のリソースと通信できるようにします。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](#)-許可されていますMediaStoreメトリクスはAmazonに送信するにはCloudWatch。MediaStoreは、デフォルトのメトリクスポリシーを設定しません。
- [オブジェクトのライフサイクルポリシー](#)-オブジェクトを保持する期間を制御しますMediaStoreコンテナ。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 のコンテナポリシー-MediaStore

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、すべてのAWS Elemental へのアクセスが許可されます。MediaStoreコンテナに対する操作。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](#)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。

トピックス

- [コンテナポリシーを表示する](#)
- [コンテナポリシーを編集する](#)
- [コンテナポリシーの例](#)

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. を開く MediaStore コンソールで: <https://console.aws.amazon.com/mediastore/>。
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. を開く MediaStore コンソールで: <https://console.aws.amazon.com/mediastore/>。
2. [Containers (コンテナ)] ページで、コンテナの名前を選択します。
3. [Edit policy (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”](#)」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。

トピックス

- [コンテナポリシーの例: デフォルト](#)
- [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](#)
- [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](#)
- [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 \(HTTP 対応\)](#)
- [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](#)
- [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](#)
- [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](#)
- [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](#)

コンテナポリシーの例: デフォルト

コンテナを作成するとき、AWS ElementalMediaStore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、次の操作を実行できます。[ポリシーの編集](#) デフォルトポリシーのアクセス権限が、コンテナに使用するアクセス権限と一致しない場合は、コンテナで指定します。

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 `PublicReadOverHttps` です。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

このポリシーの例では、へのアクセスが許可されます。GetObjectそしてDescribeObject指定したの root 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) に対するオペレーション<other acct number>。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 CrossAccountRolePostOverHttps です。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。また、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 にアクセスできませんが、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 CrossAccountFullAccessExceptPost です。任意のオブジェクトに対する、PutObject 以外のすべての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ります)。

```
{
  "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: 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、すべての AWS Elemental へのアクセスが許可されます。MediaStore指定されたコンテナ内のオブジェクトに対するオペレーション。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) の IP アドレスが許可されています。ただし、を除きます。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 値は標準の CIDR 表記を使用します。詳細については、「」を参照してください。[IP アドレス条件演算子IAM ユーザーガイド](#)

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

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

Cross-Origin Resource Sharing (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 での CORS のサポート付きMediaStoreでは、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構築できますMediaStoreそして、クロスオリジンアクセスを選択的に許可してMediaStoreリソースの使用料金を見積もることができます。

Note

Amazonを使っている場合CloudFrontCORS ポリシーが設定されているコンテナからコンテンツを配布するには、[AWS Elemental のディストリビューションを設定するMediaStore](#) (CORS を設定するためのキャッシュ動作を編集する手順を含む)。

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 を使用して CORS を有効にする方法について説明します。MediaStoreコンソール、またはプログラムでMediaStoreREST API と AWS SDK。

トピックス

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

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: AWS Elemental でライブストリーミング動画を配信するとしますMediaStore コンテナ名LiveVideo。ユーザーは、`http://livevideo.mediastore.ap-southeast-2.amazonaws.com` などの特定のオリジンから、動画マニフェストエンドポイント

www.example.com をロードします。を使いたいのはJavaScript認証されていないコンテナから発信される動画にアクセスするには、ビデオプレーヤーGETそしてPUTをリクエストします。通常、ブラウザはブロックしますJavaScriptこれらのリクエストを許可するとします。ただし、コンテナに対して CORS ポリシーを設定することで、これらのリクエストを明示的に有効化できます。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。ただし、任意のオリジンからのリクエストを許可したい。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 に追加する方法について説明します。MediaStoreコンテナ。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](#)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)

- を開くMediaStoreコンソール<https://console.aws.amazon.com/mediastore/>。
- [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。

- [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。
- 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)

- 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

Cross-Origin Resource Sharing (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. を開く MediaStore コンソール <https://console.aws.amazon.com/mediastore/>。
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

Cross-Origin Resource Sharing (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. を開く MediaStore コンソール <https://console.aws.amazon.com/mediastore/>。
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

Cross-Origin Resource Sharing (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. を開く MediaStore コンソール <https://console.aws.amazon.com/mediastore/>。
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”](#) 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。

ヘッダーがない場合、AWS ElementalMediaStoreは、リクエストをクロスオリジンリクエストとして扱わず、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 CORSRule と一致する必要があります。たとえば、オリジン CORSRule を許可するように `http://www.example.com` を設定すると、リクエストの

https://www.example.com オリジンと http://www.example.com:80 オリジンのいずれも、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。

トピックス

- [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](#)
- [CORS ポリシーの例: 特定のドメインへの読み取りアクセス](#)

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 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。MediaStore コンテナ。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

次のポリシーでは、`https://www.example.com` からコンテンツを取得するには MediaStore コンテナ。リクエストには、`https://www.example.com` からのすべての HTTP ヘッダーが含まれます。このサービスでは、`https://www.example.com` 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": [  
      "https://www.example.com"  
    ],  
    "MaxAgeSeconds": 3000  
  }  
]
```

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

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。指定したオブジェクトが有効期限に達すると、AWS Elemental MediaStore によって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](#)」を参照してください。

トピックス

- [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](#)
- [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を表示する](#)
- [オブジェクトのライフサイクルポリシーを編集する](#)
- [オブジェクトのライフサイクルポリシーを削除する](#)
- [オブジェクトのライフサイクルポリシーの例](#)

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 にオブジェクトを保持する期間を管理しますMediaStoreコンテナ。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](#)
- [DELETE Object](#)
- [ライフサイクル移行](#)

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: wildcard` に設定されます。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.m3u8`、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: numeric` に設定されます。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: EXPIRE` に設定されます。

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。また、一時データルールが原因で期限切れになるオブジェクトは、CloudWatch 有効期限が切れるイベント。

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 **wildcard** に設定します。同じルールで **prefix** と **wildcard** を混在さ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: numeric に設定されます。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: EXPIRE に設定されます。

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 `wildcard` に設定します。同じルールで `prefix` と `wildcard` を混在さ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: "numeric": [">=" , 30]` に設定されます。
- `action: ARCHIVE` に設定されます。

例

「LiveEvents」という名前のテナに、4 つのサブフォルダ Football、Baseball、Basketball、AwardsShow があるとします。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。

- 最初のルールは AWS Elemental に指示します。MediaStoreに保存されているオブジェクトを削除するにはLiveEvents/FootballフォルダとLiveEvents/Baseball128 日以上経過したフォルダです。
- 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 番目のルールは、サービスのオブジェクトを削除するように指示します。Footballパターンに一致するフォルダindex*.m3u815秒以上経過した後。MediaStore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](#)」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 ElementalMediaStore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](#)」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. を開く MediaStore コンソール <https://console.aws.amazon.com/mediastore/>。
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. を開く MediaStore コンソール <https://console.aws.amazon.com/mediastore/>。
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、代替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. を開く MediaStore コンソール <https://console.aws.amazon.com/mediastore/>。
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. を開く MediaStore コンソール <https://console.aws.amazon.com/mediastore/>。
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。

トピックス

- [オブジェクト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](#)
- [オブジェクト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](#)
- [オブジェクトライフサイクルポリシーの例: 低頻度アクセスのストレージクラスへの移行](#)
- [オブジェクトライフサイクルポリシーの例: 複数のルール](#)
- [オブジェクトライフサイクルポリシーの例: コンテナを空にする](#)

オブジェクト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"
  }
]
```

オブジェクトライフサイクルポリシーの例: 低頻度アクセスの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。コンテナにオブジェクトが保持されている場合、このポリシーが適用される前に、MediaStore では、ポリシーが有効になってから 1 日後にオブジェクトを削除します。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 のメトリクスポリシー MediaStore

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

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 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、rule1 オブジェクトがとという名前の 2 つのルールに一致する場合 rule2、CloudWatch これらのルールの 2 つのデータポイントが表示されます。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](#)
- [メトリクスポリシーの表示](#)
- [メトリクスポリシーの編集](#)
- [メトリクスポリシーの例](#)

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 が Amazon MediaStore 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 CloudWatch。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](#)」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. [https://console.aws.amazon.com/mediastore/ MediaStore](https://console.aws.amazon.com/mediastore/) でコンソールを開きます。
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. [https://console.aws.amazon.com/mediastore/ MediaStore](https://console.aws.amazon.com/mediastore/) でコンソールを開きます。
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 が Amazon MediaStore 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 CloudWatch。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、「[メトリクスポリシーの例](#)」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. [https://console.aws.amazon.com/mediastore/ MediaStore](https://console.aws.amazon.com/mediastore/) でコンソールを開きます。
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](#)
- [メトリクスポリシーの例: パスレベルのメトリクス](#)
- [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](#)
- [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](#)
- [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](#)

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 CloudWatch がメトリクスをコンテナレベルで Amazon MediaStore に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 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 CloudWatch がメトリクスをコンテナレベルで Amazon に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 CloudWatch ObjectGroupName=baseballGroup フォルダーへのリクエストのディメンションはです。
- 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 CloudWatch がメトリクスをコンテナレベルで Amazon MediaStore に送信する必要があることを示します。また、baseball/saturday との 2 MediaStore 内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 football/saturday。MediaStore リクエストのメトリクスは次のとおりです。

- baseball/saturday CloudWatch ObjectGroupName=baseballGroup フォルダーへのリクエストのディメンションはです。
- football/saturday CloudWatch フォルダーへのリクエストにはディメンションがありません ObjectGroupName=footballGroup。

```
{
  "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 CloudWatch がメトリクスをコンテナレベルで Amazon MediaStore に送信する必要があることを示します。また、MediaStore 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 CloudWatch がメトリクスをコンテナレベルで Amazon MediaStore に送信する必要があることを示します。さらに、sports/football/saturdayとの2 MediaStore つのフォルダのメトリックを送信する必要がありますsports/football。

MediaStore sports/football/saturday CloudWatch フォルダーへのリクエストのメトリクスのディメンションはですObjectGroupName=footballGroup1。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 のフォルダMediaStore

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 ElementalMediaStoreは、オブジェクトをフォルダに保存します。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、というコンテナがあるとしします。moviesをクリックし、という名前のファイルをアップロードしますmlaw.tsパスでpremium/canada。AWS ElementalMediaStoreは、オブジェクトを premium フォルダの下の canada サブフォルダに保存します。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。

AWS ElementalMediaStoreフォルダから最後のオブジェクトを削除すると、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 というフォルダ内にはファイルが存在せず、1 つのサブフォルダ canada のみが存在するとしします。canada サブフォルダには、mlaw.ts というファイルが 1 つ含まれています。ここで mlaw.ts ファイルを削除すると、premium フォルダと canada フォルダの両方が自動的に削除されます。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。

