



ユーザーガイド

AWS Elemental MediaTailor



AWS Elemental MediaTailor: ユーザーガイド

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

Amazon の商標およびトレードドレスは Amazon 以外の製品およびサービスに使用することはできません。また、お客様に誤解を与える可能性がある形式で、または Amazon の信用を損なう形式で使用することもできません。Amazon が所有していないその他のすべての商標は Amazon との提携、関連、支援関係の有無にかかわらず、それら該当する所有者の資産です。

Table of Contents

とは AWS Elemental MediaTailor	1
オリジンサーバーの要件	1
MediaTailor の概念	2
HLS プレイリストタイプ	2
DASH マニフェストタイプ	4
広告挿入の概念	10
チャンネルアセンブリの概念	11
広告挿入の仕組み	12
広告挿入イベントフロー	13
関連サービス	17
MediaTailor へのアクセス	17
料金	17
リージョン	18
設定	19
にサインアップする AWS アカウント	19
MediaTailor の使用開始	20
MediaTailor 広告挿入の使用開始	20
前提条件	21
ステップ 1: MediaTailor にアクセスする	21
ステップ 2: ストリームを準備する	21
ステップ 3: ADS リクエスト URL とクエリパラメータを設定する	23
ステップ 4: 設定を作成する	24
ステップ 5: 設定をテストする	26
ステップ 6: MediaTailor に再生リクエストを送信する	27
(オプション) ステップ 7: アクティビティをモニタリングする	28
ステップ 8: クリーンアップする	30
MediaTailor チャンネルアセンブリの使用開始	30
前提条件	31
ステップ 1: ソースロケーションを作成する	31
ステップ 2: VOD ソースをソースロケーションに追加する	32
ステップ 3: チャンネルを作成する	33
ステップ 4: チャンネルのスケジュールにプログラムを追加する	35
ステップ 5 (オプション): MediaTailor を使用してパーソナライズされた広告をストリームに挿入する	37

ステップ 6: チャンネルの開始	38
ステップ 7: チャンネルのテスト	38
ステップ 8: クリーンアップする	38
広告の挿入	40
サポートされているオーディオおよびビデオコーデック	41
広告挿入動作について	41
VOD に対する広告ステッチ動作	41
ライブ広告ステッチング動作	44
サーバーガイド広告挿入の概要	47
再生設定で を有効にする	47
サーバーガイドセッションを作成する	47
SGAI による広告追跡	49
マニフェストハートビートを使用したガイド付きプリフェッチ	49
SGAI 機能の互換性	50
SGAI ライブ設定	53
SGAI VOD 設定	56
広告サーバー統合要件	60
VAST 要件	60
VPAID 要件	63
設定管理	64
設定の作成	64
設定の詳細の表示	72
設定の編集	72
設定の削除	72
コンテンツソースの統合	73
入カファイル要件	73
HLS ソースの統合	74
MPEG-DASH ソースの統合	85
SigV4 とのオリジンインタラクションの保護	104
Google の広告プラットフォームとの統合	108
サーバー側の統合	109
クライアント側の統合	110
広告ブレイク抑制による広告ブレイク動作のカスタマイズ	111
広告ブレイク抑制の設定	112
バンパー広告挿入	124
バンパーの設定	124

プレロール広告挿入	126
スレート広告挿入	127
スレートの設定	127
パーソナライゼーションしきい値の仕組み	128
スレート設定と VPAID	131
広告トランスコード	132
ダイナミックトランスコーディング	132
カスタムトランスコードプロファイル	134
広告のプリフェッチ	138
プリフェッチの仕組み	139
プリフェッチスケジュールの作成	144
TPS ベースのトラフィックシェーピング	150
プリフェッチスケジュールの削除	151
前提条件付き広告	152
前提条件付き広告の要件	152
前提条件付き広告ワークフロー	155
収益化関数	156
関数が MediaTailor ワークフローにどのように適合するか	157
「クイックスタートガイド」	158
ライフサイクルフック	162
関数のタイプと構成	170
の作成と管理	180
JSONata 式リファレンス	184
関数の例	194
トラブルシューティングとモニタリング	198
制限	207
動的広告変数	210
セッション変数	212
プレイヤー変数	225
ドメイン変数	227
設定エイリアス	228
ADS パラメータを渡す	236
パラメータルーティング	239
MediaPackage の統合	241
セッションの動作	243
パラメータリファレンス	244

パラメータのトラブルシューティング	246
エイリアスのトラブルシューティング	250
MediaTailor マニフェストクエリパラメータ	253
オリジンクエリパラメータ	254
MediaTailor の文字制限	255
MediaTailor HLS の暗黙的なセッション	257
MediaTailor DASH の暗黙的なセッション	258
MediaTailor の明示的なセッション初期化	260
MediaTailor のプロトコル固有の動作	262
MediaTailor CDN 統合	264
広告追跡データの報告	266
サーバー側の追跡	266
クライアント側の追跡	282
オーバーレイ広告	358
オーバーレイ広告を使用するための前提条件	359
開始方法	360
ログ記録とメトリクス	373
MediaTailor でのオーバーレイ広告の請求	375
広告 ID デコレーション	375
セッションの状態	377
マニフェストと広告メタデータの挿入	378
広告決定サーバー (ADS) インタラクション	402
クライアント側の追跡 API	403
線形アセンブルストリームの作成	406
ソースロケーションの使用	406
ソースロケーションの作成	407
ソースロケーション用の認証の設定	409
VOD ソースの使用	418
ライブソースの使用	422
パッケージ設定の使用	427
マニフェストキャッシュ	428
チャンネルの使用	428
チャンネルの作成	428
チャンネルの出力でのソースグループの使用	431
チャンネルを削除する	431
プログラムの追加	432

プログラムの作成	432
オーディエンスコホートと代替コンテンツの定義	440
対象者固有のマニフェストの生成	443
広告と広告時間枠を挿入する	444
広告挿入を設定する	444
広告時間枠の SCTE-35 メッセージ	445
チャンネルの再生のタイムシフト	451
CDN の使用	453
CDN 選択	454
CDN 統合を計画する	455
トラフィック要件の見積もり	457
最適化戦略を設定する	458
計画をカスタマイズする	459
モニタリングとスケーリングを設定する	462
コストの最適化	463
実装をテストする	465
一般的な問題のトラブルシューティング	467
CDN 統合をセットアップする	469
CDN 統合コンポーネントと要件	470
CDN 統合の前提条件	21
CDN 統合のセットアップ手順	471
MediaTailor CDN 統合に必要なヘッダー	471
CDN ルーティング動作を設定する	473
CDN マッピングを設定する	477
セキュリティのベストプラクティス	480
CDN を使用した広告挿入	481
必要なもの	482
[開始する前に]	482
CDN 統合の利点	483
CDN アーキテクチャを理解する	484
基本的な広告挿入を設定する	491
チャンネルアセンブリを使用した SSAI	495
CDN パフォーマンスの最適化	502
CDN オペレーションのモニタリング	504
CDNs トラブルシューティング	509
CDN を使用したチャンネルアセンブリ	513

必要なもの	514
[開始する前に]	514
CDN 統合の利点	515
CDN アーキテクチャを理解する	516
基本セットアップ	519
ベース URLsを設定する	519
広告挿入の実装	520
タイムシフト表示を設定する	523
CDN オペレーションのモニタリング	524
完全な最適化ガイド	525
MediaPackage CDN 統合	525
MediaPackage と CDN ワークフローについて	526
前提条件	527
基本設定の設定	527
統合の検証	529
次の手順	531
マニフェストでのフィルタリング	531
CDN キャッシュ	537
CDN パフォーマンス	542
CDN 統合のトラブルシューティング	547
CloudFront 統合	552
CloudFront の基本的なセットアップ	552
パフォーマンスの最適化	559
マルチリージョンのレジリエンス	563
モニタリングとトラブルシューティング	567
サードパーティー CDN のセットアップ	572
前提条件	572
CDN キャッシュルールを設定する	573
CDN 設定を確認する	576
CDN パフォーマンスの最適化	577
サードパーティーの CDN 問題のトラブルシューティング	578
CDN パフォーマンスの最適化	580
キャッシュの最適化	581
リクエストルーティングの最適化	585
パフォーマンスベンチマーク	586
高度な最適化	587

CDN モニタリング	590
重要な CDN パフォーマンスメトリクス	590
モニタリングツールのセットアップ	592
実装チェックリスト	594
関連トピック	594
CDN 統合テスト	595
テストの前提条件	595
体系的なテスト方法	596
デプロイ前チェックリスト	601
テストツールのリファレンス	602
CDN 統合のトラブルシューティング	603
クイック診断チェックリスト	604
インテグレーションテスト	606
マニフェスト 404 エラー	609
マニフェスト配信の問題	610
セグメント配信の問題	612
セッション管理の問題	614
広告時間枠のタイミングの問題	616
パフォーマンスとレイテンシーの問題	618
デバイスの動作に一貫性がない	625
トラブルシューティングの準備	625
ワークフロー固有のトラブルシューティングガイド	628
CDN 統合ログ分析リファレンス	629
CDN ログの解釈	630
エラーコードリファレンス	632
ログ分析ツール	634
MediaTailor ログ記録設定リソース	634
CloudFormation Automation	635
を使用する理由 AWS CloudFormation	636
デプロイの準備	637
テンプレートをデプロイします。	640
デプロイされたリソースを使用する	641
デプロイのテストと検証	644
のデプロイ問題のトラブルシューティング	645
テンプレートをカスタマイズする	647
テンプレートリファレンス	651

本番稼働用 CloudFront 設定	656
3 オリジンアーキテクチャ	656
キャッシュ動作の優先順位と設定	658
CDN 統合のサポートを受ける	664
サポートにエスカレーションするタイミング	665
サポート用情報の収集	665
サポートリソース	666
サポートケースのベストプラクティス	667
セキュリティ	668
データ保護	669
データ暗号化	670
Identity and Access Management	670
オーディエンス	671
アイデンティティを使用した認証	671
ポリシーを使用したアクセスの管理	672
が IAM と AWS Elemental MediaTailor 連携する方法	674
アイデンティティベースのポリシーの例	679
リソースベースのポリシーの例	682
AWS マネージドポリシー	683
サービスにリンクされたロールの使用	685
ID とアクセスのトラブルシューティング	688
コンプライアンス検証	691
耐障害性	691
インフラストラクチャセキュリティ	692
サービス間での混乱した使節の防止	692
ログ記録とモニタリング	693
CloudWatch アラーム	694
CloudTrail ログ	694
AWS Trusted Advisor	694
モニタリングとタグ付け	695
ログの表示	696
ADS ログ	696
マニフェストログ	754
トランスコードログ	759
販売ログの使用	762
CloudWatch Logs へのログの書き込み	766

広告挿入セッションログのボリュームの制御	775
ログとイベントのフィルタリング	778
デバッグログの生成	781
CloudWatch メトリクスによるモニタリング	786
AWS Elemental MediaTailor CloudWatch メトリクス	787
AWS Elemental MediaTailor CloudWatch デイメンション	803
メトリクスを使用して古いマニフェストを診断する	803
API コールの記録	806
AWS Elemental MediaTailor CloudTrail の情報	806
AWS Elemental MediaTailor ログファイルエントリについて	807
チャンネルアセンブリアラートの受信	809
マニフェスト関連のアラートのトラブルシューティング	815
アラートの表示	816
アラートの処理	817
リソースのタグ付け	817
サポートされているリソース	817
タグの制限	818
タグの管理	818
ワークフローモニター	818
ワークフローモニターのコンポーネント	820
サポートされるサービス	820
ワークフローモニターの設定	821
ワークフローモニターの使用	840
デュアルスタックのエンドポイント	843
デュアルスタックエンドポイントとは	843
サポートされている機能	843
デュアルスタックのエンドポイント形式	843
デュアルスタックのエンドポイントの使用	844
AWS マネジメントコンソール	844
AWS Command Line Interface	844
AWS SDK	844
考慮事項	845
クォータ	846
広告挿入のクォータ	846
チャンネルアセンブリのクォータ	851
トラブルシューティング	856

イベントフローのトラブルシューティング	856
不完全なフロー	856
タイミング分析	858
よくある問題	860
モニタリングのベストプラクティス	863
その他のリソース	865
再生のトラブルシューティング	865
クライアントエラー	866
サーバーエラー	868
例	870
ビーコン重複排除	871
パラメータのトラブルシューティング	872
広告スキップのトラブルシューティング	872
症状	872
スキップ理由	872
NEW_CREATIVE スキップ	873
ADS タイムアウトスキップ	875
VAST 解析のスキップ	876
期間不一致のスキップ	877
セッション変数のスキップ	878
広告スキップのモニタリング	879
予防のベストプラクティス	880
参考資料	881
MediaTailor のリソース	885
ドキュメント履歴	887
.....	cmviii

とは AWS Elemental MediaTailor

AWS Elemental MediaTailor は、AWS クラウドで実行されるスケーラブルな広告挿入およびチャネルアセンブリサービスです。MediaTailor を使用することで、オーバーザトップ (OTT) 動画アプリケーションでのブロードキャスト品質を維持しながら、的を絞った広告コンテンツを視聴者に提供し、リニアストリームを作成することができます。MediaTailor 広告挿入は、ビデオオンデマンド (VOD) ワークフローとライブワークフローに対して Apple HTTP Live Streaming (HLS) と MPEG Dynamic Adaptive Streaming over HTTP (DASH) をサポートします。

AWS Elemental MediaTailor 広告挿入は、従来の広告追跡システムよりも重要な進歩を提供します。広告は収益化が進み、動画の品質と解像度の一貫性が向上し、マルチプラットフォーム環境での管理が容易になります。MediaTailor は、すべての IP 接続デバイスが、他のコンテンツをレンダリングする場合と同じ方法を使用して広告をレンダリングできるようにすることで、広告ワークフローを簡素化します。このサービスは、広告視聴数の高度な追跡機能も提供し、これによってコンテンツの収益化がさらに強化されます。

AWS Elemental MediaTailor チャネルアセンブリは、既存のビデオオンデマンド (VOD) コンテンツを使用してリニアストリーミングチャネルを作成できるマニフェスト専用サービスです。オリジンサーバーから直接提供されるコンテンツセグメントに MediaTailor が関与することは一切ありません。その代わりに、MediaTailor はオリジンからマニフェストを取得し、それらを使用して、基盤となるコンテンツセグメントを参照するライブスライディングマニフェストウィンドウをアセンブルします。

MediaTailor のチャネルアセンブリは、ストリームを SCTE-35 マーカーで調節することなく、広告ブレイクをストリームに挿入することによって、チャネルの収益化を容易にします。チャネルアセンブリは、MediaTailor 広告挿入、または別のサーバー側広告挿入サービスで使用できます。

オリジンサーバーの要件

AWS Elemental MediaTailor には、オリジンサーバー通信に関する特定の要件があります。

- サポートされているポート - MediaTailor は、標準の HTTP および HTTPS ポートを使用するオリジンのみを受け入れます。
 - HTTP 接続のポート 80
 - HTTPS 接続用のポート 443

MediaTailor は、オリジンサーバー通信のカスタムポートをサポートしていません。

- プロトコル要件 - 安全な通信のために、MediaTailor では特定のオリジンタイプと認証シナリオに HTTPS が必要です。詳細については、「[the section called “コンテンツソースの統合”](#)」を参照してください。

MediaTailor の概念

AWS Elemental MediaTailor ユーザーガイド全体で使用されている概念の概要を次に示します。

HLS プレイリストタイプ

HTTP ライブストリーミング (HLS) は、多変量プレイリストとメディアプレイリストの 2 つの主要なタイプのプレイリストを使用します。MediaTailor ワークフローを設定およびトラブルシューティングするには、これらのプレイリストタイプの違いを理解することが不可欠です。

マルチバリエーションプレイリスト

多変量プレイリストは、HLS ストリーミングワークフローの最上位インデックスファイルであり、コンテンツの使用可能なすべてのレンディションを一覧表示します。メディアプレイリストへの参照が含まれていますが、メディアセグメント自体は含まれていません。多変量プレイリストを使用すると、プレイヤーはネットワーク条件、デバイス機能、またはユーザー設定に基づいて最適なレンディションを選択できます。

このプレイリストタイプは、さまざまなコンテキストの他のいくつかの名前でも知られています。

- マスタープレイリスト (非推奨用語)
- マスターマニフェスト (一般的な業界用語)
- プライマリプレイリスト
- メインプレイリスト
- インデックスファイル
- マスター M3U8

MediaTailor ワークフローでは、多変量プレイリストは再生リクエストのエントリーポイントであり、広告パーソナライゼーションが開始される場所です。

Example 多変量プレイリストの例

```
#EXTM3U
#EXT-X-VERSION:3
```

```
#EXT-X-INDEPENDENT-SEGMENTS
#EXT-X-STREAM-INF:BANDWIDTH=2665726,AVERAGE-
BANDWIDTH=2526299,RESOLUTION=960x540,CODECS="avc1.640029,mp4a.40.2",SUBTITLES="subtitles"
index_1.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=4786126,AVERAGE-
BANDWIDTH=4547599,RESOLUTION=1280x720,CODECS="avc1.640029,mp4a.40.2",SUBTITLES="subtitles"
index_2.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=8171787,AVERAGE-
BANDWIDTH=7768157,RESOLUTION=1920x1080,CODECS="avc1.640029,mp4a.40.2",SUBTITLES="subtitles"
index_3.m3u8
```

メディアプレイリスト

メディアプレイリストには、コンテンツの特定のレンディション (品質レベル) の実際のメディアセグメント情報が含まれます。これには、1 つのレンディションの再生に必要なタイミング情報、セグメント URLs、およびその他のメタデータが含まれます。各メディアプレイリストは、解像度、ビットレート、その他の特性の異なる組み合わせを表します。

このプレイリストタイプは、他のいくつかの名前でも知られています。

- バリエーションプレイリスト
- 子マニフェスト (一般的な業界用語)
- チャンクリスト
- メディア M3U8
- レンディションプレイリスト

MediaTailor ワークフローでは、メディアプレイリストは、コンテンツセグメントと広告セグメントの両方を適切なシーケンスに含めるようにパーソナライズされます。

Exampleメディアプレイリストの例

```
#EXTM3U
#EXT-X-VERSION:6
#EXT-X-TARGETDURATION:6
#EXT-X-MEDIA-SEQUENCE:123456
#EXT-X-DISCONTINUITY-SEQUENCE:0
#EXTINF:6.0,
segment123456.ts
#EXTINF:6.0,
segment123457.ts
#EXT-X-DISCONTINUITY
#EXTINF:5.0,
```

```
ad-segment1.ts
#EXTINF:5.0,
ad-segment2.ts
#EXT-X-DISCONTINUITY
#EXTINF:6.0,
segment123458.ts
```

Note

多変量プレイリストとメディアプレイリストはどちらも、.m3u8拡張機能を使用するテキストベースのファイルです。これらのファイルの用語はドキュメントやコンテキストによって異なりますが、HLS ストリーミングアーキテクチャでは基本的なルールは変わりません。

MediaTailor を設定するときは、通常、コンテンツオリジンのマルチバリエーションプレイリストへの URL を指定します。MediaTailor は、マルチバリエーションプレイリストとメディアプレイリストの両方のパーソナライゼーションを処理し、設定に従って広告を挿入します。

HLS プレイリストの仕様の詳細については、[HTTP Live Streaming RFC](#) を参照してください。

DASH マニフェストタイプ

Dynamic Adaptive Streaming over HTTP (DASH) は、メディアプレゼンテーション説明 (MPD) マニフェストを使用してストリーミングコンテンツを配信します。MediaTailor ワークフローの設定とトラブルシューティングには、DASH マニフェストの構造とコンポーネントを理解することが不可欠です。

MPD (メディアプレゼンテーションの説明)

MPD は、メディアコンテンツの構造と可用性を説明する DASH ストリーミングのプライマリマニフェストファイルです。これには、ストリーミングコンテンツを構成する期間、適応セット、表現、セグメントに関する情報が含まれています。

このマニフェストタイプは、さまざまなコンテキストの他のいくつかの名前でも知られています。

- DASH マニフェスト
- DASH MPD
- マスターマニフェスト (HLS と比較した場合)

- プレゼンテーションマニフェスト

MediaTailor ワークフローでは、MPD は再生リクエストのエントリポイントであり、広告パーソナライゼーションが開始される場所です。

Example MPD マニフェストの例

```
<?xml version="1.0" encoding="UTF-8"?>
<MPD xmlns="urn:mpeg:dash:schema:mpd:2011" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011
DASH-MPD.xsd" profiles="urn:mpeg:dash:profile:isoff-live:2011"
type="dynamic" minBufferTime="PT5.000S" maxSegmentDuration="PT2.005S"
availabilityStartTime="2020-01-01T00:00:00Z" publishTime="2020-01-01T12:30:00Z"
minimumUpdatePeriod="PT2.000S" timeShiftBufferDepth="PT5M">
  <Period id="1" start="PT0.000S">
    <AdaptationSet id="1" contentType="video" segmentAlignment="true"
bitstreamSwitching="true" frameRate="30000/1001" maxWidth="1920" maxHeight="1080"
par="16:9">
      <Representation id="1" mimeType="video/mp4" codecs="avc1.640028" width="1920"
height="1080" bandwidth="5000000">
        <SegmentTemplate timescale="90000" initialization="init-stream
$RepresentationID$.m4s" media="chunk-stream$RepresentationID$-$Number%05d$.m4s"
startNumber="1" duration="180000"/>
      </Representation>
      <Representation id="2" mimeType="video/mp4" codecs="avc1.4d401f" width="1280"
height="720" bandwidth="2800000">
        <SegmentTemplate timescale="90000" initialization="init-stream
$RepresentationID$.m4s" media="chunk-stream$RepresentationID$-$Number%05d$.m4s"
startNumber="1" duration="180000"/>
      </Representation>
      <Representation id="3" mimeType="video/mp4" codecs="avc1.4d401e" width="640"
height="360" bandwidth="1100000">
        <SegmentTemplate timescale="90000" initialization="init-stream
$RepresentationID$.m4s" media="chunk-stream$RepresentationID$-$Number%05d$.m4s"
startNumber="1" duration="180000"/>
      </Representation>
    </AdaptationSet>
    <AdaptationSet id="2" contentType="audio" segmentAlignment="true" lang="eng">
      <Representation id="4" mimeType="audio/mp4" codecs="mp4a.40.2"
audioSamplingRate="48000" bandwidth="128000">
        <SegmentTemplate timescale="48000" initialization="init-stream
$RepresentationID$.m4s" media="chunk-stream$RepresentationID$-$Number%05d$.m4s"
startNumber="1" duration="96000"/>
      </Representation>
    </AdaptationSet>
  </Period>
</MPD>
```

```
    </AdaptationSet>
  </Period>
</MPD>
```

Period

Period は、DASH プレゼンテーションの一時的なセクションです。各期間には 1 つ以上の適応セットが含まれ、メディア時間の範囲を表します。広告挿入ワークフローでは、通常、個別の期間を使用してコンテンツと広告を区別します。

このコンポーネントは、他のいくつかの名前でも知られています。

- コンテンツセグメント
- タイムラインセクション
- プログラムセグメント

MediaTailor ワークフローでは、期間を使用してメインコンテンツを広告コンテンツから分離し、各広告は通常独自の期間で表されます。

Example 期間の例

```
<Period id="ad-period-1" start="PT30.000S" duration="PT15.000S">
  <AdaptationSet id="1" contentType="video" segmentAlignment="true"
    bitstreamSwitching="true" frameRate="30000/1001" maxWidth="1920" maxHeight="1080"
    par="16:9">
    <Representation id="1" mimeType="video/mp4" codecs="avc1.640028" width="1920"
      height="1080" bandwidth="5000000">
      <SegmentTemplate timescale="90000" initialization="ad1/init-stream
$RepresentationID$.m4s" media="ad1/chunk-stream$RepresentationID$-$Number%05d$.m4s"
      startNumber="1" duration="180000"/>
    </Representation>
  </AdaptationSet>
</Period>
```

AdaptationSet

AdaptationSet は、1 つまたは複数のメディアコンテンツコンポーネントの交換可能なエンコードされたバージョンのセットをグループ化します。たとえば、ある AdaptationSet には複数のビデオ品質レベルが含まれ、別の AdaptationSet には複数のオーディオ言語オプションが含まれる場合があります。

このコンポーネントは、以下とも呼ばれます。

- メディアコンポーネントグループ
- ストリームセット
- トラックグループ

MediaTailor ワークフローでは、コンテンツと広告の間で一貫したメディアタイプを維持するために、広告挿入中に AdaptationSets が保持されます。

表記

表現は、AdaptationSet 内のメディアコンテンツの特定のエンコードされたバージョンです。各表現は、通常、ビットレート、解像度、またはその他のエンコーディングパラメータが異なるため、クライアントはネットワーク条件とデバイス機能に基づいて最適なバージョンを選択できます。

このコンポーネントは、以下とも呼ばれます。

- レンディション (HLS に類似)
- 品質レベル
- ビットレートバリエーション
- ストリームバリエーション

MediaTailor ワークフローでは、広告期間の表現は、コンテンツ期間の表現と可能な限り一致させて、スムーズな視聴体験を実現します。

セグメント

セグメントは、URL によって個別に参照できるメディアデータの単位です。セグメントには実際のメディアコンテンツ (ビデオ、オーディオなど) が含まれ、MPD 内で参照されます。DASH には主に 2 つのタイプのセグメントがあります。

- 初期化セグメント: コーデックパラメータやタイミング情報など、表現の初期化情報が含まれます。
- メディアセグメント: 表現内の特定の時間範囲の実際のメディアデータが含まれます。

MediaTailor ワークフローでは、セグメント URLs は、多くの場合、適切なコンテンツまたは広告メディアファイルを指すように変更されます。

Note

DASH マニフェストは、通常 .mpd 拡張機能を使用する XML ベースのファイルです。これらのファイルとそのコンポーネントの用語はドキュメントやコンテキストによって異なる場合がありますが、DASH ストリーミングアーキテクチャでは基本的な構造は変わりません。

MediaTailor を設定するときは、コンテンツオリジンの MPD マニフェストへの URL を指定します。次に、MediaTailor はマニフェストのパーソナライゼーションを処理します。通常は、設定に従って広告の追加の期間を挿入します。

DASH マニフェストの仕様の詳細については、[「DASH Industry Forum Implementation Guidelines」](#)を参照してください。

高度な DASH マニフェストの概念

MediaTailor で DASH マニフェストを使用する場合、次の高度な概念を理解することで、ストリーミングワークフローをより効果的に設定およびトラブルシューティングできます。

期間開始時刻の計算

DASH マニフェストでは、期間の開始時間は MPD の `availabilityStartTime` 属性と期間の `start` 属性に基づいて計算されます。ライブストリームの場合、開始時刻は可用性の開始時刻を基準にしていますが、VOD コンテンツの場合、通常はプレゼンテーションの開始時刻を基準にしています。

MediaTailor が広告期間を挿入すると、開始時間を慎重に計算して、コンテンツと広告間のシームレスな移行を確保します。この計算では、以下が考慮されます。

- 元の期間の開始時刻
- 前述の広告期間の期間
- 広告決定サーバーのレスポンスで指定された任意のタイムオフセット

プリロールタイミングの計算

DASH マニフェストのプリロール広告は、メインコンテンツが開始される前に表示されるため、特別な処理が必要です。MediaTailor は、マニフェストの開始時にプリロール広告を個別の期間として適切な開始時間と期間とともに挿入します。

プリロール広告の場合、MediaTailor:

- プレロール広告ごとに新しい期間を作成します
- 最初のプリロール期間の開始時間を 0 に設定します
- すべてのプリロール広告の合計期間を考慮して、メインコンテンツ期間の開始時間を調整します。

Live-to-VOD 移行

DASH マニフェストは、ライブストリームが終了すると、ライブ形式から VOD (ビデオオンデマンド) 形式に移行できます。この移行では、MPD の `type` 属性を `dynamic` に変更し、`timeShiftBufferDepth` や その他の他の属性を調整します `minimumUpdatePeriod`。

MediaTailor が live-to-VOD 移行中にマニフェストを処理すると、以下が保証されます。

- 広告マーカースは VOD マニフェストに保持されます
- 期間の開始時間は、適切なタイミングを維持するために調整されます。
- マニフェストは VOD 再生クライアントと互換性があります

DRM 処理

DASH マニフェストのデジタル著作権管理 (DRM) 情報は、通常、`AdaptationSets` または表現内の `ContentProtection` 要素に含まれます。MediaTailor は広告挿入中にこれらの要素を保持し、コンテンツ保護がそのまま維持されるようにします。

DRM で保護されたコンテンツを処理する場合、MediaTailor は次のようになります。

- 元のマニフェストのすべての `ContentProtection` 要素を維持します
- 該当する場合、広告コンテンツが互換性のある DRM スキームを使用していることを確認します
- マニフェスト全体で DRM 関連の属性と要素を保持します

SCC フラグ

DASH マニフェストの補足コンテンツコントロール (SCC) フラグは、コンテンツの特性と再生要件に関する追加情報を提供します。これらのフラグは通常、MPD 構造内の属性または要素として含まれます。

MediaTailor が処理する一般的な SCC フラグには以下が含まれます。

- コンテンツ評価情報
- アクセシビリティ機能 (クローズドキャプション、オーディオの説明)
- コンテンツアドバイザリ通知

- 再生の制限

MediaTailor は、マニフェスト処理中にこれらのフラグを保持し、すべてのコンテンツメタデータがパーソナライズされたマニフェストに維持されるようにします。

これらの高度な概念を理解することで、最適なパフォーマンスを実現するように MediaTailor を設定し、DASH ストリーミングワークフローで発生する可能性のある問題をトラブルシューティングできます。

広告挿入の概念

以下は、広告挿入に関連する概念の概要です。

広告表示

広告主に販売できる広告時間枠内の特定の広告時間単位。広告ブレイクには、複数の広告表示を含めることができます。MediaTailor は、広告決定サーバーから VAST レスポンスを受信すると、パーソナライズされた広告でこれらの表示を埋めます。

広告時間枠

コマーシャルが表示されるプログラミング中の期間。広告ブレイクは、コンテンツ前 (ロール前)、コンテンツ中 (ロール中)、またはコンテンツ後 (ロール後) に発生する可能性があります。MediaTailor は、SCTE-35 などのマーカーを通じてマニフェストの広告ブレイクを識別します。

広告決定サーバー (ADS)

現在の広告キャンペーンと視聴者の好みなどの基準に基づいて広告スポット仕様を提供するサーバー。

設定

ユーザーが操作する MediaTailor 内のオブジェクト。設定は、オリジンサーバーと広告決定サーバー (ADS) に関する位置情報を保持します。また、MediaTailor 内外のアクセスポイントを提供するエンドポイントも保持します。

ダイナミックトランスコーディング

コンテンツがリクエストされる時に、広告の品質と形式をプライマリ動画コンテンツに一致させるプロセス。ダイナミックトランスコーディングは、ストレージの要件を削減し、再生が広告と動画コンテンツ間をシームレスに移行することを確実にします。

マニフェスト操作

マニフェストが適切な広告フラグメントとコンテンツフラグメントを参照するように、オリジンサーバーからのマニフェストを書き換えるプロセス。広告は、広告決定サーバー (ADS) からの VAST レスポンスによって決定されます。MediaTailor は、再生の進行に沿って、コンテンツストリームへの広告挿入または広告置換を実行します。

VAST と VMAP

動画広告配信テンプレート (VAST) と動画複数広告再生リスト (VMAP) は、広告決定サーバー (ADS) が MediaTailor からの広告リクエストに対して送信する XML レスポンスです。このレスポンスは、MediaTailor がマニフェストに挿入する広告を決定します。VMAP には、広告時間枠のタイミングと広告表示も含まれます。MediaTailor 広告挿入の背景にあるロジックの詳細については、「[AWS Elemental MediaTailor 広告挿入動作について](#)」を参照してください。MediaTailor が VAST と連動する方法の詳細については、「[the section called “広告サーバー統合要件”](#)」を参照してください。

チャンネルアセンブリの概念

以下は、チャンネルアセンブリに関連する概念の概要です。

チャンネル

チャンネルは、ソースマニフェストをリニアストリームにアセンブルします。各チャンネルには、プレイヤーがアクセスする再生 URL が含まれた 1 つ、または複数の出力があります。チャンネル出力は、ユーザーが VOD ソース用に作成するパッケージ設定の内容に対応します。チャンネルには、チャンネルのストリームで VOD ソースがいつ再生されるかを決定するスケジュールが含まれています。

パッケージ設定

パッケージ設定は VOD ソースを表すもので、特定のパッケージ化された形式の特性が含まれています。パッケージ設定をチャンネル出力に関連付けて、HTTP Live Streaming (HLS) などの VOD ソースのパッケージ形式の再生ストリームを作成します。

スケジュール

各チャンネルは、チャンネルのスケジュールに沿って配置されたプログラムで構成されています。スケジュールは、チャンネルのリニアストリームでプログラムがいつ再生されるかを決定します。

ソースロケーション

ソースロケーションは、アセットが保存されているオリジンサーバーを表します。Amazon S3、HTTP サーバー、コンテンツ配信ネットワーク (CDN)、MediaPackage などのパッケージングインフラストラクチャを使用できます。

VOD ソース

VOD ソースは、映画やテレビ番組のエピソードといった単一のコンテンツを表します。VOD ソースをプログラムに関連付けて、それらをチャンネルのリニアストリームに追加します。

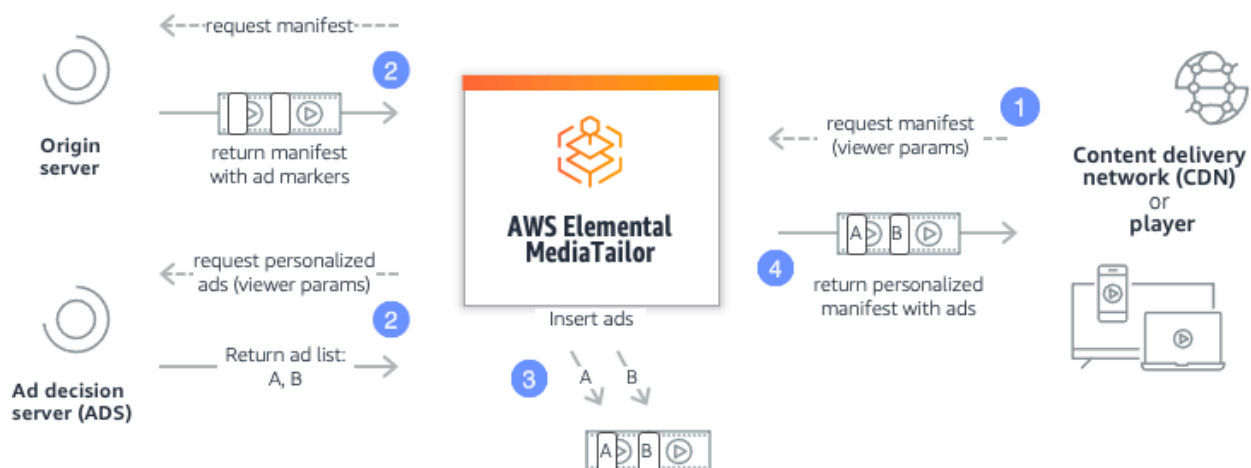
対象者

オーディエンスは、オプションで代替コンテンツを持つことができるビューワーコホートを定義します。標準リニアチャンネルでオーディエンスを定義できます。

MediaTailor 広告挿入の仕組み

AWS Elemental MediaTailor は、コンテンツ配信ネットワーク (CDN)、オリジンサーバー、広告決定サーバー (ADS) とやり取りして、パーソナライズされた広告をライブコンテンツとビデオオンデマンドコンテンツ内の広告ブレイクに結び付けます。

以下は、MediaTailor 広告挿入の仕組みの概要です。



1. プレイヤー、または Amazon CloudFront などの CDN が、HLS または DASH コンテンツに対するリクエストを MediaTailor に送信します。リクエストには、視聴者に関する情報 (パーソナライゼーションに使用されるもの) を伴うプレイヤーからのパラメータが含まれています。
2. MediaTailor が、視聴者情報が含まれる ADS にリクエストを送信します。ADS が、視聴者情報と現行の広告キャンペーンに基づいて広告を選択し、MediaTailor への VAST または VMAP レスポンスで、広告クリエイティブへの URLs を返します。

広告を事前調整した場合、URLs は事前にトランスコードされた広告を指します。事前にトランスコードされた広告による広告ステッチングの詳細については、「」を参照してください [前提条件付き広告](#)。

3. MediaTailor はマニフェストを操作して、ADS から返 URLs を含め、オリジンコンテンツのエンコード特性に合わせてトランスコードします。前提条件付き広告を使用している場合は、その広告がテンプレートマニフェストと一致することを確認する責任があります。

広告がコンテンツに一致するようにトランスコードされていない場合、MediaTailor はその広告の挿入をスキップし、次のリクエストに対応できるように MediaConvert を使用して広告の準備を整えます。

4. MediaTailor が、完全にパーソナライズされたマニフェストをリクエスト元の CDN またはプレイヤーに返します。

ADS は、広告の開始時点、広告の中間時点、広告の終了時点などの視聴マイルストーンに基づいて、閲覧された広告を追跡します。再生の進行に伴って、プレイヤーまたは MediaTailor が広告追跡ビーコンを ADS の広告追跡 URL に送信し、広告がどのくらい閲覧されたのかを記録します。プレイヤーと MediaTailor のどちらがセッションに関するこれらのビーコンを送信するかは、MediaTailor とのセッション開始時にプレイヤーが指定します。

広告挿入の開始方法については、「[MediaTailor の使用開始](#)」を参照してください。

AWS Elemental MediaTailor 広告挿入イベントフロー

AWS Elemental MediaTailor は、予測可能な一連のイベントを通じて、サーバー側の広告挿入のマニフェストパーソナライゼーションを処理します。このイベントフローを理解することで、MediaTailor が広告機会をパーソナライズされた視聴体験に変換する方法を理解し、問題が発生した場合のトラブルシューティングの基盤を提供できます。

各広告挿入の機会は、MediaTailor が可視性とモニタリングのためにログに記録するイベントの時系列シーケンスに従います。これらのイベントは、広告機会の検出から追跡情報の配信まで、マニフェストのパーソナライゼーションプロセスにおける重要なマイルストーンを表します。

標準イベントシーケンス

MediaTailor が広告挿入オポチュニティを正常に処理すると、通常、次の一連のイベントが発生します。

1. 広告機会の検出 - MediaTailor は、コンテンツマニフェストで広告マーカ (SCTE-35 など) を検出し、広告パーソナライゼーションの機会が利用可能であることを示します。
2. 広告決定サーバーリクエスト - MediaTailor は、設定された広告決定サーバー (ADS) にリクエストを送信して、ビューワーパラメータや広告時間枠期間など、検出されたオポチュニティの広告を取得します。
3. 広告レスポンス処理 - MediaTailor は、広告クリエイティブ情報、追跡 URLs、および含まれる広告に関するメタデータを含むレスポンスを ADS から受け取ります。VAST、VMAP、VPAID レスポンス形式の詳細については、「」を参照してください [広告サーバー統合要件](#)。
4. マニフェストパーソナライゼーション - MediaTailor は広告レスポンスを正常に処理し、必要に応じて広告をトランスコードし、広告セグメントとオリジンコンテンツの両方 URLs を含むパーソナライズされたマニフェストを生成します。次に、MediaTailor はパーソナライズされたマニフェストを再生デバイスに送信します。
5. ビーコンのアクティベーションの追跡 - 追跡ビーコンは、広告イベント (インプレッション、四分位数、完了など) を広告サーバーやその他の測定システムにレポートするために起動されます。サーバー側のレポートモード (デフォルト) では、MediaTailor はプレーヤーセグメントリクエストに基づいてビーコンを起動します。クライアント側のレポートモードでは、再生デバイスは、パーソナライズされたマニフェストで提供された URLs を使用してビーコンを起動します。

一般的なタイミング: このシーケンスは、広告決定サーバーの応答時間と広告クリエイティブ処理要件に応じて、通常 2~5 秒以内に完了します。

一般的なイベントバリエーション

すべての広告挿入試行が標準の成功シーケンスに従っているわけではありません。MediaTailor は、イベントフローを変更する可能性のあるさまざまなシナリオを処理します。

空の広告レスポンスシナリオ

広告決定サーバーが広告を返さない場合:

1. 広告機会の検出は通常どおり行われます
2. 広告決定サーバーリクエストが正常に送信されました
3. ADS は広告なしで空のレスポンスを返します

4. このオポチュニティのパーソナライズされたマニフェストには広告が含まれません

これは、広告インベントリの可用性、ターゲット基準、または広告決定サーバーで設定されたビジネスルールが原因で発生する可能性がある通常のシナリオです。この場合、ストリームがライブまたは live-to-VOD ストリームであり、スレートが設定されていない限り、基盤となるコンテンツが表示されます。詳細については、「[MediaTailor スレート広告挿入](#)」を参照してください。

エラーシナリオ

技術的な問題により広告挿入が成功しない場合、イベントフローはさまざまな時点で中断される可能性があります。

- 広告決定サーバーのタイムアウト - ADS へのリクエストが設定されたタイムアウトしきい値を超えているため、広告の取得ができません。
- 通信エラー - ネットワークまたは接続の問題により、MediaTailor が広告決定サーバーに到達できなくなります。
- レスpons解析エラー - ADS は、無効な VAST 形式または構造が原因で MediaTailor が解析できないレスポンスを返します。

エラーシナリオでは、MediaTailor は通常、広告なしでコンテンツの再生を継続し、中断のない視聴体験を実現します。スレートが設定されたライブストリームまたは live-to-VOD ストリームの場合、代わりにスレートコンテンツが表示されることがあります。詳細については、「[MediaTailor スレート広告挿入](#)」を参照してください。

VAST リダイレクトシナリオ

最初の VAST レスポンスに別の広告サーバーへのリダイレクトが含まれている場合:

1. 広告機会の検出と最初の ADS リクエストが正常に実行される
2. 初期レスポンスにはリダイレクト命令が含まれています
3. MediaTailor はリダイレクトに従って最終的な広告レスポンスを取得します
4. マニフェストのパーソナライゼーションは最終的な広告コンテンツに進みます
5. ビーコンのアクティベーションの追跡が正常に行われる

VAST リダイレクトはプログラムによる広告で一般的であり、広告サーバーの連鎖とリアルタイムの意思決定を可能にします。MediaTailor は、広告取得プロセス中に最大 7 つのラッパーリダイレクトと無制限の HTTP リダイレクトを許可します。

イベントタイミングの概要

MediaTailor イベントフローのタイミング特性を理解することは、広告挿入パフォーマンスの適切な期待値を設定するのに役立ちます。

- 広告機会検出 - MediaTailor がプレイヤーからGetManifestリクエストを受け取り、コンテンツマニフェストで広告マーカールを検出したときに発生します。
- 広告決定サーバーのインタラクション - 通常、ADS 応答時間とネットワーク条件に応じて 100 ~ 500 ミリ秒かかります。
- マニフェストパーソナライゼーション - 通常、ADS レスポンスを受信してから 50 ~ 200 ミリ秒以内に完了します。
- ビーコンタイミングの追跡 - レポートモードによって異なります。サーバー側のビーコンの場合、タイミングはクライアントプレイヤーからのセグメントリクエストに基づきます。クライアント側のビーコンはプレイヤーの実装に基づいて発生します。

パフォーマンスに関する考慮事項: 最適な視聴エクスペリエンスを維持するために、広告機会の検出からマニフェスト配信までの合計時間は、通常 5 秒未満にとどまる必要があります。上記のように、個々のコンポーネントはミリ秒単位で完了します。ただし、5 秒のしきい値は、完全な広告挿入ワークフロー中に発生する可能性のあるタイムアウト値、再試行回数、ネットワークの変動を考慮します。遅延が長いほど、ADS のパフォーマンスの問題やネットワーク接続の問題を示している可能性があります。

イベントフローの知識の使用

MediaTailor イベントフローを理解することは、以下の基盤となります。

- 実装計画 - イベントシーケンスを理解することは、プレイヤー統合と広告サーバー設定の設計に役立ちます。
- パフォーマンスの最適化 - 期待されるタイミングを理解することで、ボトルネックと最適化の機会を特定できます。
- トラブルシューティングの準備 - 通常のイベントフローに精通していると、問題が発生したときの識別が容易になります。

イベントフロー分析を使用したトラブルシューティングの詳細なガイダンスについては、「」を参照してください [トラブルシューティング](#)。イベントのログ記録とモニタリングに関する技術的な詳細については、「」を参照してください [ログの表示](#)。

関連サービス

- Amazon CloudFront は、データやビデオを視聴者に安全に配信するグローバルコンテンツ配信ネットワーク (CDN) サービスです。CloudFront を使用して、可能な限り最高のパフォーマンスでコンテンツを配信します。CloudFront の詳細については、[Amazon CloudFront ウェブサイト](#)を参照してください。
- AWS Elemental MediaPackage はジャストインタイムのパッケージングおよびオリジネーションサービスであり、リクエストを行うデバイスと互換性のある形式での配信用に、ライブ動画アセットをカスタマイズします。AWS Elemental MediaPackage MediaTailor にストリームを送信する前に、をオリジンサーバーとして使用してコンテンツを準備し、広告マーカを追加します。MediaTailor がオリジンサーバーと連動する方法の詳細については、「[MediaTailor 広告挿入の仕組み](#)」を参照してください。
- AWS Identity and Access Management (IAM) は、ユーザーの AWS リソースへのアクセスを安全に制御するのに役立つウェブサービスです。IAM により、どのユーザーがお客様の AWS リソースを使用できるか (認証)、それらのユーザーがどのリソースをどのような方法で使用できるか (認可) を制御できます。詳細については、「[セットアップ AWS Elemental MediaTailor](#)」を参照してください。

MediaTailor へのアクセス

MediaTailor には、サービスのコンソールを使用してアクセスできます。

サービスを使用するアクセス許可があることを確認する認証情報を指定 AWS アカウント して、にアクセスします。

MediaTailor コンソールにログインするには、次のリンク、**<https://console.aws.amazon.com/mediatailor/home>** を使用してください。

MediaTailor の料金

他の AWS 製品と同様に、MediaTailor を使用するための契約や最低契約金はありません。料金は、サービスの使用量に基づいて請求されます。詳細については、「[MediaTailor の料金](#)」を参照してください。

MediaTailor のリージョン

アプリケーションにおけるデータレイテンシーを低減するため、MediaTailor は、リクエストを行うリージョンエンドポイントを提供します。MediaTailor を利用できるリージョンのリストを確認するには、「[リージョンエンドポイント](#)」を参照してください。

セットアップ AWS Elemental MediaTailor

このセクションでは、ユーザーが AWS Elemental MediaTailor にアクセスするのに必要なステップについて説明していきます。MediaTailor 向けの Identity and Access Management に関する背景と追加情報については、「[の Identity and Access Management AWS Elemental MediaTailor](#)」を参照してください。

の使用を開始するには AWS Elemental MediaTailor、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](#)

にサインアップする AWS アカウント

の使用を開始するには AWS、が必要です AWS アカウント。の作成の詳細については AWS アカウント、「AWS アカウント管理 リファレンスガイド」の「[の開始方法 AWS アカウント](#)」を参照してください。

MediaTailor の使用開始

MediaTailor の使用を開始するには、2 つのチュートリアルから選択できます。1 つは広告挿入の設定、もう 1 つはチャンネルアセンブリです。ライブイベントの実践的なワークショップを試すこともできます。広告挿入チュートリアルでは、パーソナライズされた広告をコンテンツストリームに組み込むために MediaTailor に再生リクエストを送信する方法について説明します。チャンネルアセンブリチュートリアルでは、再生 URL を使用してブラウザで直接、パーソナライズされた広告を含むチャンネルのストリームを表示する方法を示します。

[ライブイベントフレームワークワークショップ](#)も参照できます。この実践的なワークショップでは、最初の検出とデプロイから最適化とスケーリングまで、AWS Elemental MediaTailor ライブイベントのセットアップの完全なライフサイクルについて説明します。ワークショップに従って、すべてのイベントとチャンネルに対して 1 つの MediaTailor 設定を使用する簡単な設定、またはイベントまたはチャンネルごとに専用設定を使用する高度なアーキテクチャを行うことができます。