トピックス

- [フォルダ名に関するルール](#)
- [フォルダの作成](#)
- [フォルダの削除](#)

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 (_)、およびコロン (:)、ピリオド (.)、ハイフン (-)、チルダ (~)、アンダースコア (_)、およびコロン (:)
- 名前は 1 文字以上である必要があります。空のフォルダ名 (など)folder1//folder3/) は使用できません。
- 名前では、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 というフォルダを 2 つの異なるコンテナ (movies/myfolder と sports/myfolder) に作成できます。
- 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 ElementalMediaStoreは、オブジェクトをフォルダに保存します。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”](#)」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。

AWS ElementalMediaStoreフォルダから最後のオブジェクトを削除すると、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 というフォルダ内にはファイルが存在せず、1 つのサブフォルダ canada のみが存在するとします。canada サブフォルダには、mlaw.ts というファイルが 1 つ含まれています。ここで mlaw.ts ファイルを削除すると、premium フォルダと canada フォルダの両方が自動的に削除されます。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](#)を参照してください。

AWS Elemental のオブジェクトMediaStore

AWS Elemental 要素MediaStoreのアセットは、と呼ばれます。オブジェクト。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](#)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。

トピックス

- [オブジェクトのアップロード](#)
- [オブジェクトのリストの表示](#)
- [オブジェクトの詳細の表示](#)
- [オブジェクトのダウンロード](#)
- [オブジェクトの削除](#)

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、オブジェクトをフォルダに保存します。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 のフォルダMediaStore](#)」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の可用性を streaming に設定します。[API を使用してオブジェクトをアップロードする](#)場合に、このヘッダーの値を設定できます。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 (:) を含めることができます。

オブジェクトをアップロードするには (コンソール)

1. を開く MediaStore コンソール <https://console.aws.amazon.com/mediastore/>。
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 例えば、premium/canada です。指定したパスにフォルダがまだ存在していない場合は、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、`content-type`、`cache-control` (呼び出し側によるオブジェクトのキャッシュ動作のコントロールを許可)、`path` (オブジェクトをコンテナ内のフォルダに配置) のパラメータを含めることもできます。

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 Elemental を使用できます。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) を表示します。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. を開く MediaStoreコンソール <https://console.aws.amazon.com/mediastore/>。

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. を開く MediaStore コンソール <https://console.aws.amazon.com/mediastore/>。
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://  
aaabbbcccddee.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 ElementalMediaStoreは変更日、コンテンツの長さ、Etag (エンティティタグ)、コンテンツタイプなどの詳細を保存します。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](#)」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. を開く MediaStore コンソール <https://console.aws.amazon.com/mediastore/>。
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",
```


Note

フォルダ内の唯一のオブジェクトを削除すると、AWS Elemental ElementalMediaStore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というフォルダ内にはファイルが存在せず、1つのサブフォルダ canadaのみが存在するとします。canada サブフォルダには、mlaw.ts というファイルが1つ含まれています。ここで mlaw.ts ファイルを削除すると、premium フォルダと canada フォルダの両方が自動的に削除されます。

オブジェクトを削除するには (コンソール)

1. を開く MediaStoreコンソール <https://console.aws.amazon.com/mediastore/>。
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://aaabbbcccddee.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](#)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](#)。

コンテナを空にするには (コンソール)

1. を開く MediaStore コンソール <https://console.aws.amazon.com/mediastore/>。
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. テキストフィールドにコンテナ名を入力することでコンテナを空にすることを確定し、Empty。

AWS Elemental のセキュリティ MediaStore

のクラウドセキュリティが最優先事項 AWS です。お客様は AWS、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からメリットを得られます。

セキュリティは、AWS とユーザーの間で共有される責任です。[責任共有モデル](#)ではこれを、クラウドのセキュリティ、およびクラウド内でのセキュリティと説明しています:

- クラウドのセキュリティ — AWS は、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 AWS クラウド。また、は、お客様が安全に使用できるサービス AWS も提供します。コンプライアンス[AWS プログラム](#)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者は定期的にセキュリティの有効性をテストおよび検証。AWS Elemental に適用されるコンプライアンスプログラムの詳細については MediaStore、「[コンプライアンスプログラム AWS による対象範囲内のサービスコンプライアンスプログラム](#)」を参照してください。
- クラウドのセキュリティ — お客様の責任は、使用する AWS サービスによって決まります。また、お客様は、データの機密性、会社の要件、適用される法律や規制など、その他の要因についても責任を負います。

このドキュメントは、の使用時に責任共有モデルを適用する方法を理解するのに役立ちます MediaStore。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 MediaStore を達成するために を設定する方法を示します。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 でのデータ保護 MediaStore](#)
- [AWS Elemental の Identity and Access Management MediaStore](#)
- [でのログ記録とモニタリング AWS Elemental MediaStore](#)
- [AWS Elemental のコンプライアンス検証 MediaStore](#)
- [AWS Elemental の耐障害性 MediaStore](#)
- [AWS Elemental のインフラストラクチャセキュリティ MediaStore](#)
- [サービス間の混乱した代理の防止](#)

AWS Elemental でのデータ保護 MediaStore

責任 AWS [共有モデル](#)、AWS Elemental でのデータ保護に適用されます MediaStore。このモデルで説明されているように、AWS はすべての を実行するグローバルインフラストラクチャを保護する責任があります AWS クラウド。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。また、使用する AWS のサービスのセキュリティ設定と管理タスクもユーザーの責任となります。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に投稿された記事「[AWS 責任共有モデルおよび GDPR](#)」を参照してください。

データ保護の目的で、認証情報を保護し AWS アカウント、AWS IAM Identity Center または AWS Identity and Access Management (IAM) を使用して個々のユーザーを設定することをお勧めします。この方法により、それぞれのジョブを遂行するために必要な権限のみが各ユーザーに付与されます。また、次の方法でデータを保護することもお勧めします:

- 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 は必須であり TLS 1.3 がお勧めです。
- で API とユーザーアクティビティのログ記録を設定します AWS CloudTrail。
- AWS 暗号化ソリューションと、内のすべてのデフォルトのセキュリティコントロールを使用します AWS のサービス。
- 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。これらは、Amazon S3 に保存されている機密データの検出と保護を支援します。
- コマンドラインインターフェイスまたは API AWS を介して 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。利用可能な 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。

お客様の E メールアドレスなどの極秘または機密情報は、タグ、または名前フィールドなどの自由形式のテキスト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、MediaStore または AWS のサービス SDK を使用して AWS CLI または他の を操作する場合も同様です。AWS SDKs 名前に使用する自由記述のテキストフィールドやタグに入力したデータは、課金や診断ログに使用される場合があります。外部サーバーへの URL を提供する場合は、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 に含めないように強くお勧めします。

データ暗号化

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

- コンテナポリシーを作成して、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制御します。詳細については、「[the section called “コンテナポリシー”](#)」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 リソースへのクロスオリジンアクセスを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”](#)」を参照してください。

AWS Elemental の Identity and Access Management MediaStore

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制御 AWS のサービス するのに役立つです。IAM 管理者は、誰を認証 (サインイン) し、誰に MediaStore リソースの使用を承認する (アクセス許可を付与する) かを制御します。IAM は、追加料金なしで AWS のサービス 使用できる です。

トピック

- [対象者](#)
- [アイデンティティを使用した認証](#)
- [ポリシーを使用したアクセスの管理](#)
- [AWS Elemental と IAM MediaStore の連携方法](#)
- [AWS Elemental のアイデンティティベースのポリシーの例 MediaStore](#)
- [AWS Elemental の MediaStore アイデンティティとアクセスのトラブルシューティング](#)

対象者

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

サービスユーザー – MediaStore サービスを使用してジョブを実行する場合、管理者から必要な認証情報とアクセス許可が与えられます。さらに多くの MediaStore 機能を使用して作業を行う

場合は、追加のアクセス許可が必要になることがあります。アクセスの管理方法を理解しておく
と、管理者に適切な許可をリクエストするうえで役立ちます。の機能にアクセスできない場合は、
MediaStore「」を参照してください[AWS Elemental の MediaStore アイデンティティとアクセスの
トラブルシューティング](#)。

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、通常、へのフルアクセスが
あります MediaStore。サービスユーザーがどの MediaStore 機能やリソースにアクセスするかを
決めるのは管理者の仕事です。その後、IAM 管理者にリクエストを送信して、サービスユーザーの
権限を変更する必要があります。このページの情報を点検して、IAM の基本概念を理解してくださ
い。会社で IAM を で使用する方法の詳細については MediaStore、「」を参照してください[AWS
Elemental と IAM MediaStore の連携方法](#)。

IAM 管理者 – IAM 管理者は、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 MediaStore。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、「」を参照してください[AWS Elemental のアイデンティティベースのポリシー
の例 MediaStore](#)。

アイデンティティを使用した認証

認証とは、ID 認証情報 AWS を使用して にサインインする方法です。として、IAM ユーザーとして
AWS アカウントのルートユーザー、または IAM ロールを引き受けて認証 (にサインイン AWS) さ
れる必要があります。

ID ソースを通じて提供された認証情報を使用して、フェデレーテッド ID AWS として にサイン
インできます。AWS IAM Identity Center (IAM Identity Center) ユーザー、会社のシングルサインオン
認証、Google または Facebook の認証情報は、フェデレーテッド ID の例です。フェデレーティ
ドアイデンティティとしてサインインする場合、IAM ロールを使用して、前もって管理者により ID
フェデレーションが設定されています。フェデレーション AWS を使用して にアクセスすると、間
接的にロールを引き受けることとなります。

ユーザーのタイプに応じて、AWS Management Console または AWS アクセスポータルにサイン
インできます。へのサインインの詳細については AWS、「ユーザーガイド」の「[へのサインイン
AWS アカウント](#)方法AWS サインイン」を参照してください。

AWS プログラムで にアクセスする場合、は Software Development Kit (SDK) とコマンドラインイ
ンターフェイス (CLI) AWS を提供し、認証情報を使用してリクエストに暗号で署名します。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。推奨される方法を使用
してリクエストを自分で署名する方法の詳細については、IAM [ユーザーガイドの API AWS リクエ
スト](#)の署名を参照してください。

使用する認証方法を問わず、追加セキュリティ情報の提供をリクエストされる場合もあります。例えば、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向上させることをお勧めします。詳細については、『AWS IAM Identity Center ユーザーガイド』の「[Multi-factor authentication](#)」(多要素認証) および『IAM ユーザーガイド』の「[AWSにおける多要素認証 \(MFA\) の使用](#)」を参照してください。

AWS アカウント ルートユーザー

を作成するときは AWS アカウント、アカウント内のすべての およびリソースへの AWS のサービス 完全なアクセス権を持つ 1 つのサインインアイデンティティから始めます。この ID 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サインインすることでアクセスできます。日常的なタスクには、ルートユーザーを使用しないことを強くお勧めします。ルートユーザーの認証情報は保護し、ルートユーザーでしか実行できないタスクを実行するときに使用します。ルートユーザーとしてサインインする必要があるタスクの完全なリストについては、『IAM ユーザーガイド』の「[ルートユーザー認証情報が必要なタスク](#)」を参照してください。

フェデレーテッドアイデンティティ

ベストプラクティスとして、管理者アクセスを必要とするユーザーを含む人間のユーザーに、一時的な認証情報を使用して にアクセスするための ID プロバイダーとのフェデレーションの使用を要求 AWS のサービスします。

フェデレーテッド ID は、エンタープライズユーザーディレクトリ、ウェブ ID プロバイダー、AWS Directory Service、アイデンティティセンターディレクトリ、または ID ソースを通じて提供された認証情報 AWS のサービス を使用して にアクセスするユーザーです。フェデレーテッド ID が にアクセスすると AWS アカウント、ロールを引き受け、ロールは一時的な認証情報を提供します。

アクセスを一元管理する場合は、AWS IAM Identity Centerを使用することをお勧めします。IAM Identity Center でユーザーとグループを作成することも、独自の ID ソース内のユーザーとグループのセットに接続して同期して、すべての AWS アカウント とアプリケーションで使用することもできます。IAM Identity Center の詳細については、『AWS IAM Identity Center ユーザーガイド』の「[What is IAM Identity Center?](#)」(IAM Identity Center とは) を参照してください。

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対して特定のアクセス許可 AWS アカウント を持つ 内のアイデンティティです。可能であれば、パスワードやアクセスキーなどの長期的な認証情報を保有する IAM ユーザーを作成する代わりに、一時認証情報を使用することをお勧めしま

す。ただし、IAM ユーザーでの長期的な認証情報が必要な特定のユースケースがある場合は、アクセスキーをローテーションすることをお勧めします。詳細については、「IAM ユーザーガイド」の「[長期的な認証情報を必要とするユースケースのためにアクセスキーを定期的にローテーションする](#)」を参照してください。

[IAM グループ](#)は、IAM ユーザーの集団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、複数のユーザーに対して一度に権限を指定できます。多数のユーザーグループがある場合、グループを使用することで権限の管理が容易になります。例えば、IAMAdmins 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権限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 人の人または 1 つの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提供されます。詳細については、『IAM ユーザーガイド』の「[IAM ユーザー \(ロールではなく\) の作成が適している場合](#)」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 AWS アカウント を持つ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、特定のユーザーには関連付けられていません。ロールを切り替える AWS Management Console ことで、[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、または AWS API AWS CLI 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:

- フェデレーションユーザーアクセス - フェデレーティッドアイデンティティに権限を割り当てるには、ロールを作成してそのロールの権限を定義します。フェデレーティッドアイデンティティが認証されると、そのアイデンティティはロールに関連付けられ、ロールで定義されている権限が付与されます。フェデレーションの詳細については、『IAM ユーザーガイド』の「[サードパーティーアイデンティティプロバイダー向けロールの作成](#)」を参照してください。IAM アイデンティティセンターを使用する場合、権限セットを設定します。アイデンティティが認証後にアクセスできるものを制御するため、IAM Identity Center は、権限セットを IAM のロールに関連付けます。権限セットの詳細については、『AWS IAM Identity Center ユーザーガイド』の「[権限セット](#)」を参照してください。
- 一時的な IAM ユーザー権限 - IAM ユーザーまたはロールは、特定のタスクに対して複数の異なる権限を一時的に IAM ロールで引き受けることができます。

- 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、別のアカウントの人物 (信頼済みプリンシパル) に許可できます。クロスアカウントアクセス権を付与する主な方法は、ロールを使用することです。ただし、一部の AWS のサービス、(ロールをプロキシとして使用する代わりに) リソースに直接ポリシーをアタッチできます。クロスアカウントアクセスにおけるロールとリソースベースのポリシーの違いについては、『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。
- クロスサービスアクセス — 一部の は、他の の機能 AWS のサービス を使用します AWS のサービス。例えば、あるサービスで呼び出しを行うと、通常そのサービスによって Amazon EC2 でアプリケーションが実行されたり、Amazon S3 にオブジェクトが保存されたりします。サービスでは、呼び出し元プリンシパルの権限、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使用してこれを行う場合があります。
- 転送アクセスセッション (FAS) – IAM ユーザーまたはロールを使用して でアクションを実行する場合 AWS、ユーザーはプリンシパルと見なされます。一部のサービスを使用する際に、アクションを実行することで、別のサービスの別のアクションがトリガーされることがあります。FAS は、 を呼び出すプリンシパルのアクセス許可を AWS のサービス、ダウストリームサービス AWS のサービス へのリクエストのリクエストと組み合わせて使用します。FAS リクエストは、サービスが他の AWS のサービス またはリソースとのやり取りを完了する必要があるリクエストを受け取った場合にのみ行われます。この場合、両方のアクションを実行するためのアクセス許可が必要です。FAS リクエストを行う際のポリシーの詳細については、「[転送アクセスセッション](#)」を参照してください。
- サービスロール - サービスがユーザー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。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 のサービスに権限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 – サービスにリンクされたロールは、 にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは に表示され AWS アカウント、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権限を表示できますが、編集することはできません。
- 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 リクエストを行う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的な認証情報を取得できます。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 ロールを使用して権限を付与する](#)」を参照してください。

IAM ロールと IAM ユーザーのどちらを使用するかについては、『IAM ユーザーガイド』の「[\(IAM ユーザーではなく\) IAM ロールをいつ作成したら良いのか?](#)」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

でアクセスを制御する AWS には、ポリシーを作成し、AWS ID またはリソースにアタッチします。ポリシーは、アイデンティティまたはリソースに関連付けられているときにアクセス許可を定義するオブジェクトです。は、プリンシパル(ユーザー、ルートユーザー、またはロールセッション) AWS がリクエストを行うときに、これらのポリシー AWS を評価します。ポリシーでの権限により、リクエストが許可されるか拒否されるかが決まります。ほとんどのポリシーは JSON ドキュメント AWS として保存されます。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 ポリシー概要](#)」を参照してください。

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどんなリソースにどんな条件でアクションを実行できるかということです。

デフォルトでは、ユーザーやロールに権限はありません。IAM 管理者は、リソースで必要なアクションを実行するための権限をユーザーに付与する IAM ポリシーを作成できます。その後、管理者はロールに IAM ポリシーを追加し、ユーザーはロールを引き継ぐことができます。

IAM ポリシーは、オペレーションの実行方法を問わず、アクションの権限を定義します。例えば、iam:GetRole アクションを許可するポリシーがあるとします。そのポリシーを持つユーザーは、AWS Management Console、AWS CLI または AWS API からロール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

アイデンティティベースポリシーは、IAM ユーザー、ユーザーのグループ、ロールなど、アイデンティティにアタッチできる JSON 権限ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。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、「IAM ユーザーガイド」の「[IAM ポリシーの作成](#)」を参照してください。

アイデンティティベースポリシーは、さらにインラインポリシーまたはマネージド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、内の複数のユーザー、グループ、ロールにアタッチできるスタンドアロンポリシーです AWS アカウント。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。マネージド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「[マネージド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

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。リソースベースのポリシーには例として、IAM ロールの信頼ポリシーや Amazon S3 バケットポリシーがあげられます。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスを制御できます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーでは、[プリンシパルを指定する](#)必要があります。プリンシパルには、アカウント、ユーザー、ロール、フェデレーテッドユーザー、またはを含めることができます AWS のサービス。

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は、IAM の AWS マネージドポリシーを使用できません。

アクセスコントロールリスト (ACL)

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするための権限を持つかをコントロールします。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

Amazon S3、AWS WAF、および Amazon VPC は、ACLs。ACL の詳細については、『Amazon Simple Storage Service デベロッパーガイド』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 は、一般的ではない追加のポリシータイプをサポートします。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権限を設定できます。

- **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によって IAM エンティティ (IAM ユーザーまたはロール) に付与できる権限の上限を設定する高度な機能です。エンティティに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権限は、エンティティのアイデンティティベースポリシーとそのアクセス許可の境界の共通部分になります。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では、アクセス許可の境界は制限されません。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。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 (SCPs) – SCPs は、 の組織または組織単位 (OU) に対する最大アクセス許可を指定する JSON ポリシーです AWS Organizations。AWS Organizations は、AWS アカウント ビジネスが所有する複数の をグループ化して一元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、サービスコントロールポリシー (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、各 を含むメンバーアカウントのエンティティのアクセス許可を制限します AWS アカウントのルートユーザー。Organizations と SCP の詳細については、『AWS Organizations ユーザーガイド』の「[SCP の仕組み](#)」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ションユーザーの一時的な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。結果としてセッションの権限は、ユーザーまたはロールのアイデンティティベースポリシーとセッションポリシーの共通部分になります。また、リソースベースのポリシーから権限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、権限は無効になります。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権限を理解するのがさらに難しくなります。複数のポリシータイプが関与する場合にリクエストを許可するかどうかが AWS を決定する方法については、IAM ユーザーガイドの「[ポリシー評価ロジック](#)」を参照してください。

AWS Elemental と IAM MediaStore の連携方法

IAM を使用して へのアクセスを管理する前に MediaStore、 で使用できる IAM 機能について学びます MediaStore。

AWS Elemental で使用できる IAM 機能 MediaStore

IAM 機能	MediaStore サポート
アイデンティティベースのポリシー	Yes

IAM 機能	MediaStore サポート
リソースベースのポリシー	はい
ポリシーアクション	Yes
ポリシーリソース	はい
ポリシー条件キー (サービス固有)	はい
ACL	No
ABAC (ポリシー内のタグ)	部分的
一時的な認証情報	Yes
プリンシパル権限	Yes
サービスロール	あり
サービスリンクロール	いいえ

MediaStore およびその他の AWS のサービスがほとんどの IAM 機能と連携する方法の概要を把握するには、「IAM ユーザーガイド」の「IAM [AWS と連携する のサービス](#)」を参照してください。

のアイデンティティベースのポリシー MediaStore

アイデンティティベースポリシーをサポートする	Yes
------------------------	-----

アイデンティティベースポリシーは、IAM ユーザー、ユーザーグループ、ロールなど、アイデンティティにアタッチできる JSON 権限ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。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、『IAM ユーザーガイド』の「[IAM ポリシーの作成](#)」を参照してください。

IAM アイデンティティベースのポリシーでは、許可または拒否するアクションとリソース、およびアクションを許可または拒否する条件を指定できます。プリンシパルは、それが添付されている

ユーザーまたはロールに適用されるため、アイデンティティベースのポリシーでは指定できません。JSON ポリシーで使用できるすべての要素については、「IAM ユーザーガイド」の「[IAM JSON ポリシーの要素のリファレンス](#)」を参照してください。

のアイデンティティベースのポリシーの例 MediaStore

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、「」を参照してください [AWS Elemental のアイデンティティベースのポリシーの例 MediaStore](#)。

内のリソースベースのポリシー MediaStore

リソースベースのポリシーのサポート	はい
-------------------	----

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。リソースベースのポリシーには例として、IAM ロールの信頼ポリシーや Amazon S3 バケットポリシーがあげられます。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスを制御できます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーでは、[プリンシパルを指定する](#)必要があります。プリンシパルには、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、またはを含めることができます AWS のサービス。

クロスアカウントアクセスを有効にするには、アカウント全体、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 がある場合 AWS アカウント、信頼されたアカウントの IAM 管理者は、プリンシパルエンティティ (ユーザーまたはロール) にリソースへのアクセス許可も付与する必要があります。IAM 管理者は、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権限を付与します。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追加する必要はありません。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。

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](#)」を参照してください。

のポリシーアクション MediaStore

ポリシーアクションに対するサポート はい

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどんなリソースにどんな条件でアクションを実行できるかということです。

JSON ポリシーの Action 要素には、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が記述されます。ポリシーアクションの名前は通常、関連付けられた AWS API オペレーションと同じです。一致する API オペレーションのない権限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、**依存アクション**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権限を付与するポリシーで使用されます。

MediaStore アクションのリストを確認するには、「サービス認証リファレンス」の「[AWS Elemental で定義されるアクション MediaStore](#)」を参照してください。

のポリシーアクションは、アクションの前に次のプレフィックス MediaStore を使用します。