トピック

- [MediaTailor 広告挿入の使用開始](#)
- [MediaTailor チャンネルアセンブリの使用開始](#)

MediaTailor 広告挿入の使用開始

を使用するには AWS Elemental MediaTailor、MediaTailor 設定にアクセス、表示、編集するためのおよび アクセス AWS アカウント 許可が必要です。これを行う方法については、「[セットアップ AWS Elemental MediaTailor](#)」を参照してください。

この入門チュートリアルでは、次のタスクの実行方法を示します。

- HLS または DASH コンテンツストリームを準備する
- 広告決定サーバー (ADS) テンプレート URL を設定する
- 再生エンドポイントが含まれる MediaTailor 設定を作成する
- プレイヤーまたはコンテンツ配信ネットワーク (CDN) を使用して MediaTailor に再生リクエストを行う

動変数とプレイヤーパラメータを使用した高度な ADS URL 設定については、「」を参照してください [ADS リクエストの MediaTailor 動的広告変数](#)。CDNs 「」を参照してください [MediaTailor マニフェストクエリパラメータ](#)。

これらを完了したら、ストリーム内のパーソナライズされた広告コンテンツに対する再生リクエストを MediaTailor に送信できるようになります。

トピック

- [前提条件](#)
- [ステップ 1: アクセス AWS Elemental MediaTailor](#)
- [ステップ 2: ストリームを準備する](#)
- [ステップ 3: ADS リクエスト URL とクエリパラメータを設定する](#)
- [ステップ 4: 設定を作成する](#)
- [ステップ 5: 設定をテストする](#)
- [ステップ 6: 再生リクエストを送信する AWS Elemental MediaTailor](#)
- [ステップ 7 \(オプション\): AWS Elemental MediaTailor アクティビティをモニタリングする](#)
- [ステップ 8: クリーンアップする](#)

前提条件

開始する前に、[セットアップ AWS Elemental MediaTailor](#) の手順を完了するようにしてください。

ステップ 1: アクセス AWS Elemental MediaTailor

IAM 認証情報を使用して、MediaTailor コンソール (<https://console.aws.amazon.com/mediatailor/home>) にサインインします。

ステップ 2: ストリームを準備する

AWS Elemental MediaTailor と互換性のある HLS または DASH のマニフェストを生成するようにオリジンサーバーを設定します。オリジンサーバーは、標準の HTTP ポート (ポート 80) または HTTPS ポート (ポート 443) を使用する必要があります。MediaTailor は、オリジンサーバー通信のカスタムポートをサポートしていません。

HLS ストリームを準備する

HLS マニフェストは以下の要件を満たす必要があります。

- マニフェストはパブリックインターネットでアクセス可能であることが必要です。
- マニフェストはライブまたはビデオオンデマンド (VOD) である必要があります。
- マニフェストに 3 以降の EXT-X-VERSION があること。

- ライブコンテンツの場合は、マニフェストに ad avail を表すマーカがあること。VOD コンテンツの場合、これはオプションで、代わりに VMAP timeoffset を使用できます。

マニフェストファイルには、以下のいずれかでマークされた広告スロットが必要です。

- #EXT-X-CUE-OUT / #EXT-X-CUE-IN (一般的) 継続時間は以下の例のようになります。

```
#EXT-X-CUE-OUT:60.00  
#EXT-X-CUE-IN
```

- #EXT-X-DATERANGE (一般的ではない) 継続時間は以下の例のようになります。

```
#EXT-X-DATERANGE:ID="",START-DATE="",DURATION=30.000,SCTE35-OUT=0xF  
#EXT-X-DATERANGE:ID="",START-DATE="",DURATION=30.000,SCTE35-OUT=0xF
```

#EXT-X-DATERANGE にあるフィールドはすべて必須です。

マニフェストで広告マーカを設定する方法は、広告がストリームに挿入されるか、ストリーム内の他のフラグメントに置き換えられるかに影響します。詳細については、「[the section called “広告挿入動作について”](#)」を参照してください。

- HLS マルチバリエーションプレイリストは、[HTTP ライブストリーミング: マスタープレイリストタグ](#)に記載されている HLS 仕様に従う必要があります。特に、#EXT-X-STREAM-INF には RESOLUTION、BANDWIDTH、および CODEC の各フィールドを含める必要があります。

ストリームを設定したら、マルチバリエーションプレイリストのコンテンツオリジン URL プレフィックスを書き留めます。このプレフィックスはこのチュートリアルの後半の AWS Elemental MediaTailor で設定を作成するのに必要です。

DASH ストリームを準備する

DASH マニフェストは以下の要件を満たす必要があります。

- マニフェストはパブリックインターネットでアクセス可能であることが必要です。
- マニフェストはライブまたはビデオオンデマンド (VOD) である必要があります。
- マニフェストが、スプライス挿入マーカまたはタイムシグナルマーカを使用して、イベントを ad avail としてマークすること。広告マーカは、平文の XML または base64 エンコードのバイナリで指定できます。スプライス挿入の場合は、ネットワーク外インジケータを有効にする必要があります。タイムシグナルマーカの場合、セグメンテーション UPID 内にあるセグメンテーションタイプ ID は、によって認識されるキューアウト値である必要があります AWS Elemental

MediaTailor。Ad avail はイベントの開始時に開始され、イベント継続時間 (指定されている場合)、または次のイベントが開始されるまで続きます。

以下は、サプライス挿入マーカを使用して ad avail として指定されたイベントの例です。この ad avail の継続時間はイベントの継続時間です。

```
<Period start="PT444806.040S" id="123586" duration="PT15.000S">
  <EventStream timescale="90000" schemeIdUri="urn:scte:scte35:2013:xml">
    <Event duration="1350000">
      <scte35:SpliceInfoSection protocolVersion="0" ptsAdjustment="180832"
tier="4095">
        <scte35:SpliceInsert spliceEventId="4026531855"
spliceEventCancelIndicator="false" outOfNetworkIndicator="true"
spliceImmediateFlag="false" uniqueProgramId="1" availNum="1" availsExpected="1">
          <scte35:Program><scte35:SpliceTime ptsTime="5672624400"/></
scte35:Program>
            <scte35:BreakDuration autoReturn="true" duration="1350000"/>
          </scte35:SpliceInsert>
        </scte35:SpliceInfoSection>
      </Event>
    </EventStream>
    <AdaptationSet mimeType="video/mp4"
      ...
    </AdaptationSet>
  </Period>
```

- 広告表示の AdaptationSet および Representation は、コンテンツストリームの期間と同じ設定である必要があります。AWS Elemental MediaTailor はこれらの設定を使用して、コンテンツストリームに合わせて広告をトランスコードし、コンテンツと広告間でスムーズに切り替わるようにします。

ストリームを設定したら、DASH マニフェストのコンテンツオリジン URL プレフィックスを書き留めておきます。このチュートリアルの後半にある [DASH マニフェストを作成する](#) で設定を作成するには AWS Elemental MediaTailor、これが必要です。

ステップ 3: ADS リクエスト URL とクエリパラメータを設定する

ADS が必要とするクエリパラメータを判断するには、ADS から広告タグ URL を生成します。この URL は ADS に対するリクエストのテンプレートとして機能し、以下の値で構成されています。

- 静的値
- によって生成された値 AWS Elemental MediaTailor (sessionまたは avail クエリパラメータで示される)
- クライアントアプリケーションから取得した、プレイヤーによって生成された値 (player_params. クエリパラメータで表されます)

Example ADS からの広告タグ URL

```
https://my.ads.com/ad?  
output=vast&content_id=12345678&playerSession=[session.id]&cust_params=[player_params.cust_params]
```

上記の中で、

- output と content_id は静的値です。
- playerSession=[session.id] は、 によって提供される動的値です AWS Elemental MediaTailor。 [session.id] の値はプレイヤーセッションごとに変更されるため、VAST リクエストの URL はセッションごとに異なります。
- cust_params はプレイヤー提供の動的値です。

プレイヤーからの多変量プレイリストリクエストは、ADS リクエスト URL のplayer_params. クエリパラメータに対応するキーと値のペアを提供する必要があります。へのリクエストでキーと値のペアを設定する方法の詳細については AWS Elemental MediaTailor、「」を参照してください [ADS リクエストの MediaTailor 動的広告変数](#)。

[ステップ 4: 設定を作成する](#) で、オリジンサーバー/ADS マッピングを MediaTailor で作成するとき、設定した「テンプレート」URL を入力します。

テスト

テスト用に ADS からの静的 VAST レスポンスを使用できます。理想的には、VAST レスポンスは、AWS Elemental MediaTailor がトランスコードできるメザニン品質の MP4 レンディションを返します。ADS からのレスポンスに複数の再生レンディションが含まれている場合、MediaTailor は品質と解像度が最高の MP4 レンディションを選択して、それをトランスコーダーに送信します。

ステップ 4: 設定を作成する

AWS Elemental MediaTailor 設定には、オリジンサーバーと ADS のマッピング情報が保持されます。

設定を作成する (コンソール)

1. MediaTailor コンソール (<https://console.aws.amazon.com/mediatailor/>) を開きます。
2. [設定] ページで、[Create configuration (設定の作成)] を選択します。
3. ページ下部の [設定] セクションで、[設定名] に設定を説明する一意の名前を入力します。名前は設定の主な識別子です。許容される最大長は 512 文字です。
4. 動画コンテンツソースには、HLS 多変量プレイリストの URL プレフィックスまたはこのストリームの DASH マニフェストからアセット ID を引いた値を入力します。たとえば、多変量プレイリスト URL が の場合 `http://origin-server.com/a/master.m3u8`、 と入力します `http://origin-server.com/a/`。別の方法として `http://origin-server.com` のような短いプレフィックスを入力することもできますが、そうする場合は、コンテンツに対するプレイヤーリクエストのアセット ID に `/a/` を含める必要があります。最大長は 512 文字です。

Note

コンテンツオリジンが HTTPS を使用している場合、その証明書は既知の認証機関からのものであることが必要です。(この目的に自己署名証明書を使用することはできません)。それ以外の場合、はコンテンツオリジンへの接続に AWS Elemental MediaTailor 失敗し、プレイヤーのリクエストに応じてマニフェストを提供できません。

5. [Ad decision server] (広告決定サーバー) には、ADS の URL を入力します。これは、「[ステップ 3: ADS リクエスト URL とクエリパラメータを設定する](#)」で説明されている変数を含む URL、またはテスト目的で使用している静的 VAST URL です。最大長は 25,000 文字です。

Note

ADS が HTTPS を使用している場合、その証明書は既知の認証機関からのものであることが必要です。(この目的に自己署名証明書を使用することはできません)。これは、ADS によって返されるメザニン広告 URL についても同様です。そうではない場合、MediaTailor はコンテンツオリジンからの広告の取得とマニフェストへのステッチに失敗します。

6. (必要に応じて DASH に使用するオプション) MediaTailor マニフェストへのアクセスに CDN ルーティングルールをセットアップしており、クライアント側のレポートを使用している、またはプレイヤーがスティッキー HTTP リダイレクトをサポートする場合は、[Location] (ロケーション) に [DISABLED] (無効) を選択します。

[Location] (ロケーション) 機能の詳細については、「[the section called “ロケーション機能”](#)」を参照してください。

7. (オプション) オリジンサーバーがシングルピリオド DASH マニフェストを生成する場合は、[DASH mpd manifest origin type] (DASH mpd マニフェストオリジンタイプ)、[SINGLE_PERIOD] の順に選択します。デフォルトで、MediaTailor は DASH マニフェストをマルチピリオドマニフェストとして処理します。詳細については、「[the section called “MPEG-DASH ソースの統合”](#)」を参照してください。
8. [Create configuration] (設定を作成) をクリックします。

AWS Elemental MediaTailor は、設定ページに新しい設定を表示します。

ステップ 5: 設定をテストする

設定を保存したら、ストリーミングプロトコルに該当する形式の URL を使用してストリームをテストします。

- 例: HLS

```
playback-endpoint/v1/master/hashed-account-id/origin-id/master.m3u8
```

- 例: DASH

```
playback-endpoint/v1/dash/hashed-account-id/origin-id/manifest.mpd
```

説明は以下のとおりです。

- playback-endpoint は、設定を作成したときに AWS Elemental MediaTailor によって生成された固有の再生エンドポイントです。

例

```
https://777788889999.mediatailor.us-east-1.amazonaws.com
```

- hashed-account-id は AWS アカウント ID です。

例

```
777788889999
```

- origin-id は、設定を作成したときに付けた名前です。

例

```
myOrigin
```

- master.m3u8 または manifest.mpd は、テストストリームからのマニフェストの名前とそのファイル拡張子です。これは、[the section called “ステップ 4: 設定を作成する”](#) で設定した動画コンテンツソースにマニフェストを付加するときに、マニフェストが完全に識別されるように定義します。

上記の例の値を使用すると、完全な URL は以下ようになります。

- 例: HLS

```
https://777788889999.mediatailor.us-east-1.amazonaws.com/v1/master/  
AKIAIOSFODNN7EXAMPLE/myOrigin/master.m3u8
```

- 例: DASH

```
https://777788889999.mediatailor.us-east-1.amazonaws.com/v1/dash/  
AKIAIOSFODNN7EXAMPLE/myOrigin/manifest.mpd
```

以下のいずれかの方法でストリームをテストできます。

- 前の例に示したように、スタンドアロンプレイヤーに URL を入力します。
- ユーザー独自のプレイヤー環境でストリームをテストします。

ステップ 6: 再生リクエストを に送信する AWS Elemental MediaTailor

AWS Elemental MediaTailorから渡された設定の再生エンドポイントに再生リクエストを送信するように、ダウンストリームプレイヤーまたは CDN を設定します。[ステップ 3: ADS リクエスト URL とクエリパラメータを設定する](#) で ADS リクエスト URL に使用したプレイヤー定義の動的変数は、いずれもプレイヤーからのマニフェストリクエスト内で定義される必要があります。

Example

以下がテンプレートの ADS URL であるとしてします。

```
https://my.ads.com/ad?  
output=vast&content_id=12345678&playerSession=[session.id]&cust_params=[player_params.cust_params]
```

次に、ADS ではなくオリジンサーバーに先行しない ads.. AWS Elemental MediaTailor passes パラメータをキーと値のペアの前に付けads.で、プレイヤーリクエスト [player_params.cust_params]で を定義します。

プレイヤーリクエスト URL は以下の HLS と DASH 例のバリエーションになります。

```
https://777788889999.mediatailor.us-east-1.amazonaws.com/v1/master/  
AKIAIOSFODNN7EXAMPLE/myOrigin/master.m3u8?ads.cust_params=viewerinfo
```

```
https://777788889999.mediatailor.us-east-1.amazonaws.com/v1/dash/AKIAIOSFODNN7EXAMPLE/  
myOrigin/manifest.mpd?ads.cust_params=viewerinfo
```

がプレイヤーリクエスト AWS Elemental MediaTailor を受信すると、リクエスト内の情報に基づいてプレイヤー変数が定義されます。結果として得られる ADS リクエスト URL は、このバリエーションになります。

```
https://my.ads.com/ad?  
output=vast&content_id=12345678&playerSession=<filled_in_session_id>&cust_params=viewerinfo
```

ADS に渡すキーバリューペアの設定に関する詳細については、「[ADS リクエストの MediaTailor 動的広告変数](#)」を参照してください。

ステップ 7 (オプション): AWS Elemental MediaTailor アクティビティをモニタリングする

Amazon CloudWatch と Amazon CloudWatch Logs を使用して、リクエスト数、エラー数、入力された広告表示数などの AWS Elemental MediaTailor アクティビティを追跡します。

で CloudWatch を初めて使用する場合は AWS Elemental MediaTailor、サービス間の通信を許可する AWS Identity and Access Management (IAM) ロールを作成します。

CloudWatch AWS Elemental MediaTailor へのアクセスを許可するには (コンソール)

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、[Roles] (ロール)、[Create role] (ロールを作成) の順に選択します。
3. 別の AWS アカウントロールタイプを選択します。
4. アカウント ID に ID を入力します AWS アカウント。
5. [Require external ID (外部 ID を必須にする)] を選択し、「**midas**」と入力します。このオプションにより、リクエストに正しい `sts:ExternalID` が含まれている場合にのみサービスがロールを引き受けることができるという条件が、信頼ポリシーに自動的に追加されます。
6. [Next: Permissions] (次のステップ: 許可) を選択します。
7. このロールが実行できるアクションを指定するアクセス許可ポリシーを追加します。以下のいずれかのオプションを選択した後、[Next: Review (次へ: 確認)] を選択します。
 - Amazon CloudWatch Logs への完全なアクセス権を付与するための `CloudWatchLogsFullAccess`
 - Amazon CloudWatch への完全なアクセス権を付与するための `CloudWatchFullAccess`
8. [ロール名] に「**MediaTailorLogger**」と入力し、[ロールの作成] を選択します。
9. [ロール] ページで、作成したロールを選択します。
10. 信頼関係を編集してプリンシパルを更新します。
 1. ロールの [Summary (概要)] ページで、[Trust relationship (信頼関係)] タブを選択します。
 2. [Edit trust relationship (信頼関係の編集)] を選択します。
 3. ポリシードキュメントで、プリンシパルを AWS Elemental MediaTailor サービスに変更します。以下のようになります。

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

ポリシー全体は以下のようになります。

4. 信頼ポリシーの更新 を選択します。

ステップ 8: クリーンアップする

余分な料金が発生しないように、不要な設定をすべて削除します。

設定を削除するには (コンソール)

1. AWS Elemental MediaTailor 設定ページで、次のいずれかを実行します。
 - 削除する設定の [設定名] を選択します。
 - [設定名] 列でラジオボタンを選択してから、[Delete (削除)] を選択します。
2. [設定の削除] 確認ボックスに **Delete** と入力し、もう一度 [Delete (削除)] を選択します。

AWS Elemental MediaTailor は設定を削除します。

MediaTailor チャンネルアセンブリの使用開始

この入門チュートリアルでは、次のタスクの実行方法を示します。

- ソースロケーションを作成し、それにソースコンテンツを追加する
- チャンネルを作成する
- スケジュールに従ってチャンネルのコンテンツを再生するためのプログラムリストを作成する
- 広告挿入を使用してチャンネルストリームにパーソナライズされた AWS Elemental MediaTailor 広告を追加する

これらを完了したら、ブラウザを開き、チャンネルの再生 URL を入力して、パーソナライズ広告が含まれるチャンネルのストリームを表示できるようになります。

このチュートリアルでは、MediaTailor チャンネルアセンブリの使用を開始するための基本的な手順を説明します。詳細については、「[AWS Elemental MediaTailor を使用して線形アセンブルストリームを作成する](#)」を参照してください。

推定コスト

- アクティブなチャンネルの料金は、1 時間当たり 0.10 USD です。アクティブではないチャンネルに対する料金は請求されません。

トピック

- [前提条件](#)
- [ステップ 1: ソースロケーションを作成する](#)
- [ステップ 2: VOD ソースをソースロケーションに追加する](#)
- [ステップ 3: チャンネルを作成する](#)
- [ステップ 4: チャンネルのスケジュールにプログラムを追加する](#)
- [ステップ 5 \(オプション\): MediaTailor を使用してパーソナライズされた広告をストリームに挿入する](#)
- [ステップ 6: チャンネルの開始](#)
- [ステップ 7: チャンネルのテスト](#)
- [ステップ 8: クリーンアップする](#)

前提条件

このチュートリアルを開始する前に、これらの要件を満たす必要があります。

- [セットアップ AWS Elemental MediaTailor](#) の手順を完了しておくこと。
- VOD ソースコンテンツと広告スレートの両方に使用できるアセットがあること。アセットのマニフェストへのパスを把握しておく必要があります。

Note

自動アダプティブビットレート (ABR) またはタイトルごとのエンコーディングを使用している場合は、すべてのバリエーションが同じ長さで、同じ数の子トラックを保有するようにアセットをエンコードする必要があります。最小セグメント長が 1 秒のエンコーディングテンプレートの使用が推奨されます。

ステップ 1: ソースロケーションを作成する

ソースの場所は、コンテンツが保存されているオリジンサーバーを表します。Amazon S3、標準ウェブサーバー、コンテンツ配信ネットワーク (CDN)、パッケージングオリジンなどです AWS Elemental MediaPackage。

MediaTailor はソースロケーションからマニフェストを取得し、それらを使用して、基盤となるコンテンツセグメントを参照するライブスライディングマニフェストウィンドウをアセンブルします。

ソースロケーションを作成するには、以下の手順を実行します。

ソースロケーションを作成する

- MediaTailor コンソール (<https://console.aws.amazon.com/mediatailor/>) を開きます。
- ナビゲーションペインで、[Channel assembly] (チャネルアセンブリ) > [Source locations] (ソースロケーション) とクリックします。
- ナビゲーションバーで、[Create source location] (ソースロケーションを作成) をクリックします。
- [Source location configuration] (ソースロケーションの設定) で、ソースコンテンツの識別子と場所を入力します。
 - [Name] (名前): ソースロケーションの識別子 (my-orig など)。
 - [Base URL] (ベース URL): コンテンツがホストされているオリジンサーバーのベース URL (<https://111111111111.cloudfront.net> など)。URL は標準の HTTP URL 形式で、プレフィックスを http:// または https:// にする必要があります。
- [Create source location] (ソースロケーションを作成) をクリックします。

ステップ 2: VOD ソースをソースロケーションに追加する

チャンネル用に 1 つ、または複数のソースロケーションを定義したので、次に 1 つ、または複数の VOD ソースを追加できます。VOD ソースは、単一の映画、テレビ番組のエピソード、またはハイライトクリップといった単一のコンテンツを表します。

VOD ソースには、少なくとも 1 つのパッケージ設定を作成する必要があります。各パッケージ設定には、VOD ソースのパッケージ化された形式とマニフェストの設定が含まれています。作成後、パッケージ設定をチャンネルに追加して出力を作成します。

複数のパッケージ設定を使用して、異なるチャンネル出力を作成することができます。例えば、VOD ソースが HLS と DASH の両方としてパッケージ化されている場合、それぞれの形式に 2 つのパッケージ設定を作成できます。その後、パッケージ設定のソースグループを使用して、DASH 用と HLS 用に 1 つずつ、合計で 2 つのチャンネル出力を作成できます。

VOD ソースを追加してパッケージ設定を作成する

- MediaTailor コンソール (<https://console.aws.amazon.com/mediatailor/>) を開きます。
- ナビゲーションペインで、[Channel assembly] (チャネルアセンブリ) > [Source locations] (ソースロケーション) とクリックします。

3. [Source locations] (ソースロケーション) ペインで、[ソースロケーションを作成する](#) の手順で作成したソースロケーションを選択します。
4. [Add VOD source] (VOD ソースを追加) をクリックします。
5. [VOD source details] (VOD ソースの詳細) で、VOD ソースの [Name] (名前) (my-example-video など) を入力します。
6. [Package configurations] (パッケージ設定) > *source-group-name* と進み、パッケージ設定に関する情報を入力します。

 Note

ソースのパッケージ設定の継続時間は、すべて同じ (ソースのマニフェストによって規定されている時間) にする必要があります。また、パッケージ設定内のすべてのソースで、子ストリームの数が同じである必要もあります。これらの要件を満たすため、アセットにはエンコードテンプレートを使用することをお勧めします。最小セグメント長が 1 秒のエンコーディングテンプレートの使用が推奨されます。タイトルごと、または自動アダプティブビットレート (ABR) ストリーミングはこれらの要件に違反するため、MediaTailor ではこれらのエンコードメソッドがサポートされません。

- [Source group] (ソースグループ): HLS-4K など、このパッケージ設定を説明するソースグループ名を入力します。この名前を書き留めておきます。これは、チャンネルの出力を作成するときに参照します。詳細については、「[チャンネルの出力でのソースグループの使用](#)」を参照してください。
 - [Type] (タイプ): この設定のパッケージ形式を選択します。MediaTailor は、HLS と DASH をサポートします。
 - [Relative path] (相対パス): ソースロケーションの ベース HTTP URL からマニフェストへの相対パスです。例えば、/my/path/index.m3u8 などです。
7. [Add VOD source] (VOD ソースを追加) をクリックします。
 8. この手順のステップ 4~7 を繰り返して、広告スレートの VOD ソースを追加します。