```
mediastore
```

単一のステートメントで複数のアクションを指定するには、アクションをカンマで区切ります。

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

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、「」を参照してください [AWS Elemental のアイデンティティベースのポリシーの例 MediaStore](#)。

のポリシーリソース MediaStore

ポリシーリソースに対するサポート	はい
------------------	----

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースにどのような条件でアクションを実行できるかということです。

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 要素を含める必要があります。ベストプラクティスとして、[Amazon リソースネーム \(ARN\)](#) を使用してリソースを指定します。これは、リソースレベルの権限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、リソースレベルの権限をサポートしないアクションの場合は、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すために、ワイルドカード (*) を使用します。

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

MediaStore リソースタイプとその ARNs 「サービス認証リファレンス」の [「AWS Elemental で定義されるリソース MediaStore」](#) を参照してください。どのアクションで各リソースの ARN を指定できるかについては、[「AWS Elemental で定義されるアクション MediaStore」](#) を参照してください。

MediaStore コンテナリソースには次の ARN があります。

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

ARN の形式の詳細については、「Amazon [リソースネーム \(ARNs AWS 「サービス名前空間」](#)」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"
```

のポリシー条件キー MediaStore

サービス固有のポリシー条件キーのサポート	はい
----------------------	----

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどんなリソースにどんな条件でアクションを実行できるかということです。

Condition 要素 (または Condition ブロック) を使用すると、ステートメントが有効な条件を指定できます。Condition 要素はオプションです。イコールや未満などの [条件演算子](#) を使用して条件式を作成することで、ポリシーの条件とリクエスト内の値を一致させることができます。

1つのステートメントに複数の Condition 要素を指定するか、1つの Condition 要素に複数のキーを指定すると、AWS は AND 論理演算子を使用してそれらを評価します。1つの条件キーに複数の値を指定すると、は論理ORオペレーションを使用して条件 AWS を評価します。ステートメントの権限が付与される前にすべての条件が満たされる必要があります。

条件を指定する際にプレースホルダー変数も使用できます。例えば IAM ユーザーに、IAM ユーザー名がタグ付けされている場合のみリソースにアクセスできる権限を付与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「[IAM ポリシーの要素: 変数およびタグ](#)」を参照してください。

AWS は、グローバル条件キーとサービス固有の条件キーをサポートします。すべての AWS グローバル条件キーを確認するには、「IAM ユーザーガイド」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。

MediaStore 条件キーのリストを確認するには、「サービス認証リファレンス」の「[AWS Elemental の条件キー MediaStore](#)」を参照してください。条件キーを使用できるアクションとリソースについては、「[AWS Elemental で定義されるアクション MediaStore](#)」を参照してください。

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、「」を参照してください [AWS Elemental のアイデンティティベースのポリシーの例 MediaStore](#)。

ACLs MediaStore

ACL のサポート	No
-----------	----

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするための権限を持つかを制御します。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

での ABAC MediaStore

ABAC (ポリシー内のタグ) のサポート	部分的
-----------------------	-----

属性ベースのアクセスコントロール (ABAC) は、属性に基づいて権限を定義する認可戦略です。では AWS、これらの属性はタグと呼ばれます。タグは、IAM エンティティ (ユーザーまたはロール) および多くの AWS リソースにアタッチできます。エンティティとリソースのタグ付けは、ABAC の最初の手順です。その後、プリンシパルのタグがアクセスしようとしているリソースのタグと一致した場合に操作を許可するように ABAC ポリシーを設計します。

ABAC は、急成長する環境やポリシー管理が煩雑になる状況で役立ちます。

タグに基づいてアクセスを管理するには、`aws:ResourceTag/key-name`、`aws:RequestTag/key-name`、または `aws:TagKeys` の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。

サービスがすべてのリソースタイプに対して 3 つの条件キーすべてをサポートする場合、そのサービスの値ははいです。サービスが一部のリソースタイプに対してのみ 3 つの条件キーのすべてをサポートする場合、値は「部分的」になります。

ABAC の詳細については、『IAM ユーザーガイド』の「[ABAC とは?](#)」を参照してください。ABAC をセットアップするステップを説明するチュートリアルについては、「IAM ユーザーガイド」の「[属性に基づくアクセスコントロール \(ABAC\) を使用する](#)」を参照してください。

での一時的な認証情報の使用 MediaStore

一時的な認証情報のサポート	はい
---------------	----

一部の は、一時的な認証情報を使用してサインインすると機能 AWS のサービスしません。一時的な認証情報 AWS のサービス を使用する などの詳細については、IAM ユーザーガイドの [AWS のサービス「IAM と連携する](#)」を参照してください。

ユーザー名とパスワード以外の AWS Management Console 方法で にサインインする場合、一時的な認証情報を使用します。例えば、会社の Single Sign-On (SSO) リンク AWS を使用して にアクセ

スすると、そのプロセスによって一時的な認証情報が自動的に作成されます。また、ユーザーとしてコンソールにサインインしてからロールを切り替える場合も、一時的な認証情報が自動的に作成されます。ロールの切り替えに関する詳細については、「IAM ユーザーガイド」の「[ロールへの切り替え \(コンソール\)](#)」を参照してください。

一時的な認証情報は、AWS CLI または AWS API を使用して手動で作成できます。その後、これらの一時的な認証情報を使用してにアクセスします AWS。AWS 長期的なアクセスキーを使用する代わりに、一時的な認証情報を動的に生成することをお勧めします。詳細については、「[IAM の一時的セキュリティ認証情報](#)」を参照してください。

のクロスサービスプリンシパル許可 MediaStore

フォワードアクセスセッション (FAS) をサポート	はい
----------------------------	----

IAM ユーザーまたはロールを使用してでアクションを実行すると AWS、プリンシパルと見なされます。一部のサービスを使用する際に、アクションを実行することで、別のサービスの別のアクションがトリガーされることがあります。FAS は、を呼び出すプリンシパルのアクセス許可を AWS のサービス、ダウンストリームサービス AWS のサービス へのリクエストリクエストリクエストと組み合わせて使用します。FAS リクエストは、サービスが他の AWS のサービス またはリソースとのやり取りを完了する必要があるリクエストを受け取った場合にのみ行われます。この場合、両方のアクションを実行するためのアクセス許可が必要です。FAS リクエストを行う際のポリシーの詳細については、「[転送アクセスセッション](#)」を参照してください。

MediaStore のサービスロール

サービスロールに対するサポート	あり
-----------------	----

サービスロールとは、サービスがユーザー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。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 のサービスに権限を委任するロールの作成](#)」を参照してください。

⚠ Warning

サービスロールのアクセス許可を変更すると、MediaStore 機能が破損する可能性があります。が指示する場合以外 MediaStore は、サービスロールを編集しないでください。

のサービスにリンクされたロール MediaStore

サービスにリンクされたロールのサポート いいえ

サービスにリンクされたロールは、にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは に表示され AWS アカウント、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権限を表示できますが、編集することはできません。

サービスにリンクされたロールの作成または管理の詳細については、「[IAM と提携するAWS のサービス](#)」を参照してください。表の中から、[Service-linked role] (サービスにリンクされたロール) 列に Yes と記載されたサービスを見つけます。サービスにリンクされたロールに関するドキュメントをサービスで表示するには、[Yes] リンクを選択します。

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

デフォルトでは、ユーザーとロールにはリソースを作成または変更 MediaStore するアクセス許可はありません。また、AWS Command Line Interface (AWS CLI) AWS Management Console、または AWS API を使用してタスクを実行することはできません。IAM 管理者は、リソースに必要なアクションを実行するための権限をユーザーに付与する IAM ポリシーを作成できます。その後、管理者はロールに IAM ポリシーを追加し、ユーザーはロールを引き受けることができます。

これらサンプルの JSON ポリシードキュメントを使用して、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、『IAM ユーザーガイド』の「[IAM ポリシーの作成](#)」を参照してください。

各リソースタイプの ARN の形式など MediaStore、で定義されるアクションとリソースタイプの詳細については、「サービス認証リファレンス」の「[AWS Elemental のアクション、リソース、および条件キー MediaStore](#)」を参照してください。ARNs

トピック

- [ポリシーのベストプラクティス](#)

- [MediaStore コンソールを使用する](#)
- [自分の権限の表示をユーザーに許可する](#)

ポリシーのベストプラクティス

ID ベースのポリシーは、ユーザーのアカウントで誰かが MediaStore リソースを作成、アクセス、または削除できる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ポリシーを作成したり編集したりする際には、以下のガイドラインと推奨事項に従ってください:

- AWS 管理ポリシーを開始し、最小特権のアクセス許可に移行する – ユーザーとワークロードにアクセス許可を付与するには、多くの一般的なユースケースにアクセス許可を付与する AWS 管理ポリシーを使用します。これらはで使用できます AWS アカウント。ユースケースに固有の AWS カスタマー管理ポリシーを定義して、アクセス許可をさらに減らすことをお勧めします。詳細については、『IAM ユーザーガイド』の「[AWS マネージドポリシー](#)」または「[AWS ジョブ機能の管理ポリシー](#)」を参照してください。
- 最小特権を適用する – IAM ポリシーで権限を設定するときは、タスクの実行に必要な権限のみを付与します。これを行うには、特定の条件下で特定のリソースに対して実行できるアクションを定義します。これは、最小特権権限とも呼ばれています。IAM を使用して権限を適用する方法の詳細については、『IAM ユーザーガイド』の「[IAM でのポリシーと権限](#)」を参照してください。
- IAM ポリシーで条件を使用してアクセスをさらに制限する – ポリシーに条件を追加して、アクションやリソースへのアクセスを制限できます。例えば、ポリシー条件を記述して、すべてのリクエストを SSL を使用して送信するように指定できます。条件を使用して、などの特定の を介してサービスアクションが使用される場合に AWS のサービス、サービスアクションへのアクセスを許可することもできます AWS CloudFormation。詳細については、『IAM ユーザーガイド』の [IAM JSON policy elements: Condition](#) (IAM JSON ポリシー要素 : 条件) を参照してください。
- IAM Access Analyzer を使用して IAM ポリシーを検証し、安全で機能的な権限を確保する – IAM Access Analyzer は、新規および既存のポリシーを検証して、ポリシーが IAM ポリシー言語 (JSON) および IAM のベストプラクティスに準拠するようにします。IAM アクセスアナライザーは 100 を超えるポリシーチェックと実用的な推奨事項を提供し、安全で機能的なポリシーの作成をサポートします。詳細については、『IAM ユーザーガイド』の「[IAM Access Analyzer ポリシーの検証](#)」を参照してください。
- 多要素認証 (MFA) を要求する – IAM ユーザーまたはルートユーザーを必要とするシナリオがある場合は AWS アカウント、セキュリティを強化するために MFA を有効にします。API オペレーションが呼び出されるときに MFA を必須にするには、ポリシーに MFA 条件を追加します。詳細については、『IAM ユーザーガイド』の「[MFA 保護 API アクセスの設定](#)」を参照してください。

IAM でのベストプラクティスの詳細については、『IAM ユーザーガイド』の「[IAM でのセキュリティのベストプラクティス](#)」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、最小限のアクセス許可のセットが必要です。これらのアクセス許可により、の MediaStore リソースの詳細を一覧表示および表示できます AWS アカウント。最小限必要な許可よりも制限が厳しいアイデンティティベースのポリシーを作成すると、そのポリシーを持つエンティティ (ユーザーまたはロール) に対してコンソールが意図したとおりに機能しません。

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。

ユーザーとロールが MediaStore 引き続きコンソールを使用できるようにするには、エンティティに MediaStore *ConsoleAccess* または *ReadOnly* AWS 管理ポリシーもアタッチします。詳細については、「IAM ユーザーガイド」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。

自分の権限の表示をユーザーに許可する

この例では、ユーザーアイデンティティにアタッチされたインラインおよびマネージドポリシーの表示を IAM ユーザーに許可するポリシーの作成方法を示します。このポリシーには、コンソールで、または AWS CLI または AWS API を使用してプログラムでこのアクションを実行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:ListGroupsWithUser",
        "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": "*"
}
]
```

AWS Elemental の MediaStore アイデンティティとアクセスのトラブルシューティング

次の情報は、および IAM の使用時に発生する可能性がある一般的な問題の診断 MediaStore と修正に役立ちます。

トピック

- [でアクションを実行する権限がない MediaStore](#)
- [iam を実行する権限がありません。PassRole](#)
- [自分の 以外のユーザーに自分の MediaStore リソース AWS アカウント へのアクセスを許可したい](#)

でアクションを実行する権限がない MediaStore

「I am not authorized to perform an action in Amazon Bedrock」というエラーが表示された場合、そのアクションを実行できるようにポリシーを更新する必要があります。

次のエラー例は、mateojackson IAM ユーザーがコンソールを使用して、ある *my-example-widget* リソースに関する詳細情報を表示しようとしたことを想定して、その際に必要な *mediastore:GetWidget* アクセス許可を持っていない場合に発生するものです。

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

この場合、mediastore: *GetWidget* アクションを使用して *my-example-widget* リソースへのアクセスを許可するように、mateojackson ユーザーのポリシーを更新する必要があります。

サポートが必要な場合は、AWS 管理者にお問い合わせください。サインイン資格情報を提供した担当者が管理者です。

iam を実行する権限がありません。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は、ポリシーを更新して iam:PassRole を渡すことができるようにする必要があります MediaStore。

一部の AWS のサービスでは、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する代わりに、そのサービスに既存のロールを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権限が必要です。

次の例のエラーは、という IAM marymajor ユーザーがコンソールを使用して iam:PassRole アクションを実行しようとする場合に発生します MediaStore。ただし、このアクションをサービスが実行するには、サービスロールから付与された権限が必要です。Mary には、iam:PassRole をサービスに渡す権限がありません。

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

この場合、Mary のポリシーを更新して iam:PassRole アクションの実行を許可する必要があります。

サポートが必要な場合は、AWS 管理者にお問い合わせください。サインイン資格情報を提供した担当者が管理者です。

自分の 以外のユーザーに自分の MediaStore リソース AWS アカウント 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人が、リソースにアクセスするために使用できるロールを作成できます。ロールの引き受けを委託するユーザーを指定できます。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:

- がこれらの機能 MediaStore をサポートしているかどうかを確認するには、「」を参照してください [AWS Elemental と IAM MediaStore の連携方法](#)。
- 所有 AWS アカウントしているのリソースへのアクセスを提供する方法については、[IAM ユーザーガイドの「所有 AWS アカウントしている別の IAM ユーザーへのアクセスを提供する」](#)を参照してください。
- リソースへのアクセスをサードパーティーに提供する方法については AWS アカウント、IAM ユーザーガイドの [「サードパーティー AWS アカウントが所有するへのアクセスを提供する」](#)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提供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 \(ID フェデレーション\) へのアクセス権限」](#)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。

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

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。でのログ記録とモニタリングの詳細については、MediaStore 「」を参照してください [AWS Elemental でのモニタリングとタグ付け MediaStore](#)。

モニタリングは、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 AWS Elemental MediaStore 上で重要な部分です。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 アラームを使用すると、指定した期間にわたって 1 つのメトリクスを監視できます。メトリクスが特定のしきい値を超えると、Amazon SNS トピックまたは AWS Auto Scaling policy. CloudWatch alarms は特定の状態にあるため、アクションを呼び出しません。その代わりに、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、「[によるモニタリング CloudWatch](#)」を参照してください。

AWS CloudTrail ログ

CloudTrail は、 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記録を提供します AWS Elemental MediaStore。によって収集された情報を使用して CloudTrail、 に対するリクエスト MediaStore、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](#)」を参照してください。

AWS Trusted Advisor

Trusted Advisor は、数十万の AWS お客様にサービスを提供することから学んだベストプラクティスを活用しています。は AWS 環境 Trusted Advisor を検査し、コスト削減、システムの可用性とパフォーマンスの向上、セキュリティギャップの解消に役立つ機会があればレコメンデーションを作成します。すべての AWS 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 のコンプライアンス検証 MediaStore

AWS のサービス が特定のコンプライアンスプログラムの範囲内にあるかどうかを確認するには、コンプライアンスプログラム[AWS のサービス による対象範囲内のコンプライアンスプログラム](#)を参照し、関心のあるコンプライアンスプログラムを選択します。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。

を使用して、サードパーティーの監査レポートをダウンロードできます AWS Artifact。詳細については、「[でのレポートのダウンロード AWS Artifact](#)」の」を参照してください。

を使用する際のお客様のコンプライアンス責任 AWS のサービス は、お客様のデータの機密性、貴社のコンプライアンス目的、適用される法律および規制によって決まります。では、コンプライアンスに役立つ以下のリソース AWS を提供しています。

- [セキュリティとコンプライアンスのクイックスタートガイド](#) – これらのデプロイガイドでは、アーキテクチャ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 AWS を にデプロイする手順について説明します。
- [アマゾン ウェブ サービスにおける HIPAA セキュリティとコンプライアンスのアーキテクチャー](#) – このホワイトペーパーでは、企業が AWS を使用して HIPAA 対象アプリケーションを作成する方法について説明します。

Note

すべての AWS のサービスが HIPAA の対象となるわけではありません。詳細については、「[HIPAA 対応サービスのリファレンス](#)」を参照してください。

- [AWS コンプライアンスリソース](#) – このワークブックとガイドのコレクションは、お客様の業界や地域に適用される場合があります。
- [AWS カスタマーコンプライアンスガイド](#) – コンプライアンスの観点から責任共有モデルを理解します。このガイドでは、ガイダンスを保護し AWS のサービス、複数のフレームワーク (米国国立標準技術研究所 (NIST)、Payment Card Industry Security Standards Council (PCI)、国際標準化機構 (ISO) を含む) のセキュリティコントロールにマッピングするためのベストプラクティスをまとめています。
- 「[デベロッパーガイド](#)」の「[ルールによるリソースの評価](#)」 – この AWS Config サービスは、リソース設定が社内プラクティス、業界ガイドライン、および規制にどの程度準拠しているかを評価します。AWS Config
- [AWS Security Hub](#) – これにより AWS のサービス、内のセキュリティ状態を包括的に確認できます AWS。Security Hub では、セキュリティコントロールを使用して AWS リソースを評価し、セキュリティ業界標準とベストプラクティスに対するコンプライアンスをチェックします。サポートされているサービスとコントロールのリストについては、「[Security Hub のコントロールリファレンス](#)」を参照してください。
- [Amazon GuardDuty](#) – これにより AWS アカウント、疑わしいアクティビティや悪意のあるアクティビティがないか環境を監視することで、、、ワークロード、コンテナ、データに対する潜在的な脅威 AWS のサービスを検出します。GuardDuty は、特定のコンプライアンスフレームワークで義務付けられている侵入検知要件を満たすことで、PCI DSS などのさまざまなコンプライアンス要件への対応に役立ちます。
- [AWS Audit Manager](#) – これにより AWS のサービス、AWS 使用状況を継続的に監査し、リスクの管理方法と規制や業界標準への準拠を簡素化できます。

AWS Elemental の耐障害性 MediaStore

AWS グローバルインフラストラクチャは、AWS リージョン およびアベイラビリティゾーンを中心に構築されています。物理的に分離および分離された複数のアベイラビリティゾーン AWS リージョン を提供し、低レイテンシー、高スループット、高冗長ネットワークで接続されます。アベイラビリティゾーンでは、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が高く、フォールトトレラントで、スケーラブルです。

AWS リージョン およびアベイラビリティ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。

グローバル AWS インフラストラクチャに加えて、MediaStore には、データの耐障害性とバックアップのニーズをサポートするのに役立ついくつかの機能が用意されています。

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

マネージドサービスである AWS Elemental MediaStore は AWS グローバルネットワークセキュリティで保護されています。AWS セキュリティサービスと [AWS インフラストラクチャ AWS を保護する方法](#) については、[AWS 「クラウドセキュリティ」](#) を参照してください。インフラストラクチャセキュリティのベストプラクティスを使用して AWS 環境を設計するには、「セキュリティの柱 AWS Well-Architected Framework」の [「Infrastructure Protection」](#) を参照してください。

が AWS 公開している API コールを使用して、ネットワーク MediaStore 経由で にアクセスします。クライアントは以下をサポートする必要があります:

- Transport Layer Security (TLS)。TLS 1.2 は必須で TLS 1.3 がお勧めです。
- DHE (楕円ディフィー・ヘルマン鍵共有) や ECDHE (楕円曲線ディフィー・ヘルマン鍵共有) などの完全前方秘匿性 (PFS) による暗号スイート。これらのモードは、Java 7 以降など、ほとんどの最新システムでサポートされています。

また、リクエストには、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、テンポラリ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。

サービス間の混乱した代理の防止

混乱した代理問題は、アクションを実行するためのアクセス許可を持たないエンティティが、より特権のあるエンティティにアクションの実行を強制できてしまう場合に生じる、セキュリティ上の問題です。では AWS、サービス間のなりすましにより、混乱した代理問題が発生する可能性があります。サービス間でのなりすましは、1 つのサービス (呼び出し元サービス) が、別のサービス (呼び出し対象サービス) を呼び出すときに発生する可能性があります。呼び出し元サービスは、本来ならア

アクセスすることが許可されるべきではない方法でその許可を使用して、別のお客様のリソースに対する処理を実行するように操作される場合があります。これを防ぐために、AWSには、アカウント内のリソースへのアクセス権が付与されたサービスプリンシパルですべてのサービスのデータを保護するために役立つツールが用意されています。

リソースポリシーで [aws:SourceArn](#) および [aws:SourceAccount](#) のグローバル条件コンテキストキーを使用して、AWS Elemental がリソースに別のサービスに MediaStore 付与するアクセス許可を制限することをお勧めします。クロスサービスアクセスにリソースを 1 つだけ関連付けたい場合は、[aws:SourceArn](#) を使用します。そのアカウント内のリソースをクロスサービスの使用に関連付けることを許可する場合は、[aws:SourceAccount](#) を使用します。

混乱した代理問題から保護するための最も効果的な方法は、リソースの完全な ARN を指定して、[aws:SourceArn](#) グローバル条件コンテキストキーを使用することです。リソースの完全な ARN が不明な場合や、複数のリソースを指定する場合には、グローバルコンテキスト条件キー [aws:SourceArn](#) で、ARN の未知部分を示すためにワイルドカード文字 (*) を使用します。例えば、`arn:aws:servicename::*:123456789012:*` です。

[aws:SourceArn](#) の値に Amazon S3 バケット ARN などのアカウント ID が含まれていない場合は、両方のグローバル条件コンテキストキーを使用して、アクセス許可を制限する必要があります。

の値は、リージョンとアカウントでの CloudWatch ログ MediaStore を発行する設定 [aws:SourceArn](#) である必要があります。

次の例は、[aws:SourceArn](#) および [aws:SourceAccount](#) グローバル条件コンテキストキーを使用して、混乱した代理問題 MediaStore を回避する方法を示しています。

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