ステップ 3: チャンネルを作成する

チャンネルは、ソースをライブリニアストリームにアセンブルします。各チャンネルには、VOD ソースのパッケージ設定に対応する 1 つ、または複数の出力が含まれています。

最初にチャンネルを作成してから、プログラムを作成することによってチャンネルのスケジュールに VOD ソースを追加します。

チャンネルを作成する

1. MediaTailor コンソール (<https://console.aws.amazon.com/mediatailor/>) を開きます。
 2. ナビゲーションペインで、[Channel assembly] (チャンネルアセンブリ) > [Channels] (チャンネル) とクリックします。
 3. ナビゲーションバーで、[Create channel] (チャンネルを作成) をクリックします。
 4. [Channel details] (チャンネルの詳細) で、チャンネルに関する詳細情報を入力します。
 - [Name] (名前): チャンネルの名前を入力します。
 - [Playback mode] (再生モード): 許可されるプログラム推移の種類と、プログラムの終了後に何が行われるかを決定します。デフォルトのループモードを使用します。
 5. [次へ] を選択します。
 6. [Output details] (出力の詳細) で、この出力の設定を定義します。
 - [Manifest name] (マニフェスト名): **index** などのマニフェスト名を入力します。MediaTailor が、形式の拡張子 (HLS には .m3u8 など) を付加します。
-  **Note**

チャンネル出力ごとに、一意のマニフェスト名を入力する必要があります。
- [Output type] (出力タイプ): チャンネルのストリーミング形式を選択します。DASH と HLS がサポートされています。[ステップ 1: ソースロケーションを作成する](#) で作成したパッケージ設定に対応する形式を選択します。
 - [Source group] (ソースグループ): [ステップ 1: ソースロケーションを作成する](#) で作成したソースグループの名前を入力します。
7. [Manifest settings] (マニフェスト設定) で、マニフェスト設定に関する追加情報を入力します。
 - [Manifest window (sec)] (マニフェストの時間枠 (秒)): 各マニフェストに含まれている時間枠 (秒単位) です。最小値は 30 秒で、最大値は 3,600 秒です。
8. [次へ] を選択します。
9. [Channel policy] (チャンネルポリシー) で、[Do not attach channel policy] (チャンネルポリシーをアタッチしない) を選択します。このオプションは、AWS アカウント 認証情報にアクセスできるユーザーのみに再生を制限します。

10. [次へ] を選択します。
11. [Review and create] (確認と作成) ペインで設定を確認します。
12. [Create channel] (チャンネルを作成) をクリックします。

 Note

チャンネルは停止状態で作成されます。チャンネルは、ユーザーがチャンネルを開始するまでアクティブになりません。

ステップ 4: チャンネルのスケジュールにプログラムを追加する

チャンネルができたので、チャンネルのスケジュールにプログラムを追加します。各プログラムには、アカウント内のソースロケーションからの VOD ソースが含まれています。チャンネルのスケジュールは、チャンネルのストリームでプログラムが再生される順序を決定します。

各プログラムには、1 つまたは複数の広告ブレイクを設定できます。広告ブレイクは、広告スレートとして使用する VOD ソースを指定することによって挿入します。広告ブレイクの継続時間は、スレートの継続時間によって決まります。オプションで、MediaTailor 広告挿入などのサーバー側の広告挿入サーバーを使用して、広告ブレイクをパーソナライズできます。

チャンネルのスケジュールにプログラムを追加する

1. MediaTailor コンソール (<https://console.aws.amazon.com/mediatailor/>) を開きます。
2. ナビゲーションペインで、[Channel assembly] (チャンネルアセンブリ) > [Channels] (チャンネル) とクリックします。
3. [Channels] (チャンネル) ペインで、[ステップ 3: チャンネルを作成する](#) 手順で作成したチャンネルを選択します。
4. [Program details] (プログラムの詳細) で、プログラムに関する詳細情報を入力します。
 - [Name] (名前): チャンネルのスケジュールに追加するプログラムの名前です。
 - [Source location name] (ソースロケーション名): [Select an existing source location] (既存のソースロケーションを選択する) を選択して、[Select an existing source location] (既存のソースロケーションを選択する) のドロップダウンメニューから [ステップ 1: ソースロケーションを作成する](#) で作成したソースロケーションを選択します。
 - [VOD source name] (VOD ソース名): [Select an existing VOD source] (既存の VOD ソースを選択する) を選択して、このチュートリアルで先ほど作成した VOD ソースを選択します。

5. [Playback configuration] (再生設定) で、プログラムがチャンネルのスケジュールにいつ、どのように挿入されるかを定義します。
 - [Transition type] (推移タイプ): この値は [Relative] (相対的) に固定されています。相対的な遷移タイプは、このプログラムがプログラムリスト内の他のプログラムに相対して配信されることを示します。
 - [Relative position] (相対位置): このプログラムがチャンネルのスケジュールの最初のプログラムである場合は、この設定を省略できます。チャンネルのスケジュールの最初のプログラムではない場合は、プログラムを付加するプログラムリスト内の場所を選択します。これには、[Before program] (プログラムの前) または [After program] (プログラムの後) を選択できます。
 - [Relative program] (相対プログラム): これがスケジュールの最初のプログラムである場合は、この設定を省略できます。チャンネルのスケジュールの最初のプログラムではない場合は、[Use existing program] (既存のプログラムを使用する) を選択して、[チャンネルのスケジュールにプログラムを追加する](#) で作成したプログラム名を選択します。
6. [Add ad break] (広告ブレイクを追加) をクリックします。[Ad breaks] (広告ブレイク) で、広告ブレイクを設定します。
 - [Slate source location name] (スレートのソースロケーション名): [Select an existing source location] (既存のソースロケーションを選択する) を選択して、このチュートリアルで先ほど作成したスレートが保存されているソースロケーションを選択します。
 - [VOD source name] (VOD ソース名): [Select an existing VOD source] (既存の VOD ソースを選択する) を選択して、このチュートリアルで先ほど追加したスレート用に使用している VOD ソースを選択します。広告ブレイクの継続時間は、スレートの継続時間によって決まります。
 - [Offset in milliseconds] (ミリ秒単位のオフセット): この値は、ミリ秒単位での広告ブレイクの開始時間を、プログラムの開始時点に相対するオフセットとして指定します。VOD ソースの期間より短く、プログラムの VOD ソース内のすべてのトラック (すべてのオーディオ、ビデオ、クローズドキャプショントラック) のセグメント境界と一致する値を入力します。そうしないと、広告ブレイクはスキップされます。例えば、0 を入力すると、プログラムの開始前に再生されるプレロール広告ブレイクが作成されます。注: 。
7. [Add program] (プログラムを追加) をクリックします。

プログラムの詳細については、「[Configuring ad breaks for your program](#)」を参照してください。

リニアストリームでの広告の使用の詳細については、「[オプションの設定](#)」を参照してください。

ステップ 5 (オプション): MediaTailor を使用してパーソナライズされた広告をストリームに挿入する

これでプログラムが含まれるチャンネルが作成されました。希望する場合は、MediaTailor を使用して、チャンネルのストリーム内のプログラムにある広告ブレイクにパーソナライズされた広告を挿入できます。

前提条件

続行する前に、以下の要件を満たす必要があります。

- 広告決定サーバー (ADS) があること。
- [チャンネルのスケジュールにプログラムを追加する](#) 手順での [Ad break] (広告ブレイク) の設定が完了していること。

MediaTailor を使用してチャンネルのストリームにパーソナライズされた広告を追加する

1. MediaTailor コンソール (<https://console.aws.amazon.com/mediatailor/>) を開きます。
2. ナビゲーションペインで [Configurations] (設定) をクリックします。
3. [Required settings] (必須の設定) で、設定に関する基本的な必須情報を入力します。
 - [Name] (名前): 設定の名前です。
 - [Content source] (コンテンツソース): チャンネルの出力からの再生 URL (ファイル名と拡張子を除く) を入力します。MediaTailor の設定に関する詳細については、「[必須の設定](#)」を参照してください。
 - [Add decision server] (広告決定サーバー): ADS の URL を入力します。
4. オプションで、[Configuration aliases] (設定エイリアス)、[Personalization details] (パーソナライゼーションの詳細)、および [Advanced settings] (アドバンスド設定) を設定できます。これらの設定については、「[オプションの設定](#)」を参照してください。
5. ナビゲーションバーで [Create configuration] (設定を作成) をクリックします。

MediaTailor 広告挿入の使用に関する詳細については、「[AWS Elemental MediaTailor を使用して広告を挿入する](#)」を参照してください。

ステップ 6: チャンネルの開始

これでチャンネルが作成されました。ただし、チャンネルのストリームにアクセスする前に、チャンネルを開始する必要があります。チャンネルがアクティブになる前にアクセスしようとすると、MediaTailor が HTTP 4xx エラーコードを返します。

チャンネルの開始

1. MediaTailor コンソール (<https://console.aws.amazon.com/mediatailor/>) を開きます。
2. ナビゲーションペインで、[Channel assembly] (チャンネルアセンブリ) > [Channels] (チャンネル) とクリックします。
3. ナビゲーションバーで、[Start] (開始) をクリックします。

ステップ 7: チャンネルのテスト

チャンネルが正常に動作していることを確認するには、ウェブブラウザを開き、チャンネルの出力からの URL を入力します。チャンネルのストリームが表示されるはずですが、

状況によっては、期待される動作を実現するために、キャッシュの消去が必要になる場合があります。

ステップ 8: クリーンアップする

このチュートリアルで作成したチャンネルを使用し終わったら、チャンネルを削除してクリーンアップする必要があります。

チャンネル料金の請求は、チャンネルステータスが [停止] に変わり次第、停止されます。今後のためにチャンネルを保持したいが、料金は発生させたくないという場合は、当面の間チャンネルを停止して、後ほど再度開始することができます。

チャンネルを削除する

1. MediaTailor コンソール (<https://console.aws.amazon.com/mediatailor/>) を開きます。
2. ナビゲーションペインで、[Channel assembly] (チャンネルアセンブリ) > [Channels] (チャンネル) とクリックします。

3. 削除するチャンネルを選択します。
4. チャンネルが実行中の場合は、[Actions] (アクション) ドロップダウンメニューから [Stop] (停止) を選択します。チャンネルは、停止してから削除する必要があります。
5. チャンネルが停止したら、[Actions] (アクション) ドロップダウンメニューから [Delete] (削除) を選択します。

AWS Elemental MediaTailor を使用して広告を挿入する

設定は、AWS Elemental MediaTailor内で操作するオブジェクトです。設定には、オリジンサーバーと広告決定サーバー (ADS) のマッピング情報が保持されています。広告が利用できない場合、または広告が ad avail 全体を埋めない場合に MediaTailor が使用するデフォルトの再生を定義することもできます。

MediaTailor でコンテンツ配信ネットワーク (CDN) を使用する場合は、CDN 情報を設定に追加する前に CDN で動作ルールを設定する必要があります。CDN のセットアップに関する詳細については、「[CDN を使用して MediaTailor の広告パーソナライゼーションとコンテンツ配信を最適化する](#)」を参照してください。

トピック

- [サポートされているオーディオおよびビデオコーデック](#)
- [AWS Elemental MediaTailor 広告挿入動作について](#)
- [MediaTailor サーバーガイド広告挿入の概要と実装](#)
- [MediaTailor 広告サーバー統合要件](#)
- [MediaTailor 再生設定の管理](#)
- [MediaTailor 広告挿入用のコンテンツソースの統合](#)
- [Google の広告プラットフォーム AWS Elemental MediaTailor との統合](#)
- [広告ブレイク抑制による広告ブレイク動作のカスタマイズ](#)
- [MediaTailor バンパー広告挿入](#)
- [MediaTailor プリロール広告挿入](#)
- [MediaTailor スレート広告挿入](#)
- [での広告トランスコーディング AWS Elemental MediaTailor](#)
- [広告のプリフェッチ](#)
- [での前提条件付き広告の使用 AWS Elemental MediaTailor](#)
- [MediaTailor 収益化関数の使用](#)
- [ADS リクエストの MediaTailor 動的広告変数](#)
- [MediaTailor マニフェストクエリパラメータ](#)
- [広告追跡データの報告](#)
- [オーバーレイ広告](#)
- [広告 ID デコレーション](#)

サポートされているオーディオおよびビデオコーデック

MediaTailor は、次のコーデックをサポートしています。

- オーディオコーデック: mp4a、ac-3、および ec-3
- ビデオコーデック: h.264 (AVC)、h.265 (HEVC)、av01 (AV1)

AWS Elemental MediaTailor 広告挿入動作について

AWS Elemental MediaTailor は、オリジンマニフェストに広告を置き換えるか挿入することで、ライブまたはビデオオンデマンド (VOD) コンテンツに広告をステッチします。広告が挿入されるか置き換えられるかは、オリジンマニフェストでの広告ブレイクの設定方法と、コンテンツが VOD とライブのどちらであるかによって決まります。広告時間枠は、広告が表示されるプログラミング中の期間であり、広告表示は広告時間枠内の広告時間の特定単位であり、広告で埋めることができます。

- 広告置換では、MediaTailor がコンテンツセグメントを広告に置き換えます。
- 広告挿入では、MediaTailor がセグメントが存在しない場所に広告コンテンツを挿入します。

MediaTailor がライブおよび VOD コンテンツに広告をステッチする方法については、該当するトピックを選択してください。

トピック

- [VOD に対する広告ステッチ動作](#)
- [ライブ広告ステッチング動作](#)

VOD に対する広告ステッチ動作

MediaTailor は、オリジンマニフェストでの広告マーカの設定方法、および広告決定サーバー (ADS) が VMAP レスポンスを送信するかどうかに基づいて、VOD コンテンツでの広告の挿入または置き換えを実行します。

マーカ設定による広告動作については、以下のセクションを参照してください。

広告マーカが存在する場合

AWS Elemental MediaTailor は、SCTE-35 広告マーカがオリジンマニフェストに存在する広告を挿入します。EXT-X-CUE-OUT 値が 0 時間の広告マーカは、広告挿入を示しています。

HLS 広告マーカのガイドライン

ポストロールおよび広告ポッドの SCTE シグナル通知については、以下のガイドラインに従ってください。

プレロール広告

HLS のポストロールの場合、最後のコンテンツセグメントの前に CUE-OUT/IN マーカを置く必要があります。HLS 仕様では、タグのデコレータがセグメントの前で明示的に宣言される必要があるためです。

例えば、以下の宣言を検討してみましょう。

```
#EXT-X-CUE-OUT: 0
#EXT-X-CUE-IN
#EXTINF:4.000,
Videocontent.ts
#EXT-X-ENDLIST
```

AWS Elemental MediaTailor は次のようなポストロールを挿入します。

```
#EXTINF:4.000,
Videocontent.ts
#EXT-X-DISCONTINUITY
#EXTINF:3.0,
Adsegment1.ts
#EXTINF:3.0,
Adsegment2.ts
#EXTINF:1.0,
Adsegment3.ts
#EXT-X-ENDLIST
```

Example 2: 広告ポッド

CUE-OUT/IN タグは、セグメントに明示的にアタッチされる必要があります。広告ポッドの動作を模倣するために、複数の CUE-OUT/IN タグを連続して使用することはできません。

例えば、以下の宣言は、広告ポッドを表現するための CUE-OUT/IN の有効な使用方法です。

```
#EXT-X-CUE-OUT: 0
#EXT-X-CUE-IN
#EXTINF:4.000,
Somecontent1.ts
```

```
#EXT-X-CUE-OUT: 0
#EXT-X-CUE-IN
#EXTINF:4.000,
Somecontent2.ts
#EXT-X-CUE-OUT: 0
#EXT-X-CUE-IN
#EXTINF:4.000,
Videocontent.ts
```

上記の宣言により、以下のような出力が得られます。

```
Ad 1
Somecontent.ts
Ad 2
Somecontent2.ts
Videocontent.ts
Post-Roll Ad 3
```

以下の宣言は無効です。

```
#EXT-X-CUE-OUT: 0
#EXT-X-CUE-IN
#EXT-X-CUE-OUT: 0
#EXT-X-CUE-IN
#EXT-X-CUE-OUT: 0
#EXT-X-CUE-IN
#EXTINF:4.000,
Videocontent.ts
```

広告マーカが存在しない場合

広告マーカは、マニフェストで広告ブレイクをシグナリングするために推奨される方法です。広告マーカは必須ではありません。マニフェストに DASH または HLS の広告マーカが含まれていない場合、MediaTailor は ADS を 1 回呼び出し、レスポンスに基づいて広告ブレイクを作成します。

- ADS が VAST レスポンスを送信する場合、MediaTailor はマニフェストの冒頭にある広告ブレイクにレスポンスからのすべての広告を挿入します。これがプレロールです。
- ADS が VMAP レスポンスを送信する場合、MediaTailor は広告ブレイクの時間オフセットを使用してブレイクを作成し、それらを指定されたタイミング (プレロール、ミッドロール、またはポストロール) でマニフェスト全体に挿入します。MediaTailor は、マニフェスト内の各広告ブレイクに、VMAP レスポンス内の各広告ブレイクからのすべての広告を使用します。

Note

セグメントで挿入ポイントと VOD コンテンツの VMAP が重複する場合は、MediaTailor が最も近い挿入ポイントに切り下げます。

Tip

ミッドロール広告ブレイクを作成したいが、ADS が VMAP をサポートしないという場合は、マニフェストに広告マーカがあることを確認してください。MediaTailor は、次のセクションの説明にあるように、マーカの場所に広告を挿入します。

Note

サーバーガイド広告挿入方法の場合、MediaTailor はマニフェストの上部にプリロール広告を挿入し、プレイヤーは他の広告タイプの前にそれらを再生します。

ライブ広告ステッチング動作

ライブストリームでは、AWS Elemental MediaTailor は常に広告置換を実行し、広告マーカ間の合計時間を可能な限り厳密に保ちます。広告マーカに DURATION 属性が含まれる場合、MediaTailor はその値を使用して広告ブレイクの継続時間を判断します。ライブワークフローでは、すべての CUE-OUT インジケータに期間または一致する CUE-IN インジケータが必要です。

MediaTailor は、HLS および DASH のライブコンテンツに対して広告置換を実行します。MediaTailor が広告ブレイクの配置とタイミングを計算する方法については、「[the section called “広告マーカ”](#)」と「[the section called “広告マーカ”](#)」を参照してください。

広告の選択と置き換え

AWS Elemental MediaTailor には、広告決定サーバー (ADS) VAST レスponsから広告が次のように含まれます。

- 継続時間が指定されている場合、MediaTailor はその時間内に収まる一連の広告を選択して、それらを含めます。

- 継続時間が指定されていない場合、MediaTailor は、メインコンテンツに戻ることを示す広告マーカースに到達するまで、可能な限り多くの広告を再生します。

AWS Elemental MediaTailor は、ライブ広告の置き換え中に次のガイドラインに従います。

- MediaTailor は、広告をトリミングしたりカットしたりすることなく、完全に再生しようとします。
- 広告ブレイクの終了を示す広告マーカースに到達すると、MediaTailor は常に基盤となるコンテンツに戻ります。これは、現在再生中の広告の短縮化につながる可能性があります。
- 継続時間が終了すると、MediaTailor は基盤となるコンテンツに戻ります。
- 広告ブレイクの継続時間中に再生する広告がなくなると、MediaTailor はスレートを再生する (スレートが設定されている場合)、または基盤となるコンテンツストリームの再生を再開します。これは通常、広告ブレイクの継続時間を埋めるために十分なトランスコード済みの広告がない場合に発生します。

Tip

ブレイク内で許容される広告が配信されない時間の制限は、パーソナライゼーションのしきい値設定を使用して定義できます。詳細については、[PlaybackConfiguration リファレンス](#)を参照してください。

サーバーガイド広告挿入のライブプリロール

ライブプリロールは、サーバー側の広告挿入と比較して、サーバーガイドの広告挿入方法では動作が異なります。

サーバー側の広告挿入 (ステッチモード)

プレロール広告は、各視聴者のセッションの開始時にライブコンテンツの一部を置き換えます。各ビューワーは、ストリームに参加するタイミングに応じて異なるタイミングでプリロールを表示します。

サーバーガイド広告挿入方法

MediaTailor は、CUE="PRE,ONCE"属性を持つすべてのメディアマニフェストの上部にプリロール日付範囲タグを配置します。これにより、プレイヤーは同じパーソナライズされていないマニ

フェストを共有しているにもかかわらず、再生の開始時に一度プレロール広告をリクエストして再生します。

設定要件:

- ライブプリロール広告決定サーバー: プリロール広告の VAST エンドポイントを設定する (ミッドロール広告とは異なる場合があります)
- ライブプリロールの最大許容期間: プリロール広告の最大期間を設定します (オプション - 省略すると、返されたすべての広告が使用されます)。

技術的な実装:

- 事前ロール日付範囲タグの使用 `START-DATE="1970-01-01T00:00:00.000Z"` (Unix エポック)
- プリロールのアセットリストリクエストは、通常の ADS の代わりに設定されたプリロール広告決定サーバーを使用します。
- プレイヤーは、アセットリストデータの `availId="aws-mediatailor-preroll-1"` を通じてプリロールリクエストを識別します。

Important

ライブストリームの場合、プリロール広告はコンテンツを遅延させるのではなくカバーします。今後のバージョンでは、追加の設定オプションを使用してコンテンツ遅延モードをサポートする場合があります。

Note

プリロールの動作は、サーバーガイド広告挿入のライブコンテンツと VOD コンテンツによって異なります。ライブコンテンツには明示的なプリロール設定が必要ですが、VOD コンテンツにはデフォルトで通常の広告決定サーバーを使用したプリロールが含まれます。

例

- 広告時間枠の期間を 70 秒に設定し、ADS レスポンスに 40 秒の広告が 2 つ含まれている場合、は 40 秒の広告のいずれかを AWS Elemental MediaTailor 再生します。残った時間は、設定された

スレート、または基盤となるコンテンツに切り替えられます。このプロセス中に MediaTailor がキューインインジケータに到達した場合は、その時点で直ちに基盤となるコンテンツに切り替わります。

- 広告ブレークの継続時間が 30 秒に設定されており、ADS レスポンスから提供された最短の広告が 40 秒である場合、MediaTailor は広告を再生しません。広告スレートが設定されている場合、MediaTailor はスレートを 30 秒間、またはキューインインジケータに到達するまで再生します。広告スレートが設定されていない場合、MediaTailor は基盤となるコンテンツを再生します。

MediaTailor サーバーガイド広告挿入の概要と実装

AWS Elemental MediaTailor サーバーガイド広告挿入 (SGAI) は、広告をメディアプレイリストに直接ステッチするのではなく、個別のプレイリストとして参照することで、サーバー側の広告挿入に代わる手段を提供します。このアプローチにより、キャッシュ可能なマニフェストによるパフォーマンスが向上し、スケーラビリティが向上します。

MediaTailor でサーバーガイド広告挿入を使用する方法については、次のリストから該当するトピックを選択してください。

再生設定で を有効にする

プレイヤーがサーバーガイド広告挿入を使用できるようにするには、MediaTailor 再生設定 `PLAYER_SELECT` で `Insertion Mode` に設定する必要があります。これにより、プレイヤーはセッション初期化時にステッチまたはガイド付き広告挿入を選択できます。

サーバーガイドセッションを作成する

再生セッションを作成するときは、ガイドモードを選択します。これを行う方法は、プレイヤーが暗黙的なセッションを使用するか明示的なセッションを使用するかによって異なります。

暗黙的に作成されたサーバーガイド付きセッション

HLS マルチバリエーションプレイリストリクエスト `aws.insertionMode=GUIDED` に追加します。例:

```
playback-endpoint/v1/master/hashed-account-id/origin-id/index.m3u8?  
aws.insertionMode=GUIDED
```

コードの説明は以下のとおりです。

- playback-endpoint は、設定を作成したときに AWS Elemental MediaTailor によって生成された固有の再生エンドポイントです。

例

```
https://777788889999.mediatailor.us-east-1.amazonaws.com
```

- hashed-account-id は AWS アカウント ID です。

例

```
777788889999
```

- origin-id は、設定を作成したときに付けた名前です。

例

```
myOrigin
```

- index.m3u8 または は、テストストリームからのマニフェストの名前とそのファイル拡張子です。これは、[the section called “ステップ 4: 設定を作成する”](#) で設定した動画コンテンツソースにマニフェストを付加するときに、マニフェストが完全に識別されるように定義します。

上記の例の値を使用すると、完全な URL は以下ようになります。

- 例:

```
https://777788889999.mediatailor.us-east-1.amazonaws.com/v1/master/777788889999/  
myOrigin/index.m3u8?aws.insertionMode=GUIDED
```

明示的に作成されたサーバーガイドセッション

プレイヤーが MediaTailor 設定のセッション初期化プレフィックスエンドポイントPOSTに HTTP で送信する JSON メタデータinsertionMode=GUIDEDに を追加します。

次の例は、JSON メタデータの構造を示しています。

```
{  
  # other keys, e.g. "adsParams"
```



```
"insertionMode": "GUIDED"      # this can be either GUIDED or STITCHED
}
```

この初期化メタデータでは、再生セッションはセラーガイド付き広告挿入を使用します。

SGAI による広告追跡

SGAI は、サーバー側とクライアント側の広告追跡の両方をサポートしています。セッションの初期化時にレポートモードを設定します。セッション中にモードを変更することはできません。

サーバー側の追跡 (デフォルト)

MediaTailor は、プレイヤーが広告セグメントをリクエストすると VAST ビーコンを自動的に発射します。アセットリストの広告 URIs には、暗号化されたビーコンメタデータ (awsBeaconData、awsBeaconDomain、) が含まれていますawsConfigurationName。プレイヤーは HLS #EXT-X-DEFINE:QUERYPARAM変数置換をサポートする必要があります。アセットリストレスポンスには TRACKINGセクションは含まれません。

SGAI サーバー側のビーコンの仕組みの詳細については、「」を参照してください[サーバーガイド広告挿入によるサーバー側の追跡 \(SGAI\)](#)。

クライアント側の追跡

セッション初期化リクエストaws.reportingMode=CLIENTに を追加します。アセットリストレスポンスには、広告の再生中にプレイヤーが起動するビーコン URLs を含むTRACKINGセクションが含まれます。GetTracking API エンドポイントは SGAi セッションには使用されません。代わりに、各アセットリストレスポンスには、データの直接追跡が含まれます。追跡データは、サーバー側の広告挿入 (SSAI) 追跡レスポンスと同じ JSON スキーマを使用します。

詳細については、「[サーバーガイド広告挿入](#)」を参照してください。

マニフェストハートビートを使用したガイド付きプリフェッチ

ライブ SGAi ストリームの場合、セッション初期化リクエストaws.guidedPrefetchMode=MANIFESTに を追加することで、マニフェストベースの広告プリフェッチを有効にできます。

```
https://7777888899999.mediataylor.us-east-1.amazonaws.com/v1/master/7777888899999/myOrigin/index.m3u8?aws.insertionMode=GUIDED&aws.guidedPrefetchMode=MANIFEST
```

有効にすると、MediaTailor はセッション識別子 (?aws.sessionId=<id>) をクエリパラメータとして、多変量プレイリストの各インタースティシャルメディアマニフェスト (/v1/i-media) URL に追加します。プレイヤーが i メディアマニフェストを更新するたびに、リクエストはセッション ID で MediaTailor に到達します。これは、MediaTailor が今後の広告時間枠のプリフェッチリクエストをキューに入れるために使用されます。

- aws.guidedPrefetchMode パラメータは、MANIFEST (有効) と OFF (無効、デフォルト) の 2 つの値を受け入れます。
- ガイド付きプリフェッチモードは、SGAI セッションでのみ有効です。ステッチセッションで使用する、エラーが返されます。
- DASH はガイド付きプリフェッチモードをまだサポートしていません。
- ガイド付きプリフェッチはレポートモードとは無関係です。ビーコンはプリフェッチ時ではなく再生時に発生します。
- ガイド付きプリフェッチを使用する場合は、i メディアマニフェストを CDN にキャッシュしないでください。プリフェッチメカニズムは、MediaTailor に直接到達するプレイヤーのマニフェスト更新リクエストによって異なります。CDN が /v1/i-media レスポンスをキャッシュする場合、MediaTailor はハートビートリクエストを受信せず、プリフェッチをトリガーできません。

MediaTailor サーバーガイド広告挿入機能の互換性マトリックス

AWS Elemental MediaTailor には、機能の互換性が異なる 2 つの広告挿入方法が用意されています。サーバーガイド広告挿入はサーバー側の広告挿入とは異なり、一部の MediaTailor 機能との互換性に影響します。この表を使用して、各広告挿入方法でどの機能が機能するかを理解します。

広告挿入方法による機能の互換性

機能	サーバー側の広告挿入 (SSAI)	サーバーガイド広告挿入 (SGAI)
広告プリフェッチ	✓ サポート対象	まだサポートされていません。
広告抑制	✓ サポート対象	該当しない
ロール前の広告動作	MediaTailor 設定で制御	MediaTailor 設定で制御

機能	サーバー側の広告挿入 (SSAI)	サーバーガイド広告挿入 (SGAI)
クライアント側の広告追跡	GetTracking API を使用する	アセットリストで TRACKINGセクションを使用 (GetTracking API は使用されません)
サーバー側の広告追跡	✓ サポート — セッション ID を使用した/v1/segment リクエストに基づいてビーコンが起動します	✓ サポート (HLS のみ) — を介して広告 URIs に埋め込まれた暗号化されたビーコンデータを使用したセッションレスビーコンを使用します#EXT-X-DEFINE:QUERYPARAM 。HLS v11以降が必要です。DASH はまだサポートされていません。
広告 ID デコレーション	✓ サポート対象	✗ 互換性がない

互換性の詳細

広告プリフェッチ

広告プリフェッチは現在サポートされていません。

広告抑制

プレイヤーは再生する広告のみを取得するため、広告抑制はサーバーガイドの広告挿入方法ではサポートされていません。

ロール前の広告動作

プリロール広告タイミングは、挿入方法によって異なります。

- サーバー側の広告挿入: MediaTailor は、設定に基づいてプレロール広告が再生されるタイミングを制御します
- サーバーガイド広告挿入: MediaTailor はマニフェストの上部にプリロール広告を挿入します。プレイヤーは最初にこれらの広告を表示し、コンテンツを開始します。

広告の追跡

クライアント側の追跡は、広告挿入方法に応じて異なるメカニズムを使用します。

- サーバー側の広告挿入 (SSAI): GetTracking API エンドポイントを使用します
- サーバーガイド広告挿入 (SGAI): MediaTailor は、各アセットリストレスポンスの TRACKING セクションで追跡情報を提供します。GetTracking API エンドポイントは使用されません。セッション初期化レスポンスには含まれません `trackingUrl`。

サーバー側の追跡もメソッドによって異なります。

- サーバー側の広告挿入 (SSAI): MediaTailor は、プレイヤーがセッション ID `/v1/segment/` を使用して を介してステッチされた広告セグメントを取得するときにビーコンを起動します。
- サーバーガイド広告挿入 (SGAI): MediaTailor はセッションレスビーコンを使用します。MediaTailor は `awsBeaconData`、アセットリストで返される広告マニフェスト URIs に暗号化されたビーコンデータ (、`awsBeaconDomain`、`awsConfigurationName`) を埋め込みます。広告マニフェストは `#EXT-X-DEFINE:QUERYPARAM` タグを使用するため、プレイヤーはこれらの値をセグメント URLs。プレイヤーが各広告セグメントをリクエストすると、MediaTailor はデータを復号し、適切なビーコンを起動して、コンテンツセグメントにリダイレクトします。サーバー側のレポートがアクティブな場合、MediaTailor はアセットリストレスポンスから TRACKING セクションを省略します。詳細については、「[サーバーガイド広告挿入によるサーバー側の追跡 \(SGAI\)](#)」を参照してください。

広告 ID デコレーション

広告 ID デコレーションは、X-AD-CREATIVE-SIGNALING ヘッダーを入力するフィールドは、マニフェストが書き込まれるときではなく、アセットリストがフェッチされたときにのみ認識されるため、サーバーガイド広告挿入と互換性がありません。

ライブストリームの MediaTailor サーバーガイド広告挿入設定

AWS Elemental MediaTailor ライブコンテンツ用のサーバーガイド広告挿入は、キャッシュ可能なマニフェストを通じてパフォーマンスに大きなメリットをもたらします。ライブコンテンツ用に SGAI を設定すると、VOD と同じコアパラメータが使用されますが、ライブストリームの特性とリアルタイム処理に関する特定の考慮事項があります。

ライブ SGAI の要件

ライブコンテンツの SGAI を有効にする前に、以下があることを確認してください。

- ライブストリームに正しくフォーマットされた DATERANGE マーカーが含まれている
- 広告時間枠の期間は一貫性があり、予測可能です
- CDN は SGAI マニフェストを適切にキャッシュするように設定されています
- プレイヤーがサーバーガイドの広告挿入ワークフローをサポート
- 広告決定サーバーがライブコンテンツのリアルタイムリクエストを処理できる

プレイヤーの要件

プレイヤーは、SGAI ライブマニフェストを適切に処理するように設定する必要があります。

- サーバーガイド広告挿入ワークフローのサポート
- マニフェストからの広告挿入ガイダンスを処理する機能
- ライブストリームのタイミングと同期の適切な処理
- HLS コンテンツの場合: CLASS 属性を使用した HLS バージョン 8 と EXT-X-DATERANGE のサポート。サーバー側のビーコン用のバージョン 11。
- HLS コンテンツの場合: EXT-X-DEFINE 変数置換のサポート

ライブ再生設定

ライブコンテンツの SGAI を有効にするには、以下の設定で再生設定を作成します。

Example SGAI ライブ再生設定

```
{
```

```
"Name": "LiveSGAIConfig",  
"VideoContentSourceUrl": "https://your-live-origin.com/live/stream.m3u8",  
"AdDecisionServerUrl": "https://your-ads.com/ads",  
"PersonalizationThresholdSeconds": 1,  
"InsertionMode": "PLAYER_SELECT"  
}
```

以下は、ライブ SGAI 設定に関する重要な考慮事項です。

VideoContentSourceUrl

適切にフォーマットされた SCTE-35 DATERANGE マーカーを持つライブ HLS ストリームを指す必要があります。ストリームは、一貫したセグメント期間とビットレートバリエーションを維持する必要があります。

SGAI ライブマニフェストリクエスト

SGAI ライブマニフェストは、従来の広告挿入と同じ URL パターンを使用します。

```
https://your-config.mediatailor.region.amazonaws.com/v1/master/config-name/  
manifest.m3u8?aws.insertionMode=GUIDED
```

ライブ SGAI のマニフェストベースのプリフェッチ

ライブ SGAI ワークフローでは、マニフェストベースのプリフェッチハートビートを有効にして、広告フィルレイテンシーを減らすことができます。マニフェストリクエスト `aws.guidedPrefetchMode=MANIFEST` に追加します。

```
https://your-config.mediatailor.region.amazonaws.com/v1/master/config-name/  
manifest.m3u8?aws.insertionMode=GUIDED&aws.guidedPrefetchMode=MANIFEST
```

有効にすると、MediaTailor はセッション識別子 (`?aws.sessionId=<id>`) をクエリパラメータとして、多変量プレイリストの各インタースティシャルメディアマニフェスト (`/v1/i-media`) URL に追加します。プレイヤーが i メディアマニフェストを更新するたびに、リクエストはセッション ID で MediaTailor に到達します。これは、MediaTailor がセッションを識別し、今後の広告時間枠のプリフェッチリクエストをキューに入れるために使用されます。

⚠ Important

ガイド付きプリフェッチを使用する場合は、i メディアマニフェストを CDN にキャッシュしないでください。プリフェッチハートビートメカニズムは、MediaTailor に直接到達するプレイヤーのマニフェスト更新リクエストによって異なります。CDN がキャッシュして/v1/i-mediaレスポンスを提供する場合、MediaTailor はハートビートリクエストを受信せず、プリフェッチをトリガーできません。aws.guidedPrefetchMode=MANIFEST が使用中のときに MediaTailor に/v1/i-media/*リクエストを渡すように CDN を設定します。

ガイド付きプリフェッチは、レポートモードから独立しています。サーバー側 (デフォルト) 追跡とクライアント側 (aws.reportingMode=CLIENT) 追跡のどちらを使用する場合でも、ビーコンは広告がプリフェッチされたときではなく再生時に発生します。MediaTailor での広告プリフェッチの仕組みに関する一般的な情報については、「」を参照してください[広告のプリフェッチ](#)。

SGAI ライブ設定のテスト

SGAI ライブセットアップを以下の検証ステップで検証します。

1. テストマニフェストの生成

SGAI ライブマニフェスト URL をリクエストし、適切な広告挿入ガイダンスを使用してキャッシュ可能なコンテンツを返すことを確認します。

2. CDN キャッシュの検証

CDN が設定された TTL 値に従って SGAI マニフェストをキャッシュしていることを確認します。

3. 広告挿入をテストする

SGAI マニフェストで提供されているガイダンスを使用して、プレイヤーが広告を正常に挿入できることを確認します。

4. パフォーマンスのモニタリング

CloudWatch メトリクスを使用して、オリジン負荷の低減とキャッシュヒット率の向上を検証します。

VOD コンテンツの MediaTailor サーバーガイド広告挿入設定

AWS Elemental MediaTailor VOD コンテンツのサーバーガイド付き広告挿入は、キャッシュ性の高いマニフェストとサーバー処理の削減により、パフォーマンスに大きなメリットをもたらします。VOD コンテンツ用に SGAI を設定すると、video-on-demandアセットの静的な性質を活用してキャッシュ効率を最大化し、オリジンリクエストを最小限に抑えることができるため、表示パターンが繰り返される大規模なコンテンツライブラリに最適です。

VOD SGAI の要件

SGAI for VOD コンテンツを有効にする前に、以下があることを確認してください。

- VOD コンテンツには、適切にフォーマットされた広告マーカ (SCTE-35 または時間指定メタデータ) が含まれています。
- コンテンツは、一貫した可用性を備えた信頼性の高いオリジンに保存されます。
- CDN は、適切な TTL 値を使用して SGAI マニフェストをキャッシュするように設定されています
- プレイヤーがサーバーガイドの広告挿入ワークフローをサポート
- 広告決定サーバーは VOD 固有のメタデータとターゲティングを処理できます

プレイヤーの要件

プレイヤーは、SGAI VOD マニフェストと広告挿入を処理するように設定する必要があります。

- サーバーガイド広告挿入ワークフローのサポート
- VOD マニフェストからの広告挿入ガイダンスを処理する機能
- VOD 再生中のクライアント側の広告挿入のサポート
- 広告時間枠全体でのシークオペレーションの適切な処理
- コンテンツ期間と位置追跡のサポート

VOD 再生設定

VOD コンテンツの SGAI を有効にするには、次の設定で再生設定を作成します。

Example SGAI VOD 再生設定

```
{
```



```
"Name": "VODSGAIConfig",  
"VideoContentSourceUrl": "https://your-vod-origin.com/content/",  
"AdDecisionServerUrl": "https://your-ads.com/ads",  
"PersonalizationThresholdSeconds": 5,  
"InsertionMode": "PLAYER_SELECT"  
}
```

VOD SGAI 設定の主な考慮事項は次のとおりです。

VideoContentSourceUrl

一貫した URL パターンを持つ VOD コンテンツライブラリを指す必要があります。オリジンが予想されるリクエストボリュームを処理でき、信頼性の高いコンテンツ配信を提供できることを確認します。

ConfigurationAliases

マニフェストのキャッシュ可能性に影響を与えることなく、広告ターゲティングに使用できるコンテンツ期間、ジャンル、シリーズ情報などの VOD 固有のパラメータを含めます。

ManifestProcessingRules

広告マーカのパススルーを有効にして、元のコンテンツのタイミング情報を保持します。これは、定義済みの広告ブレイク位置を持つ VOD コンテンツにとって特に重要です。

SGAI VOD マニフェストリクエスト

SGAI VOD マニフェストは、従来の VOD 広告挿入と同じ URL パターンを使用します。

```
https://your-config.mediatailor.region.amazonaws.com/v1/master/config-name/content-path/manifest.m3u8?aws.insertionMode=GUIDED
```

VOD 固有の広告ターゲティング

VOD コンテンツは、独自の広告ターゲティングの機会を可能にします。

コンテンツメタデータのターゲティング

VOD コンテンツメタデータを活用して広告ターゲティングを改善します。

- ジャンルとカテゴリ: コンテンツタイプ (ドラマ、コメディ、ドキュメンタリー) に基づくターゲット広告
- コンテンツ評価: 年齢に適した広告コンテンツを確保する (G、PG、R 評価)
- シリーズとシーズン: シリーズ継続性または関連コンテンツのターゲット広告
- リリース日: コンテンツ経過時間に基づくターゲット (新規リリースとカタログコンテンツ)
- コンテンツ期間: コンテンツの合計長に基づいて広告ロードを調整する

コンテキストターゲティングの表示

VOD 表示パターンにより、コンテキストに応じた広告ターゲティングが可能になります。

- 時刻: コンテンツが視聴されるタイミングに基づいて広告をターゲットにする
- ヒンジウォッチング: 複数のエピソードを視聴するユーザーの広告頻度を調整する
- 完了率: ユーザーの過去のコンテンツ完了パターンに基づくターゲット
- デバイスタイプ: デバイスを表示するための広告形式を最適化する (テレビ、モバイル、タブレット)

SGAI VOD 設定のテスト

以下の検証ステップで SGAI VOD の設定を確認します。

1. テストマニフェストの生成

さまざまなコンテンツタイプの SGAI VOD マニフェスト URLs をリクエストし、適切な広告挿入ガイダンスに従ってキャッシュ可能なコンテンツを返すことを確認します。

2. CDN キャッシュの検証

CDN が設定された TTL 値に従って SGAI マニフェストをキャッシュし、高いキャッシュヒット率を達成していることを確認します。

3. 広告挿入をテストする

さまざまな VOD コンテンツの SGAI マニフェストで提供されているガイダンスを使用して、プレイヤーが広告を正常に挿入できることを確認します。

4. シーク操作をテストする

VOD 内で探すコンテンツが広告時間枠全体で正しく動作し、適切な再生位置が維持されていることを確認します。

5. パフォーマンスのモニタリング

CloudWatch メトリクスを使用して、オリジン負荷の低減、キャッシュヒット率の向上、広告挿入率の成功を検証します。

主要なテストシナリオ

これらの特定の VOD シナリオをテストします。

- 人気コンテンツ: 頻繁にアクセスされる VOD アセットの高いキャッシュヒット率を検証する
- ロングフォームコンテンツ: 複数の広告時間枠がある映画やロングエピソードでの広告挿入をテストする
- シリーズコンテンツ: シリーズのエピソード間で一貫した広告ターゲティングを検証する
- さまざまなジャンル: コンテンツのメタデータとジャンルに基づいて広告ターゲティングをテストする

VOD SGAI 最適化のベストプラクティス

SGAI VOD 実装を最適化してパフォーマンスを最大化します。

キャッシュの最適化

- TTL 値を最大化する: コンテンツが変更されないため、VOD マニフェストのキャッシュ期間を長くする
- キャッシュキーの最小化: キャッシュキーのバリエーションを減らしてヒット率を向上させる
- 一般的なコンテンツを事前ウォーミングする: トレンドまたは注目の VOD コンテンツのキャッシュマニフェスト
- キャッシュパフォーマンスのモニタリング: キャッシュヒット率を追跡し、使用パターンに基づいて最適化する

コンテンツ配信の最適化

- 一貫した URL パターン: 予測可能な URL 構造を使用してキャッシュを改善する
- メタデータの標準化: 信頼性の高い広告ターゲティングのために一貫したコンテンツメタデータを確保する
- 広告時間枠の配置: 自然なコンテンツ移行のために広告時間枠の配置を最適化する

- 品質バリエーション: SGAI が VOD コンテンツのすべてのビットレートバリエーションで機能することを確認する

MediaTailor 広告サーバー統合要件

広告サーバーを と統合するには AWS Elemental MediaTailor、広告サーバーが VAST および VMAP のサポートされているバージョンの IAB 仕様に準拠する XML を送信する必要があります。タグが整形形式であることを確認にするには、公開されている VAST バリデータを使用できます。

AWS Elemental MediaTailor は、広告決定サーバーからの VAST および VMAP レスポンスをサポートします。は、クライアント側の広告挿入用のクライアント側のレポート API を介した VPAID メタデータのプロキシ AWS Elemental MediaTailor もサポートしています。クライアント側のレポートについては、「[クライアント側の広告追跡](#)」を参照してください。

MediaTailor は、以下の VAST、VMAP、および VPAID バージョンをサポートしています。

- [VAST 4.3](#) まで

MediaTailor は VAST 4.3 経由のレスポンスバージョンを受け入れますが、VAST 4.0 以降の一部の高度な機能はサポートされていません。

- [VMAP 1.0](#)
- [VPAID 2.0](#)

VAST 要件

広告サーバーの VAST レスポンスには、IAB 準拠の TrackingEvents 要素と、impression のような標準のイベントタイプが含まれている必要があります。標準以外の追跡イベントを含めると、は VAST レスポンス AWS Elemental MediaTailor を拒否し、表示の広告を提供しません。

VAST 3.0 では、一連の順次的なリニア広告の配信である広告ポッドのサポートが導入されました。広告ポッド内の特定の広告が利用できない場合、は ADS のインタラクションログに CloudWatch のエラー AWS Elemental MediaTailor を記録します。その後、ポッド内の次の広告の挿入を試行します。MediaTailor はこのように、使用できる広告が見つかるまでポッド内の広告を繰り返し表示します。

ターゲティング

特定のプレイヤーを広告のターゲットにするには、広告のタグと URL 用のテンプレートを作成できます。詳細については、「[ADS リクエストの MediaTailor 動的広告変数](#)」を参照してください。

AWS Elemental MediaTailor は、広告サーバーの VAST リクエストを送信するとき user-agent、およびサーバー側の追跡呼び出しを行うときに、プレイヤーの および x-forwarded-for ヘッダーをプロキシします。広告サーバーがこれらのヘッダーを処理できることを確認してください。または、[session.user_agent] または [session.client_ip] を使用して、広告タグと広告 URL のクエリ文字列でこれらの値を渡すこともできます。詳細については、「[ADS リクエストの MediaTailor セッション変数](#)」を参照してください。

広告呼び出し

AWS Elemental MediaTailor は、設定で定義されているように VAST 広告 URL を呼び出します。広告呼び出しを実行するときは、プレイヤー固有またはセッション固有のパラメータを置き換えます。MediaTailor は VAST レスポンスで最大 7 レベルの VAST ラッパーとリダイレクトをフォローします。ライブストリーミングのシナリオの場合、MediaTailor は、接続されているプレイヤーの ad avail 開始と同時に広告呼び出しを実行します。実際には、ジッターのため、これらの広告呼び出しには数秒間に渡って行われる場合があります。広告サーバーが、このタイプの呼び出しに必要な同時接続数に対応できることを確認してください。MediaTailor は、ライブワークフローの VAST レスポンスのプリフェッチをサポートしています。詳細については、「[広告のプリフェッチ](#)」を参照してください。

クリエイティブの処理

が ADS VAST レスポンス AWS Elemental MediaTailor を受信すると、クリエイティブごとにトランスコード MediaFile の最高ビットレートを識別し、これをソースとして使用します。このファイルを on-the-fly トランスコーダーに送信して、プレイヤーの多変量プレイリストのビットレートと解像度に適合するレンディションに変換します。最良の結果を得るには、最大ビットレートのメディアファイルが高品質の MP4 アセットであり、そのマニフェストプリセットが有効であることを確認してください。マニフェストプリセットが無効な場合、トランスコードジョブが失敗し、広告は配信されません。無効なプリセットの例には、ProRes などのサポートされていない入力ファイル形式、および解像度 855 x 481 などの特定のレンディション仕様などがあります。

メディアファイル入力でサポートされている形式のリストについては、AWS Elemental MediaConvert 「ユーザーガイド」の「サポートされている入力形式の MP4 行」を参照してください。 <https://docs.aws.amazon.com/mediaconvert/latest/ug/reference-codecs-containers-input.html>

クリエイティブインデックス作成

AWS Elemental MediaTailor は、<Creative>要素で指定されたid属性の値で各クリエイティブを一意にインデックス化します。クリエイティブの ID が指定されていない場合、MediaTailor はインデックスにメディアファイル URL を使用します。