```
    },  
    "StringEquals": {  
      "aws:SourceAccount": "123456789012"  
    }  
  }  
}
```

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

モニタリングは、AWS Elemental とAWSその他のソリューションの信頼性、可用性、MediaStore およびパフォーマンスを維持するための重要な要素です。AWSには、MediaStoreを監視して異常を検出した場合に報告し、必要に応じて自動的に対処するために、次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 コールや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 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ユーザーガイド](#)を参照してください。
- Amazon は、AWSのリソースおよび CloudWatchAWS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、Amazon EC2 インスタンスの CPU CloudWatch 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リソースの変更を記述したシステムイベントのストリームを提供します。通常、AWS CloudWatch サービスはイベントにイベント通知を数秒で配信しますが、1分以上かかる場合もあります。CloudWatch Events 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他のAWS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 を使用すると、Amazon EC2 インスタンス、などからのログファイルのモニタリング、保存 CloudTrail、アクセスができます。CloudWatch ログは、ログファイル内の情報をモニタリングし、特定のしきい値が満たされたときに通知します。高い耐久性を備えたストレージにログデータをアーカイブすることも可能です。詳細については、[Amazon CloudWatch Logs ユーザーガイド](#)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

- [を使用した AWS Elemental MediaStore API コールのログ記録 AWS CloudTrail](#)
- [アマゾン AWS Elemental MediaStore ルのモニタリング CloudWatch](#)
- [AWS Elemental MediaStore リソースのタグ付け](#)

を使用した AWS Elemental MediaStore API コールのログ記録 AWS CloudTrail

AWS Elemental MediaStore は AWS CloudTrail、でユーザー、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を記録するサービスであると統合されています MediaStore。CloudTrail MediaStore コンソールからの呼び出しや API へのコード呼び出しを含む、の MediaStore API MediaStore コールのサブセットをイベントとしてキャプチャします。証跡を作成すると、CloudTrail のイベントなど、Amazon S3 バケットへの継続的な配信を有効にすることができます MediaStore。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。によって収集された情報を使用して CloudTrail、に対して行われたリクエスト MediaStore、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。

設定や有効化の方法などの詳細については、[AWS CloudTrail ユーザーガイドを参照してください](#)。

CloudTrail

トピック

- [AWS Elemental MediaStore 情報 CloudTrail](#)
- [例: AWS Elemental MediaStore ログファイルエントリ](#)

AWS Elemental MediaStore 情報 CloudTrail

CloudTrail は、AWS アカウント作成時にアカウントで有効になります。AWS Elemental でサポートされているイベントアクティビティが発生すると MediaStore、そのアクティビティは [Event history (イベント履歴)] (CloudTrail イベント履歴) で AWS のその他サービスのイベントと共にイベントに記録されます。AWS アカウントで最近のイベントを表示、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。

AWS のイベントなど、MediaStore アカウントのイベントの継続的なレコードについては、追跡を作成します。追跡により、CloudTrail 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、コンソールで追跡を作成するときに、追跡がすべての AWS リージョンに適用されません。追跡は、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をログに記録し、指定した Amazon S3 バケットにログファイルを配信します。さらに、AWS CloudTrail より詳細な分析と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サービスを設定できます。詳細については、次のトピックを参照してください。

- [追跡を作成するための概要](#)
- [CloudTrail サポート対象のサービスとインテグレーション](#)
- [の Amazon SNS 通知の設定 CloudTrail](#)
- [CloudTrail 複数のリージョンからのログファイルの受け取りおよび CloudTrail 複数のアカウントからのログファイルの受け取り](#)

AWS Elemental MediaStore は、CloudTrail 以下のオペレーションをログファイルのイベントとしてログファイルにログに記録します。

- [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。同一性情報は次の判断に役立ちます。

- リクエストが、ルートユーザーと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 の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 ログファイルエントリ

追跡は、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ようにする設定です。CloudTrail ログファイルには、1 つ、または複数のログエントリが含まれています。イベントはあらゆるソース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などの情報が含まれます。CloudTrail ログファイルは、公開 API コール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。

CloudTrail CreateContainer オペレーションを示すログエントリの例は、次のとおりです。

```
{
  "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":  
    "MNCTGH4HRQJ27GRMBVDPIVHEP4L02BN6MUVHBCPSH0AWNS0KSXC024B2UE0BBND5D0NRXTMFK3TOJ4G7AHWMESI",  
  "eventID": "7085b140-fb2c-409b-a329-f567912d704c",  
  "eventType": "AwsApiCall",  
  "recipientAccountId": "111122223333"  
}
```

アマゾンAWS Elemental MediaStore ルのモニタリング CloudWatch

raw データを収集して読み取り可能なメトリクスに処理する AWS Elemental MediaStore を使用して CloudWatch をモニタリングできます。CloudWatch 履歴情報にアクセスし、ウェブアプリケーションまたはサービスの動作をよりの確に把握するために、15 か月間統計が保持されます。また、特定のしきい値をモニタリング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報告し、必要に応じて自動的に対処するために、次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 ElementalAWS などのサービスのログファイルを監視、保存、およびアクセスできます MediaStore。ログを使用すると、CloudWatch 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 では、アプリケーションログに存在する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信することができます。CloudWatch ログデータが 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、アプリケーションログの特定のリテーム (例:ValidationException「」) をモニタリングしたり、PutObject特定の期間に行われたリクエストの数をカウントしたりできます。検索した語句が見つかったら、CloudWatch Logs CloudWatch は指定された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 Events は、MediaStore オブジェクトなどの CloudWatch 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。通常、AWS CloudWatch サービスはイベントにイベント通知を数秒で配信しますが、1分以上かかる場合もあります。イベント (DeleteObjectリクエストなど) に一致するルールを設定して、これらのイベントを1つ以上のターゲット関数またはストリーム

にルーティングできます。CloudWatch オペレーションの変更が発生すると、イベントはその変更を認識します。また、CloudWatch Events 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、是正措置を実行します。

CloudWatch ログ

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 ログ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 などの名前のロググループが、「」というロググループが、「」/aws/mediastore/MyContainerName になります。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 のアクセス許可のセットアップ CloudWatch

AWS Identity and Access Management(IAM) を使用して、Amazon MediaStore へのアクセス権を AWS Elemental に付与するロールを作成します CloudWatch。CloudWatch アカウントに対してログを公開するには、これらのステップを実行する必要があります。CloudWatch は、お使いのアカウントに関するメトリクスを自動的に発行します。

MediaStore へのアクセスを許可するには CloudWatch

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] を選択し、[ポリシーの作成] を選択します。
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/*"
  }
]
```

このポリシーでは MediaStore、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、ロググループとログストリームを作成できます。

4. [Review policy] (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、[ルール]、[ルールを作成] 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: Permissions] (次のステップ: アクセス許可) を選択します。
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、[Next: Tags (次の手順: タグ)] を選択します。
12. [Next: Review (次の手順: 確認)] を選択し、新しいユーザーをプレビューします。
13. [ルール名] に「**MediaStoreAccessLogs**」と入力し、[ルールの作成] 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] (信頼関係の編集) をクリックします。
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。プリンシパルは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] (信頼ポリシーの更新) をクリックします。

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 はアクセスログを収集しません。コンテナでアクセスのログ記録を有効にすると、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 MediaStore 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 は Amazon MediaStore 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 でのアクセスログのトラブルシューティング MediaStore