以下の宣言例はクリエイティブ ID を示しています。

```
<Creatives>
  <Creative id="57859154776" sequence="1">
```

独自のクリエイティブ ID を定義する場合は、各クリエイティブに新しい一意の ID を使用してください。クリエイティブ IDsを再利用しないでください。は、繰り返し使用するためにクリエイティブコンテンツ AWS Elemental MediaTailor を保存し、インデックス付き ID でそれぞれを検索します。新しいクリエイティブを受信すると、サービスはまずその ID をインデックスと照合します。ID が存在する場合、MediaTailor は受信コンテンツを再度処理する代わりに、保存されているコンテンツを使用します。クリエイティブ ID が再利用されると、MediaTailor は保存されている古い広告を使用し、新しい広告を再生しません。

広告提供パートナーが提供する VAST 拡張機能

クリエイティブ IDs との衝突を防ぐために、広告提供パートナーが提供する拡張機能を VAST レスポンスに使用できます。MediaTailor は SpringServe、Publica、FreeWheel からの拡張機能をサポートしています。VAST 拡張オーバーライドを有効にすると、MediaTailor はデフォルトのクリエイティブ ID を拡張値に置き換えます。

この機能を有効にするには、[AWS サポートチケットを送信](#)して、VAST 拡張機能ベースのクリエイティブ IDsの有効化をリクエストします。サポートチケットに次の情報を含めます。

- AWS リージョン
- AWS アカウント ID
- MediaTailor 再生設定名

アカウントで VAST 拡張機能ベースのクリエイティブ IDs が有効になっていることを検証するには、ステージングまたはテスト再生設定でRAW_ADS_RESPONSEログ記録を有効にすることもリクエストすることをお勧めします。ログ記録を使用すると、ADS が受信した元の VAST レスポンスを表示し、正しいクリエイティブ IDs が使用されていることを確認することができます。

VPAID 要件

VPAID は、公開者がインタラクティブ性の高い動画広告を配信したり、収益化されたストリームのビューアビリティメトリクスを提供したりすることを可能にします。VPAID の詳細については、「[VPAID 仕様](#)」を参照してください。

AWS Elemental MediaTailor は、server-side-stitched VAST MP4 リニア広告と client-side-inserted VPAID インタラクティブクリエイティブを同じ広告表示で組み合わせます。VAST レスポンス内での表示順序も保持します。MediaTailor は、最大 7 レベルのラッパーを通じて VPAID リダイレクトに従います。クライアント側のレポートのレスポンスには、ラップされていない VPAID メタデータが含まれています。

VPAID を使用するには、以下のガイドラインに従ってください。

- VPAID creatives の MP4 スレートを設定します。は、設定されたスレートで VPAID 広告スロットを AWS Elemental MediaTailor 埋め、クライアントプレーヤーがインタラクティブ広告の実行に使用する VPAID 広告メタデータを提供します。スレートが設定されていない場合、MediaTailor は VPAID 広告が表示されるときに、通常どおりクライアント側のレポートを通じて広告メタデータを渡します。また、欠落しているスレートに関するエラーを CloudWatch にログします。詳細については、「[MediaTailor スレート広告挿入](#)」と「[MediaTailor 再生設定の作成](#)」を参照してください。
- クライアント側のレポートを使用します。は、クライアント側のレポート API を通じて VPAID AWS Elemental MediaTailor をサポートします。詳細については、「[クライアント側の広告追跡](#)」を参照してください。

理論的には、VPAID でデフォルトのサーバー側のレポートモードを使用することが可能です。ただし、サーバー側のレポートを使用する場合は、VPAID 広告の存在に関する情報および関連するメタデータが失われます。それらの情報はクライアント側の API を介してのみ使用可能であるためです。

- ライブシナリオでは、で示される広告表示が EXT-X-CUE-OUT: Duration、VPAID でのユーザーのインタラクティブ性に対応するのに十分な長さであることを確認します。例えば、VAST XML で 30 秒長の VPAID 広告が指定されている場合は、広告表示を 30 秒を超えるように設定することを検討してください。この追加時間により、ユーザーは広告を操作する機会が増えます。時間を追加しないと、広告表示の残りの期間が VPAID 広告に対応するのに十分な長さではないため、VPAID メタデータが失われる可能性があります。

MediaTailor 再生設定の管理

このセクションでは、AWS Elemental MediaTailor 再生設定を管理するための主要なタスクについて説明します。新しい設定を作成してコンテンツストリームを設定し、再生デバイスへのアクセスを提供する方法、既存の設定の詳細を表示する方法、設定を編集してオリジンサーバーや広告決定サーバーなどの設定を更新する方法、不要になった設定を削除する方法について説明します。

トピック

- [MediaTailor 再生設定の作成](#)
- [MediaTailor 設定の詳細の表示](#)
- [MediaTailor 設定の編集](#)
- [MediaTailor 設定の削除](#)

MediaTailor 再生設定の作成

このトピックでは、でコンテンツストリームの受信を開始する設定を作成する方法について説明します AWS Elemental MediaTailor。また、ダウンストリーム再生デバイスがコンテンツをリクエストするためのアクセスポイントを提供する方法も示します。

AWS Elemental MediaTailor コンソール、AWS Command Line Interface (AWS CLI)>、または MediaTailor API を使用して設定を作成できます。AWS CLI または MediaTailor API を使用して設定を作成する方法については、[AWS Elemental MediaTailor API リファレンス](#)を参照してください。

設定を作成するときは、設定名フィールドなどの自由形式のフィールドに機密の識別情報を入れないでください。識別情報には、顧客アカウント番号などが含まれる場合があります。また、MediaTailor コンソール、REST API、または AWS SDKs で作業するときは AWS CLI、識別情報を使用しないでください。MediaTailor に入力されるデータは、いずれも診断ログ、または Amazon CloudWatch Events に取り込まれる可能性があります。

設定を追加する (コンソール)

1. MediaTailor コンソール (<https://console.aws.amazon.com/mediatailor/>) を開きます。
2. [設定] ページで、[Create configuration (設定の作成)] を選択します。
3. 以下のトピックで説明するように、設定フィールドと追加設定フィールドに入力します。
 - [必須の設定](#)
 - [オプションの設定](#)

4. [設定を作成] を選択します。

AWS Elemental MediaTailor は、設定ページの テーブルに新しい設定を表示します。

5. (推奨) マニフェストリクエストとレポートリクエスト AWS Elemental MediaTailor の CDN をでセットアップします。CDN 設定の設定再生 URLs を使用できます。マニフェストとレポートのリクエスト用の CDN の設定については、「[CDN を使用して MediaTailor の広告パーソナライゼーションとコンテンツ配信を最適化する](#)」を参照してください。

必須の設定

設定を作成するときは、次の必須の設定を含める必要があります。

名前

設定を説明する一意の名前を入力します。名前は設定の主な識別子です。許容される最大長は 512 文字です。

コンテンツソース

このストリームのマニフェストの URL プレフィックス (アセット ID を除く) を入力します。最大長は 512 文字です。

たとえば、URL プレフィックス `http://origin-server.com/a/` は の HLS 多変量プレイリスト URL `http://origin-server.com/a/main.m3u8` と の DASH MPD URL に対して有効です `http://origin-server.com/a/dash.mpd`。あるいは、`http://origin-server.com` などの短いプレフィックスを入力できますが、プレイヤーのコンテンツリクエストでアセット ID に `/a/` を含める必要があります。

Note

コンテンツオリジンが HTTPS を使用している場合、その証明書は既知の認証機関からのものであることが必要です。自己署名証明書にすることはできません。自己署名証明書を使用する場合、はコンテンツオリジンへの接続に AWS Elemental MediaTailor 失敗し、プレイヤーのリクエストに応じてマニフェストを提供できません。

広告決定サーバー

広告決定サーバー (ADS) の URL を入力します。これは、「[ステップ 3: ADS リクエスト URL とクエリパラメータを設定する](#)」で説明されている変数を含む URL、またはテスト目的で使っている静的 VAST URL です。最大長は 25,000 文字です。

Note

ADS が HTTPS を使用している場合、その証明書は既知の認証機関からのものである必要があります。自己署名証明書にすることはできません。ADS から返されるメザニン広告の URL についても同様です。自己署名証明書を使用する場合、AWS Elemental MediaTailor はコンテンツオリジンからマニフェストに広告を取得してステッチすることはできません。

オプションの設定

オプションで、MediaTailor コンソール、MediaTailor API、または () で設定エイリアス、パーソナライゼーションの詳細、詳細を設定できますAWS CLI。 MediaTailor AWS Command Line Interface

設定エイリアス

以下は、MediaTailor コンソールまたは MediaTailor API で設定できるオプションの設定エイリアスです。

プレイヤーパラメータ変数

セッションの初期化中に動的ドメイン設定を行うには、1 つ以上のプレイヤーパラメータ変数を追加します。

ドメインを動的に設定するためのプレイヤーパラメータ変数の使用に関する詳細については、「[複数のコンテンツソースの MediaTailor ドメイン変数](#)」を参照してください。

ログ設定

以下は、ログの設定です。

有効な割合

MediaTailor が CloudWatch Logs に書き込む再生設定セッションログの割合を設定します。例えば、再生設定に 1,000 個の広告挿入セッションがあり、有効な割合の値を 60 に設定する場合、MediaTailor は 600 個のセッションログを CloudWatch Logs に書き込みます。

このオプションを有効にすると、MediaTailor は、CloudWatch Logs アカウントでのセッションログの書き込みと管理を MediaTailor に許可するサービスリンクロールを自動的に作成します。詳細については、「[MediaTailor 向けのサービスリンクロールの使用](#)」を参照してください。

ログ記録戦略

MediaTailor が出力するログの収集に使用される方法を示します。CloudWatch Logs に直接ログを送信するには、`LEGACY_CLOUDWATCH` を選択します。CloudWatch Logs にログを送信するには、`VENDED_LOGS` を選択します。サポートされている送信先は、CloudWatch Logs ロググループ、Amazon S3 バケット、Amazon Data Firehose ストリームです。

提供されたログには、追加のセットアップが必要です。セットアップについては、「」を参照してください。[販売ログの使用](#)。

ADS インタラクションログのオプトインイベント

MediaTailor が、この設定で初期化されたセッションの `RAW_ADS_RESPONSE` ログを出力することを示します。

`RAW_ADS_RESPONSE` ログイベントには、ADS からの VAST または VMAP レスポンス全体が含まれます。そのため、ログは広範囲に及ぶ可能性があり、ログ記録のコストが増加する可能性があります。

ADS インタラクションログ除外イベント

MediaTailor が ADS とのやり取りを記述するログで選択したイベントを出力しないことを示します。

ADS ログイベントの説明については、「」を参照してください。[ADS ログ](#)。

マニフェストサービスインタラクションログ除外イベント

MediaTailor がマニフェストサービスとのインタラクションを記述する選択したイベントをログに出力しないことを示します。

マニフェストサービスログイベントの説明については、「」を参照してください。[マニフェストログ](#)。

広告のコンディショニング

以下はMediaTailor がコンテンツストリームに広告をステッチする前に条件付けするために実行するアクションを決定します。

ストリーミングメディアファイルのコンディショニング

MediaTailor がどの広告をステッチするかを決定する際に使用するロジックを決定します。

- ストリーミングメディアファイルのコンディショニングが Transcode に設定されている場合、MediaTailor はメディアファイルをprogressive配信でトランスコードし、マニフェストにステッチします。progressive 配信メディアファイルを含む十分な広告がない場合、MediaTailor はそれらをトランスコードしてstreaming配信に使用します。
- ストリーミングメディアファイルのコンディショニングが None に設定されている場合、MediaTailor はstreaming、配信メディアファイルをトランスコードせずにマニフェストに広告をステッチします。streaming 配信メディアファイルを含む十分な広告がない場合、MediaTailor はそれらをトランスコードしてprogressive配信に使用します。

パーソナライゼーションの詳細

以下は、MediaTailor コンソールまたは MediaTailor API で設定できるパーソナライゼーションの詳細です。

スレート広告

MediaTailor が未入力の広告時間を埋めるために使用する高品質の MP4 アセットの URL を入力します。スレート設定はほとんどのワークフローではオプションですが、VPAID 広告では必須です。スレートの動作、設定要件、スレートが表示されるタイミングに関する包括的な情報については、「」を参照してください[MediaTailor スレート広告挿入](#)。

スタートバンパー

スタートバンパーアセットの場所の URL です。バンパーは、広告ブレークの開始時または終了前に再生される短時間の動画クリップまたはオーディオクリップです。これらは Amazon の S3、または別のストレージサービスに保存できます。バンパーの詳細については、「[MediaTailor バンパー広告挿入](#)」を参照してください。

エンドバンパー

エンドバンパーアセットの場所の URL です。バンパーは、広告ブレークの開始時または終了前に再生される短時間の動画クリップまたはオーディオクリップです。これらは Amazon の S3、

または別のストレージサービスに保存できます。バンパーの詳細については、「[MediaTailor バンパー広告挿入](#)」を参照してください。

パーソナライズしきい値

スレート広告と組み合わせて使用する場合にのみ適用されます。MediaTailor がパーソナライゼーションを中止し、基盤となるコンテンツを表示するまでに、広告時間枠で許容される不足広告時間 (秒単位) の最大期間を定義します。この機能は基盤となるコンテンツストリームに依存するため、広告挿入ではなく、ライブストリームと VOD ストリームでの広告置換に適用されます。詳細な動作シナリオと包括的な情報については、「」を参照してください [パーソナライゼーションしきい値の仕組み](#)。

ライブプレロール広告決定サーバー

メインコンテンツの再生開始前の、ライブストリームの開始時点に広告を挿入するには、広告決定サーバー (ADS) からの広告プレロールの URL を入力します。これは、「[ステップ 3: ADS リクエスト URL とクエリパラメータを設定する](#)」で説明されている変数を含む URL、またはテスト目的で使用している静的 VAST URL です。最大長は 25,000 文字です。

Note

ADS が HTTPS を使用している場合、その証明書は既知の認証機関からのものであることが必要です。自己署名証明書にすることはできません。ADS から返されるメザニン広告の URL についても同様です。自己署名証明書を使用する場合、AWS Elemental MediaTailor はコンテンツオリジンからマニフェストに広告を取得してステッチすることはできません。

プレロールの仕組みについては、「[MediaTailor プリロール広告挿入](#)」を参照してください。

ライブプレロールの最大許容時間

ライブストリームの開始時点に広告を挿入している場合は、プレロール ad avail の最大許容継続時間を入力します。MediaTailor は、広告の挿入時にこの時間を超過しません。ADS からのレスポンスにこの継続時間に収まらない広告が含まれている場合、MediaTailor は継続時間を超過せずに、可能な限り多くの広告で avail を埋めます。MediaTailor が avail を埋める方法の詳細については、「[ライブ広告ステッチング動作](#)」を参照してください。

Avail 抑制モード

モードを avail 抑制 (広告抑制とも呼ばれます) に設定します。デフォルトでは、広告抑制はオフになっており、MediaTailor はすべての広告またはスレートを埋めます。モードを に設定する

とBEHIND_LIVE_EDGE、広告抑制はアクティブになり、MediaTailor はマニフェストルックバックウィンドウの有効抑制値時刻以降、広告ブレイクを埋めません。モードが に設定されている場合AFTER_LIVE_EDGE、広告抑制はアクティブになります。MediaTailor は、有効抑制期間の前後の広告時間枠を埋めません。これは、ライブエッジに有効抑制値とバッファ時間を加えたものです。

Avail 抑制値

Avail 抑制値は、HH:MM:SS 形式のライブエッジオフセット時刻です。MediaTailor は、マニフェストルックバックウィンドウのこの時刻以前の広告ブレイクを埋めません。

挿入モード

Insertion Mode は、プレイヤーがステッチド広告挿入とガイド付き広告挿入のどちらを使用できるかを制御します。デフォルトのはSTITCHED_ONLY、すべてのプレイヤーセッションにステッチド (サーバー側) 広告挿入を使用するように強制します。InsertionMode を に設定するPLAYER_SELECTと、プレイヤーはセッション初期化時にステッチまたはガイド付き広告挿入を選択できます。挿入モードを指定しないプレイヤーのデフォルトはステッチされます。

詳細設定

以下は、オプションの設定の詳細です。これらは、MediaTailor コンソール、AWS Command Line Interface (AWS CLI)、または MediaTailor API を使用して設定できます。

CDN コンテンツセグメントプレフィックス

AWS Elemental MediaTailor コンテンツセグメントの CDN パスへの URLsを使用してマニフェストを作成できるようにします。この手順を実行する前に、CDN でオリジンサーバーからセグメントをプルするためのルールを設定します。[CDN content segment prefix (CDN コンテンツセグメントプレフィックス)] に CDN プレフィックスパスを入力します。

MediaTailor と CDN の統合の詳細については、「[CDN を使用して MediaTailor の広告パーソナライゼーションとコンテンツ配信を最適化する](#)」を参照してください。

CDN 広告セグメントプレフィックス

AWS Elemental MediaTailor が広告セグメントの独自の CDN パスへの URLsを持つマニフェストを作成できるようにします。デフォルトで、MediaTailor はデフォルトのキャッシュ設定を使用して、内部 Amazon CloudFront デイストリビューションからの広告セグメントを提供します。[CDN ad segment prefix (CDN 広告セグメントプレフィックス)] フィールドに入力する前に、CDN で以下のオリジンから広告セグメントをプルするためのルールを設定する必要があります。以下に例を示します。

```
https://segments.mediatailor.<region>.amazonaws.com
```

[CDN ad segment prefix (CDN 広告セグメントプレフィックス)] に、設定の CDN プレフィックスの名前を入力します。

MediaTailor と CDN の統合の詳細については、「[CDN を使用して MediaTailor の広告パーソナライゼーションとコンテンツ配信を最適化する](#)」を参照してください。

DASH オリジンマニフェストタイプ

オリジンサーバーがシングルピリオド DASH マニフェストを生成する場合は、ドロップダウンリストを開き、[SINGLE_PERIOD] を選択します。デフォルトで、MediaTailor は DASH マニフェストをマルチピリオドマニフェストとして処理します。詳細については、「[the section called “MPEG-DASH ソースの統合”](#)」を参照してください。

DASH mpd の場所

(DASH の必要に応じてオプション) メディアプレゼンテーションの説明 (mpd) の場所。次の状況では DISABLED を選択します。

- MediaTailor マニフェストにアクセスするための CDN ルーティングルールを設定します。
- クライアント側のレポートを使用するか、プレイヤーがスティッキー HTTP リダイレクトをサポートします。

[Location] (ロケーション) 機能の詳細については、「[the section called “ロケーション機能”](#)」を参照してください。

トランスコードプロファイル名

この設定をカスタムトランスコードプロファイルに関連付ける名前です。この名前は、MediaTailor の動的トランスコードのデフォルトを上書きします。このフィールドは、AWS Support の協力を得てカスタムプロファイルを既にセットアップしている場合にのみ入力します。

広告マーカースパススルー

HLS に対して、広告マーカースパススルーを有効または無効にします。広告マーカースパススルーが有効になっている場合、MediaTailor はオリジンマニフェストからの EXT-X-CUE-IN、EXT-X-CUE-OUT、および EXT-X-SPLICEPOINT-SCTE35 広告マーカースを MediaTailor のパーソナライズされたマニフェストにパススルーします。広告マーカースの値にロジックは適用されません。これらは、オリジンマニフェストからパーソナライズされたマニフェストにそのまま

渡されます。例えば、オリジンマニフェストの EXT-X-CUE-OUT の値が 60 になっているのに広告が配置されていないという場合でも、MediaTailor はパーソナライズされたマニフェストの値を 0 に変更しません。

MediaTailor 設定の詳細の表示

設定を作成したときに指定した値に加えて、AWS Elemental MediaTailor では設定の名前、再生エンドポイント、関連するアクセス URL が表示されます。設定を表示するには、以下の手順を実行します。

設定を表示する

1. MediaTailor コンソール (<https://console.aws.amazon.com/mediatailor/>) を開きます。
2. [Configurations] (設定) ページで、表示する設定の [Configuration name] (設定名) を選択します。

MediaTailor 設定の編集

設定を編集して、オリジンサーバーと広告決定サーバー (ADS) マッピングを更新するか、 がコンテンツ配信ネットワーク (CDN) と AWS Elemental MediaTailor やり取りする方法を変更できます。

設定を編集する

1. MediaTailor コンソール (<https://console.aws.amazon.com/mediatailor/>) を開きます。
2. [設定] ページで、編集する設定の名前を選択します。
3. 設定の詳細ページで [Edit (編集)] を選択し、必要に応じて設定の内容を見直します。設定名を編集することはできません。設定の属性については、「[MediaTailor 再生設定の作成](#)」を参照してください。
4. [保存] を選択します。

MediaTailor 設定の削除

設定を削除して、 で再生できないようにすることができます AWS Elemental MediaTailor。

設定を削除する

1. MediaTailor コンソール (<https://console.aws.amazon.com/mediatailor/>) を開きます。
2. [設定] ページで、以下のいずれかを実行します。
 - 削除する設定の名前を選択します。
 - [設定名] 列で、名前の横にあるオプションを選択してから [Delete (削除)] を選択します。
3. [Delete (削除)] 確認ボックスに「Delete」と入力し、[Delete (削除)] を選択します。

MediaTailor 広告挿入用のコンテンツソースの統合

このトピックでは、さまざまなタイプの動画コンテンツソースと MediaTailor の統合について説明します。MediaTailor は、ライブコンテンツとオンデマンドコンテンツの HLS ストリーミングプロトコルと DASH ストリーミングプロトコルの両方をサポートしています。このサービスは、指定された広告時間枠中に広告の挿入または置換を実行することができ、これらの機能を有効にするために入力ビデオマニフェストの構造とフォーマットに関する特定の要件があります。以下のトピックでは、HLS および DASH コンテンツを MediaTailor と統合してパーソナライズされた広告エクスペリエンスを実現するための入力ソースの要件と手順について詳しく説明します。

トピック

- [MediaTailor 広告挿入の入力ソース要件](#)
- [HLS ソースの統合](#)
- [MPEG-DASH ソースの統合](#)
- [SigV4 との AWS Elemental MediaTailor オリジンインタラクションの保護](#)

MediaTailor 広告挿入の入力ソース要件

MediaTailor で入力ソースを機能させるには、ソースが以下の要件を満たしている必要があります。

- Apple HLS (HTTP Live Streaming) または MPEG DASH (Dynamic Adaptive Streaming over HTTP) を使用する
- ライブストリーミングまたはビデオオンデマンド (VOD) を使用する
- パブリックインターネットでのアクセスが可能であり、パブリック IP アドレスがある
- 標準の HTTP ポート (ポート 80) または HTTPS ポート (ポート 443) を使用します。MediaTailor は、オリジンサーバー通信用のカスタムポートをサポートしていません。
- [MediaTailor 広告挿入入門チュートリアル](#)で説明されているいずれかの形式で広告マーカースを含める

HLS ソースの統合

AWS Elemental MediaTailor は、ライブストリーミングとビデオオンデマンド (VOD) の EXT-X-VERSION3以上の .m3u8 HLS マニフェストをサポートします。MediaTailor が広告ブレイクに到達すると、コンテンツのタイプに基づいて広告の挿入または置き換えを試みます。その継続時間を埋めるために十分な広告がない場合、MediaTailor は基盤となるコンテンツストリーム、または設定されたスレートを表示します。コンテンツのタイプに基づいた HLS 広告動作の詳細については、「[AWS Elemental MediaTailor 広告挿入動作について](#)」を参照してください。

以下のセクションでは、Mediatailor が HLS マニフェストを処理する方法について詳しく説明します。

トピック

- [HLS 対応の広告マーカ](#)
- [広告マーカパススルーの有効化](#)
- [HLS マニフェストタグの処理](#)
- [HLS マニフェストの例](#)

HLS 対応の広告マーカ

AWS Elemental MediaTailor は、サポートされている広告マーカの入力マニフェストを解析することで、HLS マニフェスト内の広告表示境界を識別します。以下のセクションでは、MediaTailor が使用するマーカについて説明します。

EXT-X-ASSET

EXT-X-ASSET タグには、視聴者のコンテンツをパーソナライズするために広告決定サーバー (ADS) により使用されるメタデータが含まれています。EXT-X-ASSET パラメータは、カンマで区切られたキーと値のペアです。

このタグを使用するには、以下の要件を満たしている必要があります。

- オリジンマニフェストの EXT-X-ASSET 値を URL エンコードする必要があります。次の例は、キーと URL エンコードされた値を持つ EXT-X-ASSET タグを示しています。

```
#EXT-X-ASSET:GENRE=CV,CAID=12345678,EPIISODE="Episode%20Name%20Date",SEASON="Season%20Name%20and%20Number",SERIES="Series%2520Name"
```

- MediaTailor ADS 設定には、動的 [asset.] 変数とキーを含める必要があります。以下は、動的 [asset.] 変数とキーを使用した MediaTailor ADS 設定の例です。

```
https://myads.com/stub?  
c=[asset.GENRE]&g=[asset.CAID]&e=[asset.EPISODE]&s=[asset.SEASON]&k=[asset.SERIES]
```

VAST リクエストの例

次の例は、ADS に対する VAST GET リクエストを示しています。

```
https://myads.com/stub?c=CV&g=12345678&e=Episode%20Name%20Date&s=Season  
%20Name%20and%20Number&k=Series%2520Name
```

EXT-X-CUE-OUT および EXT-X-CUE-IN

このタイプの広告マーカースは最も一般的です。次の例は、このようなキューマーカのオプションを示しています。

```
#EXT-X-CUE-OUT:DURATION=120  
...  
#EXT-X-CUE-IN
```

```
#EXT-X-CUE-OUT:30.000  
...  
#EXT-X-CUE-IN
```

```
#EXT-X-CUE-OUT  
...  
#EXT-X-CUE-IN
```

EXT-X-DATERANGE

EXT-X-DATERANGE 広告マーカータグでは、SCTE35-OUT 属性を使用して、ad avail のタイミングを指定します。

Note

AWS Elemental MediaTailor は、EXT-X-DATERANGE 広告マーカースに指定された START-DATE 属性を無視します。

Ad avail は、次のいずれかの方法で指定できます。

- SCTE35-OUT および DURATION 仕様の EXT-X-DATERANGE タグ。

例

```
#EXT-X-DATERANGE:ID="splice-6FFFFFF0",START-DATE="2019-01T00:15:00Z\n",DURATION=60.000,SCTE35-OUT=0xF
```

- ペアリングされた EXT-X-DATERANGE タグ (SCTE35-OUT 仕様と SCTE35-IN 仕様)。

例

```
#EXT-X-DATERANGE:ID="splice-6FFFFFF0",START-DATE="2019-01T00:15:00Z\n",SCTE35-OUT=0xF
...
#EXT-X-DATERANGE:ID="splice-6FFFFFF0",START-DATE="2019-01T00:15:00Z\n",SCTE35-IN=0xF
```

- 以前のオプションの組み合わせ。仕様が SCTE35-OUT と DURATION の EXT-X-DATERANGE タグ、その後に SCTE35-IN 仕様の EXT-X-DATERANGE タグが続きます。この場合、MediaTailor は 2 つの仕様のうち最も早いキューイン設定を使用します。

例

```
#EXT-X-DATERANGE:ID="splice-6FFFFFF0",START-DATE="2019-01T00:15:00Z\n",DURATION=60.000,SCTE35-OUT=0xF
...
#EXT-X-DATERANGE:ID="splice-6FFFFFF0",START-DATE="2019-01T00:15:00Z\n",SCTE35-IN=0xF
```

EXT-X-SPLICEPOINT-SCTE35

SCTE-35 ペイロードを含む EXT-X-SPLICEPOINT-SCTE35 広告マーカートグを base64 エンコードのバイナリで追加します。デコードされたバイナリで、プロバイダー配置機会を開始するキューアウトマーカース 0x34 と、プロバイダー配置機会を終了するキューインマーカース 0x35 を含む SCTE-35 splice_info_section を示す必要があります。

次の例は、キューアウトマーカースとキューインマーカースを指定する base64 エンコードのバイナリペイロードを含むスプライスポイントの仕様を示しています。

```
#EXT-X-SPLICEPOINT-SCTE35:/DA9AAAAAAAAAAP/wBQb+uYbZqwAnAiVDVUVJAAAKqX//  
AAEjW4AMEU1EU05CMDAxMTMyMjE5M190NAAAmXz5JA==  
...  
#EXT-X-SPLICEPOINT-SCTE35:/DA4AAAAAAAAAAP/wBQb+tTaaAwAiAiBDVUVJAAAKqH+/  
DBFNRFNOQjAwMTEzMjIxOTJfTjUAAIiGK1s=
```

広告マーカースパススルーの有効化

HLS の場合、デフォルトで MediaTailor のパーソナライズされたマニフェストにオリジンマニフェストからの SCTE-35 広告マーカースが含まれません。広告マーカースのパススルーが有効になっていると、MediaTailor がオリジンマニフェストからの以下の広告マーカースをパーソナライズされたマニフェストにパススルーします。

- EXT-X-CUE-IN
- EXT-X-CUE-IN
- EXT-X-SPLICEPOINT-SCTE35

広告マーカースのパススルーはオプションの設定です。広告マーカースのパススルーは、MediaTailor のパーソナライズされたマニフェストに SCTE 広告マーカースを含めたい場合に使用します。一般的なユースケースには以下が含まれます。

- コンテンツ置換 - コンテンツの置き換え、またはコンテンツの制限を実行します。
- 広告追跡 - 1 つ、または複数の広告マーカースの存在または欠如に基づいて、広告追跡情報をトリガーします。
- プレイヤー設定 - 広告マーカースの存在または欠如に基づいて、プレイヤーの UI でスクラブ機能またはカウントダウンタイマー機能を有効にします。

 Note

MediaTailor は、これらのマーカの値を変更しません。例えば、オリジンマニフェストの EXT-X-CUE-OUT の値が 60 になっているのに広告が配置されていないという場合でも、MediaTailor はパーソナライズされたマニフェストの値を 0 に変更しません。

広告マーカのパススルーを有効にする

広告マーカのパススルーは、AWS マネジメントコンソール または AWS Command Line Interface () を使用して有効にできますAWS CLI。

コンソールを使用して広告マーカのパススルーを有効にする

1. MediaTailor コンソール (<https://console.aws.amazon.com/mediatailor/>) を開きます。
2. [New Configuration] (新しい設定) または [Edit Configuration] (設定を編集) をクリックします。
3. [Advanced Settings] (アドバンスド設定) セクションで、ドロップダウンメニューから [Enable] (有効) を選択します。

AWS Command Line Interface (AWS CLI) を使用して広告マーカのパススルーを有効にするには

[put-playback-configuration](#) コマンドを使用します。

HLS マニフェストタグの処理

このセクションでは、がパーソナライズされた出力マニフェストでタグ AWS Elemental MediaTailor を管理する方法について説明します。

EXT-X-CUE タグ

MediaTailor は、入力マニフェスト内の EXT-X-CUE-OUT、EXT-X-CUE-OUT-CONT、および EXT-X-CUE-IN タグを、出力マニフェスト内の EXT-X-DISCONTINUITY タグに置き換えます。DISCONTINUITY タグは次の境界をマークします。

- メインコンテンツが広告に移行する場所
- ある広告から別の広告に遷移する場所
- 広告がメインコンテンツに戻る場所

EXT-X-DATERANGE タグ

MediaTailor は、入カマニフェストからの EXT-X-DATERANGE タグを出カマニフェストにパススルーします。MediaTailor は、DATERANGE タグに対応する EXT-X-DISCONTINUITY タグも挿入します。DISCONTINUITY タグは次の境界をマークします。

- メインコンテンツが広告に移行する場所
- ある広告から別の広告に遷移する場所
- 広告がメインコンテンツに戻る場所

EXT-X-KEY タグ

MediaTailor は、入カマニフェストからの EXT-X-KEY タグをパススルーします。これらのタグは、メインコンテンツが暗号化されていることを示します。広告は無暗号化されているため、MediaTailor は ad avail の開始時点で EXT-X-KEY:METHOD=NONE を挿入します。再生がメインコンテンツに戻ると、MediaTailor は、暗号化タイプとして定義された METHOD 値と共に EXT-X-KEY タグを挿入して暗号化を再度有効にします。

認識されないタグ

MediaTailor は、入カマニフェストからのすべての不明なタグとカスタムタグを出カマニフェストにパススルーします。

HLS マニフェストの例

以下のセクションでは、HLS オリジンマニフェストとパーソナライズされたマニフェストの例を示します。これらの例を理解することは、MediaTailor ワークフローの設定とトラブルシューティングに役立ちます。

クエリパラメータを HLS マニフェストとセグメントに適用する方法については、「」を参照してください [MediaTailor HLS 暗黙的なセッションの初期化](#)。

HLS プレイリストタイプについて

HTTP Live Streaming (HLS) は、主に 2 種類のプレイリストを使用します。

マルチバリエーションプレイリスト

多変数プレイリストは、コンテンツの使用可能なすべてのレンディションを一覧表示する最上位インデックスファイルです。メディアプレイリストへの参照が含まれていますが、メディアセグ

メント自体は含まれません。このプレイリストにより、プレイヤーはネットワーク条件、デバイス機能、またはユーザー設定に基づいて最適なレンディションを選択できます。

このプレイリストタイプは、マスタープレイリスト、マスターマニフェスト、プライマリプレイリスト、メインプレイリスト、インデックスファイル、マスター M3U8 など、さまざまなコンテンツの他のいくつかの名前でも知られています。

MediaTailor ワークフローでは、多変量プレイリストは再生リクエストのエントリーポイントであり、広告パーソナライゼーションが開始される場所です。

メディアプレイリスト

メディアプレイリストには、コンテンツの特定のレンディション (品質レベル) の実際のメディアセグメント情報が含まれます。これには、1 つのレンディションの再生に必要なタイミング情報、セグメント URLs、およびその他のメタデータが含まれます。

このプレイリストタイプは、メディアプレイリスト、子マニフェスト、チャンクリスト、メディア M3U8、レンディションプレイリストとも呼ばれます。

MediaTailor ワークフローでは、メディアプレイリストは、コンテンツセグメントと広告セグメントの両方を適切なシーケンスに含めるようにパーソナライズされます。

HLS プレイリストタイプの詳細については、「」を参照してください[HLS プレイリストタイプ](#)。

HLS オリジンマニフェストの例

次の例は、HLS が AWS Elemental MediaTailor コンテンツオリジンから受信した HLS マルチバリエーションプレイリストを示しています。

```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-INDEPENDENT-SEGMENTS
#EXT-X-STREAM-INF:BANDWIDTH=2665726,AVERAGE-
BANDWIDTH=2526299,RESOLUTION=960x540,FRAME-
RATE=29.970,CODECS="avc1.640029,mp4a.40.2",SUBTITLES="subtitles"
index_1.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=3956044,AVERAGE-
BANDWIDTH=3736264,RESOLUTION=1280x720,FRAME-
RATE=29.970,CODECS="avc1.640029,mp4a.40.2",SUBTITLES="subtitles"
index_2.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=995315,AVERAGE-
BANDWIDTH=951107,RESOLUTION=640x360,FRAME-
RATE=29.970,CODECS="avc1.4D401E,mp4a.40.2",SUBTITLES="subtitles"
```



```
index_3.m3u8
#EXT-X-MEDIA:TYPE=SUBTITLES,GROUP-
ID="subtitles",NAME="caption_1",DEFAULT=YES,AUTOSELECT=YES,FORCED=NO,LANGUAGE="eng",URI="index_
```

この多変量プレイリストの例:

- #EXT-X-STREAM-INF タグは、解像度とビットレートが異なるさまざまなレンディションを定義します。
- 各レンディションはメディアプレイリストを参照します (などindex_1.m3u8)
- #EXT-X-MEDIA タグは字幕トラックを定義します

次の例は、コンテンツオリジンから HLS が AWS Elemental MediaTailor 受信した HLS メディアプレイリストを示しています。この例では、EXT-X-CUE-OUT タグと EXT-X-CUE-IN タグを使用して、ad avail 機会を表しています。

```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-TARGETDURATION:7
#EXT-X-MEDIA-SEQUENCE:8779957
#EXTINF:6.006,
index_1_8779957.ts?m=1566416212
#EXTINF:6.006,
index_1_8779958.ts?m=1566416212
#EXTINF:5.372,
index_1_8779959.ts?m=1566416212
#EXT-OATCLS-SCTE35:/DA1AAAAAsvhAP/wFAXwAAAGf+AdLfiP4AG3dAAAEBQAAXytxmQ==
#EXT-X-CUE-OUT:20.020
#EXTINF:0.634,
index_1_8779960.ts?m=1566416212
#EXT-X-CUE-OUT-CONT:ElapsedTime=0.634,Duration=21,SCTE35=/DA1AAAAAsvhAP/wFAXwAAAGf
+AdLfiP4AG3dAAAEBQAAXytxmQ==
#EXTINF:6.006,
index_1_8779961.ts?m=1566416212
#EXT-X-CUE-OUT-CONT:ElapsedTime=6.640,Duration=21,SCTE35=/DA1AAAAAsvhAP/wFAXwAAAGf
+AdLfiP4AG3dAAAEBQAAXytxmQ==
#EXTINF:6.006,
index_1_8779962.ts?m=1566416212
#EXT-X-CUE-OUT-CONT:ElapsedTime=12.646,Duration=21,SCTE35=/DA1AAAAAsvhAP/wFAXwAAAGf
+AdLfiP4AG3dAAAEBQAAXytxmQ==
#EXTINF:6.006,
index_1_8779963.ts?m=1566416212
```

```
#EXT-X-CUE-OUT-CONT:ElapsedTime=18.652,Duration=21,SCTE35=/DA1AAAAAsvhAP/wFAXwAAAGf
+ /+AdLfIP4AG3dAAAEBQAAXytxmQ==
#EXTINF:1.368,
index_1_8779964.ts?m=1566416212
#EXT-X-CUE-IN
#EXTINF:4.638,
index_1_8779965.ts?m=1566416212
#EXTINF:6.006,
index_1_8779966.ts?m=1566416212
#EXTINF:6.006,
index_1_8779967.ts?m=1566416212
#EXTINF:6.006,
index_1_8779968.ts?m=1566416212
```

このメディアプレイリストの例:

- #EXTINF タグは各セグメントの期間を指定します
- #EXT-X-CUE-OUT タグは広告時間枠の開始を示します
- #EXT-X-CUE-OUT-CONT タグは、進行中の広告時間枠に関する情報を提供します。
- #EXT-X-CUE-IN タグは広告時間枠の終了を示します

HLS のパーソナライズされたマニフェストの例

次の例は、パーソナライズされた AWS Elemental MediaTailor HLS マルチバリエーションプレイリストを示しています。

```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-MEDIA:LANGUAGE="eng",AUTOSELECT=YES,FORCED=NO,TYPE=SUBTITLES,URI=" ../../../../
manifest/43f3e412052f2808dd84ea1da90e92e914edddde/external-
canary-hls/ee1696a8-4f7f-4c4c-99de-9821131847e8/3.m3u8",GROUP-
ID="subtitles",DEFAULT=YES,NAME="caption_1"
#EXT-X-INDEPENDENT-SEGMENTS
#EXT-X-STREAM-INF:CODECS="avc1.640029,mp4a.40.2",AVERAGE-
BANDWIDTH=2526299,RESOLUTION=960x540,SUBTITLES="subtitles",FRAME-
RATE=29.97,BANDWIDTH=2665726
../../../../manifest/43f3e412052f2808dd84ea1da90e92e914edddde/external-canary-hls/
ee1696a8-4f7f-4c4c-99de-9821131847e8/0.m3u8
#EXT-X-STREAM-INF:CODECS="avc1.640029,mp4a.40.2",AVERAGE-
BANDWIDTH=3736264,RESOLUTION=1280x720,SUBTITLES="subtitles",FRAME-
RATE=29.97,BANDWIDTH=3956044
```

```

../.../manifest/43f3e412052f2808dd84ea1da90e92e914edddde/external-canary-hls/
ee1696a8-4f7f-4c4c-99de-9821131847e8/1.m3u8
#EXT-X-STREAM-INF:CODECS="avc1.4D401E,mp4a.40.2",AVERAGE-
BANDWIDTH=951107,RESOLUTION=640x360,SUBTITLES="subtitles",FRAME-
RATE=29.97,BANDWIDTH=995315
../.../manifest/43f3e412052f2808dd84ea1da90e92e914edddde/external-canary-hls/
ee1696a8-4f7f-4c4c-99de-9821131847e8/2.m3u8

```

MediaTailor がメディアプレイリスト URLs を変更して、パーソナライズされた広告挿入を可能にするセッション固有の情報を含める方法に注目してください。

次の例は、パーソナライズされた AWS Elemental MediaTailor メディアプレイリストを示しています。

```

#EXTM3U
#EXT-X-VERSION:6
#EXT-X-TARGETDURATION:7
#EXT-X-MEDIA-SEQUENCE:8779957
#EXT-X-DISCONTINUITY-SEQUENCE:0
#EXTINF:6.006,
https://777788889999.mediapackage.us-west-2.amazonaws.com/out/v1/
e309fffd02ba8498d864dcaacff7a5ad9/index_1_8779957.ts?m=1566416212
#EXTINF:6.006,
https://777788889999.mediapackage.us-west-2.amazonaws.com/out/v1/
e309fffd02ba8498d864dcaacff7a5ad9/index_1_8779958.ts?m=1566416212
#EXTINF:5.372,
https://777788889999.mediapackage.us-west-2.amazonaws.com/out/v1/
e309fffd02ba8498d864dcaacff7a5ad9/index_1_8779959.ts?m=1566416212
#EXT-X-DISCONTINUITY
#EXTINF:3.066667,
../.../.../segment/43f3e412052f2808dd84ea1da90e92e914edddde/external-canary-hls/
ee1696a8-4f7f-4c4c-99de-9821131847e8/0/8779960
#EXTINF:3.0,
../.../.../segment/43f3e412052f2808dd84ea1da90e92e914edddde/external-canary-hls/
ee1696a8-4f7f-4c4c-99de-9821131847e8/0/8779961
#EXTINF:3.0,
../.../.../segment/43f3e412052f2808dd84ea1da90e92e914edddde/external-canary-hls/
ee1696a8-4f7f-4c4c-99de-9821131847e8/0/8779962
#EXTINF:3.0,
../.../.../segment/43f3e412052f2808dd84ea1da90e92e914edddde/external-canary-hls/
ee1696a8-4f7f-4c4c-99de-9821131847e8/0/8779963
#EXTINF:2.966667,

```

```

.././../././segment/43f3e412052f2808dd84ea1da90e92e914edddde/external-canary-hls/
ee1696a8-4f7f-4c4c-99de-9821131847e8/0/8779964
#EXT-X-DISCONTINUITY
#EXTINF:6.006,
https://777788889999.mediapackage.us-west-2.amazonaws.com/out/v1/
e309ffd02ba8498d864dcaacff7a5ad9/index_1_8779963.ts?m=1566416212
#EXTINF:1.368,
https://777788889999.mediapackage.us-west-2.amazonaws.com/out/v1/
e309ffd02ba8498d864dcaacff7a5ad9/index_1_8779964.ts?m=1566416212
#EXTINF:4.638,
https://777788889999.mediapackage.us-west-2.amazonaws.com/out/v1/
e309ffd02ba8498d864dcaacff7a5ad9/index_1_8779965.ts?m=1566416212
#EXTINF:6.006,
https://777788889999.mediapackage.us-west-2.amazonaws.com/out/v1/
e309ffd02ba8498d864dcaacff7a5ad9/index_1_8779966.ts?m=1566416212
#EXTINF:6.006,
https://777788889999.mediapackage.us-west-2.amazonaws.com/out/v1/
e309ffd02ba8498d864dcaacff7a5ad9/index_1_8779967.ts?m=1566416212
#EXTINF:6.006,
https://777788889999.mediapackage.us-west-2.amazonaws.com/out/v1/
e309ffd02ba8498d864dcaacff7a5ad9/index_1_8779968.ts?m=1566416212

```

このパーソナライズされたメディアプレイリストの例:

- MediaTailor がコンテンツセグメント間に広告セグメントを挿入しました
- #EXT-X-DISCONTINUITY タグはコンテンツと広告間の遷移をマークします
- コンテンツセグメントはオリジンサーバーから提供されます (この例では MediaPackage)
- 広告セグメントは MediaTailor の広告セグメントストレージから提供されます

パーソナライズされたマニフェストの主な違い

MediaTailor が HLS マニフェストをパーソナライズすると、いくつかの重要な変更が行われます。

多変量プレイリストの変更

- バリエーションプレイリスト URLsは、セッション情報を含む MediaTailor が管理する URLsられます。
- タグの順序は、最適な再生のために再編成できます。

バリエーションプレイリストの変更

- 広告マーカー (EXT-X-CUE-OUT、EXT-X-CUE-IN) は実際の広告セグメントに置き換えられます
- コンテンツ/広告の境界に不連続マーカー (EXT-X-DISCONTINUITY) が追加される
- コンテンツセグメント URLsがオリジンまたは CDN を指すように書き換えられる
- MediaTailor URLs広告セグメントストレージを指すように広告セグメント URL が追加されます

これらの変更を理解することで、MediaTailor ワークフローの問題をトラブルシューティングし、CDN とプレイヤーを適切に設定することができます。

関連トピック

HLS マニフェストと MediaTailor の詳細については、以下のトピックを参照してください。

- [HLS プレイリストタイプ](#) - HLS プレイリストタイプの詳細な説明
- [CDN を使用して MediaTailor の広告パーソナライゼーションとコンテンツ配信を最適化する](#) - MediaTailor での CDN の使用に関する情報
- [MediaTailor 広告挿入の仕組み](#) - MediaTailor 広告挿入の仕組みの概要

MPEG-DASH ソースの統合

AWS Elemental MediaTailor は、DASH 動的プロファイルのガイドラインに従った .mpd ライブおよびビデオオンデマンド (VOD) マニフェストをサポートしています。MediaTailor は、マルチピリオドおよびシングルピリオドの DASH 準拠のマニフェスト入力を受け入れ、マルチピリオドの DASH 準拠のマニフェスト出力を配信します。

入力マニフェストには次のものがが必要です。

- splice insert または time signal のスプライス情報設定を含む SCTE-35 イベントストリーム。設定は、平文の XML または base64 エンコードのバイナリで指定できます。
- segment timelines を伴う Segment templates。

公開されたマニフェストの場合、MediaTailor はオリジンサーバーからの更新で以下を変更しないことを要求します。

- start 属性で指定されたピリオド開始時間。

- ピリオド表現のセグメントテンプレート内の presentationTimeOffset の値。

ベストプラクティスとして、広告表示にコンテンツストリーム期間と同じ AdaptationSet および Representation 設定を与えます。AWS Elemental MediaTailor は、これらの設定を使用してコンテンツストリームに合わせて広告をトランスコードし、両者をスムーズに切り替えます。

以下のセクションでは、MediaTailor が DASH マニフェストで広告を処理する方法について詳しく説明します。

トピック

- [DASH 広告マーカ](#)
- [DASH の ad avail 継続時間](#)
- [DASH マニフェストセグメントの番号付け](#)
- [DASH MPD の例](#)
- [DASH のロケーション機能](#)

DASH 広告マーカ

MediaTailor

AWS Elemental MediaTailor は SCTE-35 キューアウトマーカを使用して、次のロジックを使用して DASH マニフェスト内の広告表示を識別します。

- 複数期間 DASH: MediaTailor は、SpliceInsert または TimeSignal キューアウトマーカ Period を含む各 Event の最初の の広告を挿入します。MediaTailor は、 内の追加の Event マーカを無視します Period。
- 単一期間 DASH: MediaTailor は、SpliceInsert または TimeSignal キューアウトマーカを含む広告 Period をそれぞれ Event に挿入します。

デフォルトでは、 は DASH マニフェストを複数期間のマニフェストとして AWS Elemental MediaTailor 管理します。オリジンサーバーからシングルピリオド DASH マニフェストを処理するように設定を変更することができます。詳細については、「[the section called “設定の作成”](#)」を参照してください。

以下のセクションでは、DASH 広告マーカの処理に関する詳細と、オリジンからのデコレーションされたマニフェストについて説明します。

DASH オリジンマニフェスト XML 要件

MediaTailor が広告ブレイクを識別するには、オリジンからの DASH マニフェストの広告マーカーが適切にフォーマットされている必要があります。以下のトピックでは、クリア XML でのこれらのフォーマット要件について説明します。