AWS Elemental MediaStore のアクセスログが Amazon に表示されない場合は CloudWatch、以下の表に記載されている考えられる原因と解決策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
<p>CloudTrail ログが有効になっていても、CloudTrail イベントがまったく表示されない。</p>	<p>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せん。</p>	<p>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 “の許可を設定する CloudWatch” を参照してください。</p>
<p>DescribeContainer API リクエストを送信したが、AccessLoggingEnabled パラメータに False の値が含まれていることがレスポンスに示されている。また、MediaStoreAccessLogs ロールが DescribeLogGroup、CreateLogGroup、DescribeLogStream、または CreateLogStream の呼び出しに成功したことを示す CloudTrail イベントが全く表示されない。</p>	<p>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せん。</p> <p>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。</p>	<p>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 “の許可を設定する CloudWatch” を参照してください。</p> <p>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” を参照してください。</p>
<p>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る。CloudTrail イベントに、次のような行が含まれている場合がある。</p> <pre> "eventSource": "logs.amazonaws.com", "errorCode": "AccessDenied", "errorMessage": "User: arn:aws:sts::11112223333:assumed-role/ MediaStoreAccessLogs/ </pre>	<p>IAM ロールに AWS Elemental に対する正しいアクセス許可がありません MediaStore。</p>	<p>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 “の許可を設定する CloudWatch” を参照してください。</p>

症状	考えられる原因	解決策
<pre>MediaStoreAccessLogsSession is not authorized to perform: logs:DescribeLogGroups on resource: arn:aws:logs:us-west-2:111122223333:log-group::log-stream:",</pre>		
<p>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。</p>	<p>お客様のアカウントが、1アカウント、1アカウント、1アカウント、1アカウント、1 CloudWatch アカウント、1アカウント、1アカウント、1アカウント、1アカウント、1アカウント、1アカウント、1アカウント、1「Amazon CloudWatch Logs ユーザーガイド」のロググループのクォータを参照してください。</p>	<p>CloudWatch コンソールで、CloudWatch アカウントがロググループ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。</p>

症状	考えられる原因	解決策
一部のログが表示されますが CloudWatch、予期しているすべてのログではなく、すべてのログしか表示されない。	お客様のアカウントが、1 秒、1 秒、1 アカウント、1 秒、1 アカウント、1 秒、1 CloudWatch アカウント、1 秒、1 アカウント、1 秒、1 アカウント、1 秒、1 アカウント、1 秒、1 秒、 Amazon CloudWatch LogsPutLogEvents ユーザーガイドのクォータを参照してください 。	1 秒、 1 秒 、 1 アカウント 、1 アカウント、CloudWatch 1 アカウント、1 アカウント、1 アカウント、1 アカウント、1 秒

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":
"aaaAAA111bbbBBB222cccCCC333dddDDD444eeeEEE555ffffFFF666gggGGG777hhhHHH888iiiIII999jjjJJJ",
  "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":
"dddDDD444eeeEEE555ffffFFF666gggGGG777hhhHHH888iiiIII999jjjJJJ000cccCCC333bbbBBB222aaaAAA",
  "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 エラーコード (など `InternalServerError`)。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に含まれている [transient data rule](#) によって設定された有効期限に基づいています。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含まれていない場合、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 null です。このフィールドは、PutObject、GetObject、DescribeObject、および DeleteObject の各オペレーションにのみ適用されます。

HTTPStatus

レスポンスの HTTP ステータスの数値。

Operation

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

[Path] (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。

リクエスト

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 anonymous になります。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

MediaStore 各リクエストを一意に識別するために AWS Elemental 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ト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 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、配信すらされなかったりすることもあり得ます。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 Management Console](#) 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の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。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。

Important

通常、AWS CloudWatch サービスはイベントにイベント通知を数秒で配信しますが、1分以上かかる場合もあります。

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”](#)
2. [the section called “コンテナ状態の変更イベント”](#)

これらのイベントのサブスクライブについては、「[Amazon](#)」を参照してください CloudWatch。

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 CloudWatch MediaStore でイベントを使用する例をいくつか次に示します。

- 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](#)
- [AWS Elemental MediaStore コンテナ状態の変更イベント](#)

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。

Note

一時的なデータルールのために期限切れになるオブジェクトは、CloudWatch有効期限が切れたときにイベントを発行しません。

このイベントのサブスクライブについては、「[Amazon](#)」を参照してください CloudWatch。

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。このイベントのサブスクライブについては、「[Amazon](#)」を参照してください CloudWatch。

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 メトリックスAWS Elemental MediaStore へのモニタリング

raw データを収集して読み取り可能なメトリクスに処理する AWS Elemental MediaStore を使用して CloudWatch をモニタリングできます。CloudWatch 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またはサービスの動作をよりの確に把握することができます。また、特定のしきい値をモニタリング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 では MediaStore、メトリクスが特定のしきい値に達したときに、BytesDownloaded を監視して自分自身に 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 Management Consoleにログインし、<https://console.aws.amazon.com/cloudwatch/CloudWatch> で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. [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 Metrics

次の表に、AWS Elemental MediaStore で送信されるメトリクスを示します CloudWatch。

Note

メトリクスを表示するには、[コンテナにメトリクスポリシーを追加](#)して、メトリクスを Amazon MediaStore に送信できるようにする必要があります CloudWatch。

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)。 単位はカウント 有効なディメンション:

メトリクス	説明
	<ul style="list-style-type: none">• コンテナ名• オブジェクトグループ名• リクエストタイプ <p>有効な統計: Sum</p>
4xxErrorCount	<p>これに対して行われた HTTP MediaStore リクエストの数が、4xx エラーを発生させました。</p> <p>単位はカウント</p> <p>有効なディメンション:</p> <ul style="list-style-type: none">• コンテナ名• オブジェクトグループ名• リクエストタイプ <p>有効な統計: Sum</p>
5xxErrorCount	<p>これに対して行われた HTTP MediaStore リクエストの数が、5xx エラーを発生させました。</p> <p>単位はカウント</p> <p>有効なディメンション:</p> <ul style="list-style-type: none">• コンテナ名• オブジェクトグループ名• リクエストタイプ <p>有効な統計: Sum</p>

メトリクス	説明
BytesUploaded	<p>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。</p> <p>単位: バイト</p> <p>有効なディメンション:</p> <ul style="list-style-type: none">• コンテナ名• オブジェクトグループ名 <p>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</p>
BytesDownloaded	<p>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。</p> <p>単位: バイト</p> <p>有効なディメンション:</p> <ul style="list-style-type: none">• コンテナ名• オブジェクトグループ名 <p>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</p>

メトリクス	説明
TotalTime	<p>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 リクエストを受信してから、レスポンスの最終バイトを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。</p> <p>単位: ミリ秒</p> <p>有効なディメンション:</p> <ul style="list-style-type: none">• コンテナ名• オブジェクトグループ名• リクエストタイプ <p>有効な統計情報: 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</p>
TurnaroundTime	<p>MediaStore 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 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を送信するまでの時間を計測した値です。</p> <p>単位: ミリ秒</p> <p>有効なディメンション:</p> <ul style="list-style-type: none">• コンテナ名• オブジェクトグループ名• リクエストタイプ <p>有効な統計情報: 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</p>

メトリクス	説明
ThrottleCount	<p>MediaStore これに対して行われた HTTP リクエストの数がスロットリングされました。</p> <p>単位はカウント</p> <p>有効なディメンション:</p> <ul style="list-style-type: none"> • コンテナ名 • オブジェクトグループ名 • リクエストタイプ <p>有効な統計: Sum</p>

AWS Elemental MediaStore リソースのタグ付け

タグとは、お客様または 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。タグキーと同様に、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 リソースを識別および整理します。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までリソースの関連を示すことができリソースに同じタグを割り当てる事ができます。たとえば、AWS Elemental MediaLive入力に割り当てる同じタグを AWS Elemental MediaStore **#####**。
- 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 ユーザーガイド」](#)の [「コスト配分タグの使用」](#) (Use Cost Allocation Tags) を参照してください。

以下のセクションでは、AWS Elemental MediaStore のタグに関する詳細情報を提供します。

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

AWS Elemental MediaStore の以下のリソースはタグ付けをサポートしています。

- #####

タグの追加と管理の詳細については、「[タグの管理](#)」を参照してください。

AWS Elemental は AWS Identity and Access Management (IAM) MediaStore 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、以下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 対応の文字、数字、スペースと、文字 (. : + = @ _ / -) (ハイフン) です。Amazon EC2 リソースでは、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または削除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。

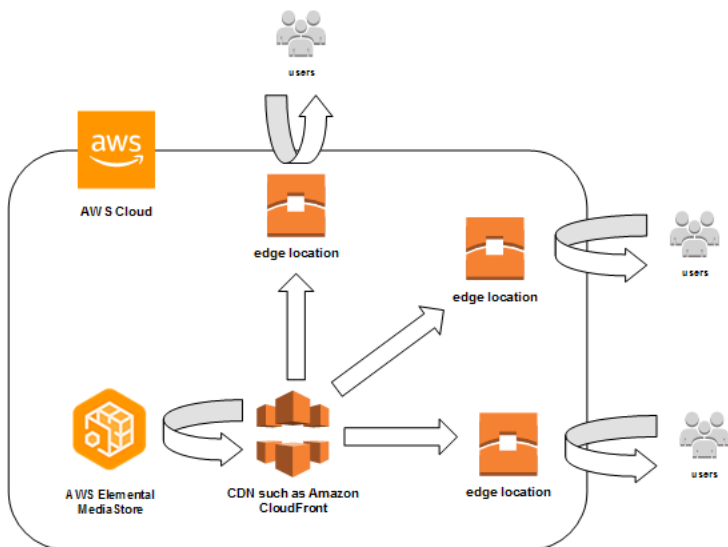
タグの管理

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。タグの使用方法については、『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

AWS Elemental に保存したコンテンツは、[Amazon CloudFront](#) などのコンテンツ配信ネットワーク (CDN) を使用して配信できます。MediaStore。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

- [Amazon CloudFront に AWS Elemental MediaStore コンテナへのアクセスを許可する](#)
- [AWS Elemental による HTTP MediaStore キャッシュの操作](#)

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

Amazon CloudFront を使用して、AWS Elemental に保存したコンテンツを配信することができます。MediaStore。これには以下の 2 つの方法があります。

- [オリジンアクセス制御 \(OAC\) の使用](#)-(推奨) の OACAWS リージョン 機能をサポートしている場合は、このオプションを使用してください CloudFront。

- [共有シークレットの使用](#)-の OACAWS リージョン 機能をサポートしていない場合は、このオプションを使用してください CloudFront。

オリジンアクセス制御 (OAC) の使用

Amazon のオリジンアクセスコントロール (OAC) 機能を使用すると、CloudFront セキュリティを強化して AWS Elemental MediaStore オリジンを保護できます。[AWS MediaStore オリジンのリクエストで署名バージョン 4 \(SIGv4\)](#) を有効にし、CloudFront リクエストに署名するタイミングと署名するかどうかを設定できます。の OAC 機能には、コンソール、API、SDK、または CLI CloudFront からアクセスでき、その使用に追加料金はかかりません。

で OAC 機能を使用する方法の詳細については MediaStore、[Amazon CloudFront 開発者ガイドの「MediaStore オリジンへのアクセスの制限」](#) を参照してください。

共有シークレットの使用

が Amazon の OACAWS リージョン 機能をサポートしていない場合は CloudFront、読み取りアクセス権限以上を付与するポリシーを AWS Elemental MediaStore コンテナにアタッチできます CloudFront。

Note

OACAWS リージョン 機能をサポートしている場合は、OAC 機能を使用することをお勧めします。以下の手順では、MediaStore コンテナへのアクセスを制限するために、MediaStore CloudFront および共有シークレットを設定する必要があります。セキュリティのベストプラクティスに従うには、この手動設定ではシークレットを定期的にローテーションする必要があります。MediaStore オリジンに OAC を使用すると、SIGv4 を使用してリクエストに署名し、CloudFront MediaStore 署名照合のために転送するように指示できるため、シークレットを使用したりローテーションしたりする必要がなくなります。これにより、メディアコンテンツが配信される前にリクエストが自動的に検証され、MediaStore CloudFront メディアコンテンツの配信がより簡単で安全になります。

CloudFront コンテナへのアクセスを許可するには (コンソール)

1. <https://console.aws.amazon.com/mediastore/> [MediaStore](#) でコンソールを開きます。
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. 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以上を付与するポリシーをアタッチします。

Example

次のポリシー例は、HTTPS [GetObject](#) 経由の公開読み取りアクセスのポリシー例と似ていますが、[HTTPSDescribeObject](#) 経由でドメインにリクエストを送信するユーザーからのコマンドを許可し、これらの要件に適合しています。さらに、次のサンプルポリシーでは、リクエストが HTTPS 接続を介して発生し、正しい Referer CloudFront MediaStore ヘッダーが含まれている場合にのみオブジェクトへのアクセスを許可するため、ワークフローのセキュリティが強化されます。

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

4. [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。

- 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 [コンテナ] ページの [情報] セクションにあります。では CloudFront、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。では CloudFront、これをオリジンパスと呼びます。ただし、この設定は省略可能です。オリジンパス詳細については、[Amazon CloudFront デベロッパガイド](#)を参照してください。
- でディストリビューションを作成し CloudFront、[AWS Elemental のコンテンツを配信するように設定 MediaStore](#)します。前のステップで収集した情報が必要です。

MediaStore ポリシーをコンテナにアタッチしたら、オリジンリクエストに HTTPS CloudFront 接続のみを使用するように設定し、正しいシークレット値を含むカスタムヘッダーを追加する必要があります。

Referer ヘッダー (コンソール) CloudFront にシークレット値を指定して HTTPS 接続経由でコンテナにアクセスするように設定するには

- CloudFront コンソールを開きます。
- オリジンページで、MediaStoreオリジンを選択します。
- [編集] を選択します。
- プロトコルには HTTPS のみを選択してください。
- [カスタムヘッダーの追加] セクションで、[ヘッダーを追加] を選択します。
- [名前] に [リファラー] を選択します。値には、<secretValue>コンテナポリシーで使用したのと同じ文字列を使用してください。
- [保存] を選択し、変更を適用します。

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

AWS Elemental は、Amazon などのコンテンツ配信ネットワーク (CDN) MediaStore CloudFront によって正しく効率的にキャッシュできる方法で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、オブジェクトをキャッシュするためのキーとして使用します。MediaStore オブジェクトがアップロードされると、ETag 各オブジェクトにを各オブジェクトに自動的に生成します。[オブジェクトの詳細を表示](#)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 オブジェクトがアップロードされると、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](#) または [API MediaStore](#) を使用してオブジェクトをコンテナに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 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

への条件付きリクエスト MediaStore

MediaStore 条件付きリクエスト ([RFC7232](#) に記載されているように If-Modified-Since If-None-Match、やなどのリクエストヘッダーを使用) と無条件要求に対して同じように応答します。つまり、MediaStore GetObject 有効な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

に代わってコンテンツを配信する CDN は、[RFC7232 セクション 4.1](#)に記載されているように、MediaStore 条件付きリクエストを処理できます。304 Not Modifiedこれは、リクエスト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 繰り返し取得されたオブジェクトの更新をオリジンサーバーにクエリする頻度を制御するには、Cache-Controlオブジェクトをにアップロードするときにそれらのオブジェクトのヘッダーを設定します MediaStore。

AWS Elemental でのクォータ MediaStore

Service Quotas 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。

次の表に、AWS Elemental でのクォータ (以前は制限と呼ばれていた) を示します MediaStore。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 Support 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。このオプションは、コンテナ間でアカウントレベルの制限を分割して、1つのコンテナがクォータ全体を使い果たさないようにするのに役立ちます。

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。
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数。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。
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限り、無制限の数のフォルダを作成できます。
Object Size	25 MB	1つのオブジェクトの最大ファイルサイズ。
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数のオブジェクトをアップロードできます。
DeleteObject API リクエストのレート	100	1秒あたりに可能なオペレーションリクエストの最大数。追加のリ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
DescribeObject API リクエストのレート	1,000	1 秒あたりに可能なオペレーションリクエストの最大数。追加のリ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 標準のアップロード可用性の API リクエストのレート	1,000	1 秒あたりに可能なオペレーションリクエストの最大数。追加のリ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可用性の API リクエストのレート	25	1 秒あたりに可能なオペレーションリクエストの最大数。追加のリ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 API リクエストのレート	5	1 秒あたりに可能なオペレーションリクエストの最大数。追加のリ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。
PutObject チャンク転送エンコーディング用の API リクエストのレート (ストリーミングアップロード可用性とも呼ばれます)	10	1 秒あたりに可能なオペレーションリクエストの最大数。追加のリ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。リクエストで、リクエストされた TPS と平均オブジェクトサイズを指定します。
PutObject 標準のアップロード可用性の API リクエストのレート	100	1 秒あたりに可能なオペレーションリクエストの最大数。追加のリ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。リクエストで、リクエストされた TPS と平均オブジェクトサイズを指定します。

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
メトリクスポリシーのルール	10	メトリクスポリシーに含めることができるルールの最大数。
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ることのできるルールの最大数。

AWS Elemental Mage MediaStore の関連情報

AWS Elemental を使用する際に役立つ関連リソースのリストです MediaStore。

- [クラスとワークショップ](#) – AWS のスキルを磨き、実践的な経験が得るために役立つセルフペースラボに加えて、ロールベースのコースと特別コースへのリンクです。
- [AWS デベロッパーセンター](#) – チュートリアルを検索、ツールのダウンロード、AWS デベロッパーイベントの確認を行います。
- [AWS デベロッパーツール](#) – AWS アプリケーションを開発および管理するためのデベロッパーツール、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。
- [ご利用開始のためのリソースセンター](#) – AWS アカウント をセットアップする方法、AWS コミュニティに参加する方法、最初のアプリケーションを起動する方法を説明します。
- [step-by-step ハンズオンチュートリアル](#) – AWS に関する情報のメインウェブページです。
- [AWS ホワイトペーパー](#) – アーキテクチャ、セキュリティ、エコノミクスなどのトピックについて、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が記述した AWS の技術ホワイトペーパーの包括的なリストへのリンクです。
- [AWS Support Center](#) – AWS Support のケースを作成して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、サービスヘルスステータス、AWS Trusted Advisor など、他の役立つリソースへのリンクも含まれています。
- [AWS Support](#) – AWS Support に関する情報のメインウェブページです。one-on-oneクラウド上のアプリケーションの構築および実行を支援するために 1 対 1 での迅速な対応を行うサポートチャネルとして機能します。
- [お問い合わせ](#) - AWS の請求、アカウント、イベント、不正使用、その他の問題などに関するお問い合わせの受付窓口です。
- [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、その他のトピックに関する詳細情報。

ユーザーガイドのドキュメント履歴

次の表は、AWS Elemental MediaStore の今回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通知については、RSS フィードでサブスクライブできます。

変更	説明	日付
オリジンアクセス制御 (OAC) の改善	AWS Elemental で OAC MediaStore を使用する方法に関する情報を追加しました。	2023 年 4 月 17 日
クォータの更新	Rules in a Metric Policyのクォータ値と説明が修正されました。	2022 年 10 月 25 日
ExpiresAt フィールド	アクセスログに、コンテナのライフサイクルポリシー内の一時データの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。	2020 年 7 月 16 日
ライフサイクル移行ルール	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。	2020 年 4 月 20 日
空のコンテナ	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。	2020 年 4 月 7 日
Amazon CloudWatch メトリクスのSupport	メトリクスポリシーを設定して、MediaStore メトリク	2020 年 3 月 30 日

スの送信先を指定できません
CloudWatch。

[オブジェクト削除ルールのワ イルドカード](#)

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。

2019 年 12 月 20 日

[オブジェクトライフサイクル ポリシー](#)

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。

2019 年 9 月 13 日

[AWS CloudFormation](#) のサ ポート

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 CloudForm
ation 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。

2019 年 5 月 17 日

ストリーミングアップロードの可否に関するクォータ	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。	2019 年 4 月 8 日
オブジェクトのチャンク転送	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。	2019 年 4 月 5 日
アクセスロギング	AWS Elemental で、MediaStore アクセスのログ記録が行えるようになりました。これにより、コンテナ内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できます。	2019 年 2 月 25 日
オブジェクトライフサイクルポリシー	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。	2018 年 12 月 12 日
オブジェクトサイズクォータの増加	オブジェクトサイズのクォータは 25 MB になりました。	2018 年 10 月 10 日
オブジェクトサイズクォータの増加	オブジェクトサイズのクォータは 20 MB になりました。	2018 年 9 月 6 日

AWS CloudTrail の統合	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。	2018 年 7 月 12 日
CDN コラボレーション	Amazon などのコンテンツ配信ネットワーク (CDN) MediaStore で AWS Elemental CloudFront を使用する方法に関する情報を追加しました。	2018 年 4 月 14 日
CORS コンフィギュレーション	AWS Elemental でクロスオリジンリソース共有 (CORS) MediaStore 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。	2018 年 2 月 7 日
新しいサービスとガイド	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 と「AWS Elemental MediaStore MediaStore ユーザーガイド」の最初のリリースです。	2017 年 11 月 27 日

Note

- AWS Media Services は、サービスの非可用性、中断、または障害が、死亡、人身傷害、財産上の損害、または環境上の損害につながる可能性がある、人命安全業務、ナビゲーションもしくは通信システム、航空交通管制、または生命維持装置などのフェイルセーフ性能を必要とするアプリケーションまたは状況での使用のために設計されておらず、そのような使用を目的とするものでもありません。

AWS 用語集

AWS の最新の用語については、「AWS の用語集リファレンス」の「[AWS 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。