SpliceInsert クリア XML で

SpliceInsert クリア XML の広告マーカーには、以下が含まれている必要があります。

- EventStream には次の属性が必要です。 `schemeIdUri=urn:scte:scte35:2013:xml`
- Event は を保持する必要があります `scte35:SpliceInfoSection`
- `scte35:SpliceInfoSection` は を保持する必要があります `scte35:SpliceInsert`
- `scte35:SpliceInsert` には次の属性が必要です。 `outOfNetworkIndicator="true"`

Example **SpliceInsert** XML での

次の例では、必要な SCTE マーカーは太字です。

```
<Period start="PT444806.040S" id="123586" duration="PT15.000S">
  <EventStream timescale="90000" schemeIdUri="urn:scte:scte35:2013:xml">
    <Event duration="1350000">
      <scte35:SpliceInfoSection protocolVersion="0" ptsAdjustment="180832" tier="4095">
        <scte35:SpliceInsert spliceEventId="4026531855"
spliceEventCancelIndicator="false" outOfNetworkIndicator="true"
spliceImmediateFlag="false" uniqueProgramId="1" availNum="1" availsExpected="1">
          <scte35:Program><scte35:SpliceTime ptsTime="5672624400"/></scte35:Program>
          <scte35:BreakDuration autoReturn="true" duration="1350000"/>
        </scte35:SpliceInsert>
      </scte35:SpliceInfoSection>
    </Event>
    .
    .
    .
  </EventStream>
</Period>
```

TimeSignal クリア XML で

TimeSignal クリア XML の広告マーカーには、以下が含まれている必要があります。

- EventStream には次の属性が必要です。 `schemeIdUri=urn:scte:scte35:2013:xml`

- Event は を保持する必要があります scte35:SpliceInfoSection
- scte35:SpliceInfoSection は を保持する必要があります scte35:TimeSignal
- scte35:SpliceInfoSection も保持する必要があります
scte35:SegmentationDescriptor
- scte35:SegmentationDescriptor には次の属性が必要です。この値は有効な です [キューアウト番号](#)。segmentationTypeId="xx"
- scte35:SegmentationDescriptor は を保持する必要があります
scte35:SegmentationUpid

キューアウト番号

以下は、 でサポートされているキューアウト番号ですsegmentationTypeId。

セグメンテーションメッセージ	segmentationTypeId 値	16 進値
ディストリビューター広告の終了	51	0x51
ディストリビューターのアドバタイズ開始	50	0x32
ディストリビューター配置の機会の終了	55	0x37
ディストリビューター配置の機会の開始	54	0x36
終了ブレイク	35	0x23
プロバイダー広告の終了	49	0x31
プロバイダーアドバタイズ開始	48	0x30
プロバイダーオーバーレイ配置の機会の終了	57	0x39

セグメンテーションメッセージ	segmentationTypeId 値	16 進値
プロバイダーオーバーレイ配置の機会の開始	56	0x38
プロバイダー配置オポチュニティの終了	53	0x35
プロバイダー配置の機会の開始	52	0x34
開始時間枠	34	0x22

Example **TimeSignal** XML での

次の例では、必要な SCTE マーカーは太字です。

```
<Period start="PT346530.250S" id="178443" duration="PT61.561S">
  <EventStream timescale="90000" schemeIdUri="urn:scte:scte35:2013:xml">
    <Event duration="5310000">
      <scte35:SpliceInfoSection protocolVersion="0" ptsAdjustment="183003" tier="4095">
        <scte35:TimeSignal>
          <scte35:SpliceTime ptsTime="3442857000"/>
        </scte35:TimeSignal>
        <scte35:SegmentationDescriptor segmentationEventId="1414668"
segmentationEventCancelIndicator="false"
segmentationDuration="8100000" segmentationTypeId="52" segmentNum="0"
segmentsExpected="0">
          <scte35:DeliveryRestrictions webDeliveryAllowedFlag="false"
noRegionalBlackoutFlag="false" archiveAllowedFlag="false" deviceRestrictions="3"/>
          <scte35:SegmentationUpid segmentationUpidType="12"
segmentationUpidLength="2">0100</scte35:SegmentationUpid>
        </scte35:SegmentationDescriptor>
      </scte35:SpliceInfoSection>
    </Event>
    .
    .
    .
  </Period>
```

DASH オリジンマニフェスト base64 でエンコードされたバイナリ要件

MediaTailor が広告ブレイクを識別するには、オリジンからの DASH マニフェストの広告マーカが適切にフォーマットされている必要があります。以下のトピックでは、base64 でエンコードされたバイナリにおけるこれらのフォーマット要件について説明します。

base64 でエンコードされたマニフェストの TimeSignal と の両方の SpliceInsert 広告マーカには、以下が含まれている必要があります。

- EventStream には次の属性が必要です。 `urn:scte:scte35:2014:xml+bin`
- Event は を保持する必要があります `scte35:Signal`
- `scte35:Signal` は、base64 でエンコードされたバイナリ `scte35:Binary` を含む を保持する必要があります。

デコードされたバイナリ `splice_info_section` は、クリア XML 広告マーカに必要な情報と同じ情報を に提供する必要があります。

- コマンドタイプは、 `splice_insert()` または のいずれかである必要があります。
`time_signal()`
- 追加設定は、[TimeSignal クリア XML](#) で「」および「」で説明されている設定に準拠している必要があります [SpliceInsert クリア XML](#) で。

デコードされたバイナリで、平文の XML が `scte35:SpliceInfoSection` 要素で提供する情報と同じ一連の情報を含む `splice_info_section` を指定する必要があります。コマンドタイプは `splice_insert()` または `time_signal()` で、平文 XML 配信のために、追加設定は前述の内容に準拠している必要があります。

このオプションの例を以下に示します。必要なマーカは太字で表示されています。

```
<Period start="PT444806.040S" id="123586" duration="PT15.000S">
  <EventStream schemeIdUri="urn:scte:scte35:2014:xml+bin" timescale="1">
    <Event presentationTime="1541436240" duration="24" id="29">
      <scte35:Signal xmlns="http://www.scte.org/schemas/35/2016">
        <scte35:Binary>/DAhAAAAAAAAAAP/wEAUAAAHaf+9/fgAg9YDAAAAAAAAA25aoh</
scte35:Binary>
      </scte35:Signal>
    </Event>
    <Event presentationTime="1541436360" duration="24" id="30">
      <scte35:Signal xmlns="http://www.scte.org/schemas/35/2016">
```

```

    <scte35:Binary>QW5vdGhlciB0ZXN0IHN0cmVudGhlc3IgaW5jb2RpbmcgdG8gQmFzZTY0IGVuY29kZWQgYmluYXJ5Lg=
  </scte35:Binary>
    </scte35:Signal>
  </Event>

.
.
.
</Period>

```

以下は、前の例に表示されている最初のイベントのデコードされたバイナリです。splice_command_type の設定は 5 です。これは、splice_insert を表します。

```

{
  "table_id": 252,
  "section_syntax_indicator": false,
  "private_indicator": false,
  "section_length": 33,
  "protocol_version": 0,
  "encrypted_packet": false,
  "encryption_algorithm": 0,
  "pts_adjustment": 0,
  "cw_index": 0,
  "tier": "0xFFFF",
  "splice_command_length": 16,
  "splice_command_type": 5,
  "splice_command": {
    "splice_event_id": 448,
    "splice_event_cancel_indicator": false,
    "out_of_network_indicator": true,
    "program_splice_flag": true,
    "duration_flag": true,
    "splice_immediate_flag": false,
    "utc_splice_time": {
      "time_specified_flag": false,
      "pts_time": null
    },
  },
  "component_count": 0,
  "components": null,
  "break_duration": {
    "auto_return": false,
    "duration": {
      "pts_time": 2160000,

```

```
        "wall_clock_seconds": 24.0,  
        "wall_clock_time": "00:00:24:00000"  
    },  
    },  
    "unique_program_id": 49152,  
    "avail_num": 0,  
    "avails_expected": 0  
},  
"splice_descriptor_loop_length": 0,  
"splice_descriptors": null,  
"Scte35Exception": {  
    "parse_status": "SCTE-35 cue parsing completed with 0 errors.",  
    "error_messages": [],  
    "table_id": 252,  
    "splice_command_type": 5  
}  
}
```

DASH の ad avail 継続時間

再生中、 が広告表示 AWS Elemental MediaTailor を検出すると、その表示の一部またはすべてが広告に置き換えられます。MediaTailor は、ad avail の開始時点で広告置換を開始し、広告を以下のように含めます。

- ad avail が継続時間を指定する場合、MediaTailor は、後続のコンテンツを上書きすることなく、継続時間範囲内に収まる数の広告を含めます。
- 継続時間が指定されていない場合、MediaTailor は ad avail の終了時点に到達するまで広告を含めます。マルチピリオドマニフェストの場合は、これがピリオドの終了時点です。シングルピリオドマニフェストの場合は、これがイベントの終了時点です。MediaTailor は、ad avail の終了時点を超えて広告を再生しません。Mediatailor が終了時点に到達すると、後続のコンテンツを上書きするのではなく、現行の広告をカットします。

が広告表示期間 AWS Elemental MediaTailor を検索する方法

AWS Elemental MediaTailor は、次の順序で期間設定を検索します。

1. Event duration
2. スプライス挿入の場合は、scte35:BreakDuration duration
3. タイムシグナルの場合は scte35:SegmentationDescriptor segmentationDuration

AWS Elemental MediaTailor でこれらの設定が見つからない場合は、期間なしで広告インクルードを管理します。

以下は、Event が含まれる duration の例です。

```
<Period start="PT444806.040S" id="123586" duration="PT15.000S">
  <EventStream timescale="90000" schemeIdUri="urn:scte:scte35:2013:xml">
    <Event duration="1350000">
      <scte35:SpliceInfoSection protocolVersion="0" ptsAdjustment="180832"
tier="4095">
        <scte35:SpliceInsert spliceEventId="4026531855"
spliceEventCancelIndicator="false" outOfNetworkIndicator="true"
spliceImmediateFlag="false" uniqueProgramId="1" availNum="1" availsExpected="1">
          <scte35:Program><scte35:SpliceTime ptsTime="5672624400"/></
scte35:Program>
          <scte35:BreakDuration autoReturn="true" duration="1350000"/>
        </scte35:SpliceInsert>
      </scte35:SpliceInfoSection>
    </Event>
    ...
```

以下の例は、継続時間が指定されていない ad avail を示します。この Event には duration が指定されておらず、scte35:SpliceInsert 要素に scte35:BreakDuration 子要素は含まれません。

```
<Period start="PT444836.720S" id="123597" duration="PT12.280S">
  <EventStream timescale="90000" schemeIdUri="urn:scte:scte35:2013:xml">
    <Event>
      <scte35:SpliceInfoSection protocolVersion="0" ptsAdjustment="180832"
tier="4095">
        <scte35:SpliceInsert spliceEventId="4026531856"
spliceEventCancelIndicator="false" outOfNetworkIndicator="true"
spliceImmediateFlag="false" uniqueProgramId="1" availNum="1" availsExpected="1">
          <scte35:Program><scte35:SpliceTime ptsTime="5675385600"/></
scte35:Program>
        </scte35:SpliceInsert>
      </scte35:SpliceInfoSection>
    </Event>
    ...
```

DASH マニフェストセグメントの番号付け

MediaTailor は、<SegmentTimeline> および media 属性を使用して定義された <SegmentTemplate> 内のメディアセグメントをサポートします。media 属性のメディアセグメントリストを指定するには、\$Number\$ 識別子または \$Time\$ 識別子を使用します。

以下は、SegmentTemplate 識別子を使用する media 属性設定の \$Number\$ の例です。

```
<SegmentTemplate initialization="index_subtitles_4_0_init.mp4?m=1532451703" media="index_subtitles_4_0_$Number$.mp4?m=1532451703" presentationTimeOffset="1062336677920" startNumber="2349899" timescale="90000">
  <SegmentTimeline>
    <S d="540540" r="2" t="1062338840080"/>
    <S d="69069" t="1062340461700"/>
  </SegmentTimeline>
</SegmentTemplate>
```

以下は、SegmentTemplate 識別子を使用する media 属性設定の \$Time\$ の例です。

```
<SegmentTemplate
initialization="asset_720p_8000K_9_init.mp4" media="asset_720p_8000K_9_$Time$.mp4"
startNumber="1" timescale="90000">
  <SegmentTimeline>
    <S d="180000" r="2" t="0"/>
    <S d="147000" t="540000"/>
  </SegmentTimeline>
</SegmentTemplate>
```

DASH MPD の例

以下のセクションでは、DASH オリジン MPDs とパーソナライズMPDs の例を示します。これらの例を理解することは、MediaTailor ワークフローの設定とトラブルシューティングに役立ちます。

クエリパラメータを DASH マニフェストとセグメントに適用する方法については、「」を参照してください [MediaTailor DASH 暗黙的なセッションの初期化](#)。

DASH MPD 構造について

Dynamic Adaptive Streaming over HTTP (DASH) は、メディアプレゼンテーション説明 (MPD) マニフェストを使用してストリーミングコンテンツを配信します。MPD は、メディアコンテンツの構造と可用性を説明する XML ドキュメントです。

MPD (メディアプレゼンテーションの説明)

MPD は、メディアコンテンツの構造と可用性を説明する DASH ストリーミングのプライマリマニフェストファイルです。ストリーミングコンテンツを構成する期間、適応セット、表現、セグメントに関する情報が含まれています。

このマニフェストタイプは、DASH マニフェスト、DASH MPD、マスターマニフェスト (HLS と比較した場合)、プレゼンテーションマニフェストなど、さまざまなコンテキストの他のいくつかの名前でも知られています。

MediaTailor ワークフローでは、MPD は再生リクエストのエントリポイントであり、広告パーソナライゼーションが開始される場所です。

Period

Period は、DASH プレゼンテーションの一時的なセクションです。各期間には 1 つ以上の適応セットが含まれ、メディア時間の範囲を表します。広告挿入ワークフローでは、通常、個別の期間を使用してコンテンツと広告を区別します。

MediaTailor ワークフローでは、期間を使用してメインコンテンツを広告コンテンツから分離し、各広告は通常独自の期間で表されます。

AdaptationSet

AdaptationSet は、1 つまたは複数のメディアコンテンツコンポーネントの交換可能なエンコードされたバージョンのセットをグループ化します。たとえば、1 つの AdaptationSet には複数のビデオ品質レベルが含まれ、もう 1 つの AdaptationSet には複数のオーディオ言語オプションが含まれる場合があります。

MediaTailor ワークフローでは、コンテンツと広告の間で一貫したメディアタイプを維持するために、広告挿入中に AdaptationSets が保持されます。

表記

表現は、AdaptationSet 内のメディアコンテンツの特定のエンコードされたバージョンです。各表現は通常、ビットレート、解像度、またはその他のエンコーディングパラメータが異なるため、クライアントはネットワーク条件とデバイス機能に基づいて最適なバージョンを選択できます。

MediaTailor ワークフローでは、広告期間の表現は、コンテンツ期間の表現と可能な限り一致させて、スムーズな視聴体験を実現します。

DASH マニフェストタイプの詳細については、「」を参照してください[DASH マニフェストタイプ](#)。

での DASH マニフェストの構造と設定の詳細については AWS Elemental MediaPackage、DASH の概要に関する MediaPackage ユーザーガイドセクションを参照してください。

Live DASH MPD の例

このセクションでは、ライブ DASH MPDs。各例では、オリジンサーバーから受信した MPD と、MediaTailor が広告で MPD をパーソナライズした後の MPD を一覧表示します。

DASH MPD スプライス挿入の例

スプライス挿入の DASH オリジン MPD の例

MPD の次の例は、コンテンツオリジンから DASH が受信したマニフェストの広告表示を示しています。この例では、SpliceInsert マーカーを使用して広告表示を示します。

```
<Period start="PT173402.036S" id="46041">
  <EventStream timescale="90000" schemeIdUri="urn:scte:scte35:2013:xml">
    <Event duration="9450000">
      <scte35:SpliceInfoSection protocolVersion="0" ptsAdjustment="183265" tier="4095">
        <scte35:SpliceInsert spliceEventId="99" spliceEventCancelIndicator="false"
outOfNetworkIndicator="true" spliceImmediateFlag="false" uniqueProgramId="1"
availNum="1" availsExpected="1">
          <scte35:Program><scte35:SpliceTime ptsTime="7835775000"/></scte35:Program>
          <scte35:BreakDuration autoReturn="true" duration="9450000"/>
        </scte35:SpliceInsert>
        <scte35:SegmentationDescriptor segmentationEventId="99"
segmentationEventCancelIndicator="false" segmentationDuration="9450000">
          <scte35:DeliveryRestrictions webDeliveryAllowedFlag="true"
noRegionalBlackoutFlag="true" archiveAllowedFlag="true" deviceRestrictions="3"/>
          <scte35:SegmentationUpid segmentationUpidType="8" segmentationUpidLength="0"/>
        </scte35:SegmentationDescriptor>
        <scte35:SegmentationTypeID segmentationType="52"/>
        <scte35:SegmentNum segmentNum="1"/>
        <scte35:SegmentsExpected segmentsExpected="1"/>
      </scte35:SpliceInfoSection>
    </Event>
  </EventStream>
```



```

<AdaptationSet mimeType="video/mp4" segmentAlignment="true"
subsegmentAlignment="true" startWithSAP="1" subsegmentStartsWithSAP="1"
bitstreamSwitching="true">
  <Representation id="1" width="960" height="540" frameRate="30000/1001"
bandwidth="1000000" codecs="avc1.4D401F">
    <SegmentTemplate timescale="30000" media="index_video_1_0_$Number
$.mp4?m=1528475245" initialization="index_video_1_0_init.mp4?m=1528475245"
startNumber="178444" presentationTimeOffset="10395907501">
      <SegmentTimeline>
        <S t="10395907501" d="60060" r="29"/>
        <S t="10397709301" d="45045"/>
      </SegmentTimeline>
    </SegmentTemplate>
  </Representation>
</AdaptationSet>
<AdaptationSet mimeType="audio/mp4" segmentAlignment="0" lang="eng">
  <Representation id="2" bandwidth="96964" audioSamplingRate="48000"
codecs="mp4a.40.2">
    <SegmentTemplate timescale="48000" media="index_audio_2_0_$Number
$.mp4?m=1528475245" initialization="index_audio_2_0_init.mp4?m=1528475245"
startNumber="178444" presentationTimeOffset="16633452001">
      <SegmentTimeline>
        <S t="16633452289" d="96256" r="3"/>
        <S t="16633837313" d="95232"/>
        <S t="16633932545" d="96256" r="4"/>
        <S t="16634413825" d="95232"/>
        <S t="16634509057" d="96256" r="5"/>
        <S t="16635086593" d="95232"/>
        <S t="16635181825" d="96256" r="4"/>
        <S t="16635663105" d="95232"/>
        <S t="16635758337" d="96256" r="5"/>
        <S t="16636335873" d="71680"/>
      </SegmentTimeline>
    </SegmentTemplate>
  </Representation>
</AdaptationSet>
</Period>

```

このオリジン MPD の例:

- <EventStream> 要素には、広告表示を示す SCTE-35 マーカーが含まれています
- <scte35:SpliceInsert> 要素は、広告表示の詳細を提供します。
- <scte35:BreakDuration> 要素は、広告時間枠の期間を指定します。

- <AdaptationSet> 要素は、使用可能なビデオストリームとオーディオストリームを定義します。

スプライス挿入の DASH パーソナライズされた MPD の例

AWS Elemental MediaTailor は、広告仕様を使用して広告表示をパーソナライズします。パーソナライズには、プレーヤーから受信した視聴者データと現在進行中の広告キャンペーンが反映されます。

次の例は、が AWS Elemental MediaTailor パーソナライズした後の広告表示を示しています。

```
<Period id="178443_1" start="PT96H15M30.25S">
  <BaseURL>http://111122223333.cloudfront.net/nbc_fallback_2/</BaseURL>
  <AdaptationSet bitstreamSwitching="false" frameRate="30/1" mimeType="video/
mp4" segmentAlignment="true" startWithSAP="1" subsegmentAlignment="true"
subsegmentStartsWithSAP="1">
    <SegmentTemplate startNumber="1" timescale="90000"/>
    <Representation bandwidth="10000000" codecs="avc1.640028" height="1080" id="1"
width="1920">
        <SegmentTemplate initialization="nbc_fallback_ad_2_1080p_10init.mp4"
media="nbc_fallback_ad_2_1080p_10_${Number%09d$.mp4" startNumber="1" timescale="90000">
            <SegmentTimeline>
                <S d="180000" r="13" t="0"/>
                <S d="176940" t="2520000"/>
            </SegmentTimeline>
        </SegmentTemplate>
    </Representation>
    <Representation bandwidth="4000000" codecs="avc1.64001f" height="720" id="2"
width="1280">
        <SegmentTemplate initialization="nbc_fallback_ad_2_720p_9init.mp4"
media="nbc_fallback_ad_2_720p_9_${Number%09d$.mp4" startNumber="1" timescale="90000">
            <SegmentTimeline>
                <S d="180000" r="13" t="0"/>
                <S d="176940" t="2520000"/>
            </SegmentTimeline>
        </SegmentTemplate>
    </Representation>
  </AdaptationSet>
  <AdaptationSet mimeType="audio/mp4" segmentAlignment="0" lang="eng">
    <Representation id="8" bandwidth="128000" audioSamplingRate="48000"
codecs="mp4a.40.2">
        <SegmentTemplate initialization="nbc_fallback_ad_2_audio_2init.mp4"
media="nbc_fallback_ad_2_audio_2_${Number%09d$.mp4" startNumber="1" timescale="90000">
            <SegmentTimeline>
```

```

        <S d="180000" r="13" t="0"/>
        <S d="176940" t="2520000"/>
    </SegmentTimeline>
</SegmentTemplate>
</Representation>
</AdaptationSet>
</Period>

```

このパーソナライズされた MPD の例:

- MediaTailor が広告コンテンツの新しい期間を作成しました
- <BaseURL> 要素は広告コンテンツの場所を指します
- <AdaptationSet> 要素はコンテンツと同様の構造を維持します。
- <Representation> 要素は、広告コンテンツにさまざまな品質レベルを提供します。

DASH MPD タイムシグナルの例

時間信号の DASH オリジン MPD の例

MPD の次の例は、コンテンツオリジンから DASH が受信したマニフェストの広告表示を示しています。この例では、TimeSignal マーカーを使用して広告表示を示します。

```

<Period start="PT173402.036S" id="46041">
  <EventStream timescale="90000" schemeIdUri="urn:scte:scte35:2013:xml">
    <Event duration="9450000">
      <scte35:SpliceInfoSection protocolVersion="0" ptsAdjustment="183265" tier="4095">
        <scte35:TimeSignal>
          <scte35:SpliceTime ptsTime="7835775000"/>
        </scte35:TimeSignal>
        <scte35:SegmentationDescriptor segmentationEventId="99"
segmentationEventCancelIndicator="false" segmentationDuration="9450000">
          <scte35:DeliveryRestrictions webDeliveryAllowedFlag="true"
noRegionalBlackoutFlag="true" archiveAllowedFlag="true" deviceRestrictions="3"/>
          <scte35:SegmentationUpid segmentationUpidType="8" segmentationUpidLength="0"/>
        </scte35:SegmentationDescriptor>
        <scte35:SegmentationTypeID segmentationType="52"/>
        <scte35:SegmentNum segmentNum="1"/>
        <scte35:SegmentsExpected segmentsExpected="1"/>
      </scte35:SpliceInfoSection>
    </Event>
  </EventStream>
</Period>

```

```

    </Event>
  </EventStream>
  <AdaptationSet mimeType="video/mp4" segmentAlignment="true"
subsegmentAlignment="true" startWithSAP="1" subsegmentStartsWithSAP="1"
bitstreamSwitching="true">
    <Representation id="1" width="960" height="540" frameRate="30000/1001"
bandwidth="1000000" codecs="avc1.4D401F">
        <SegmentTemplate timescale="30000" media="index_video_1_0_$Number
$.mp4?m=1528475245" initialization="index_video_1_0_init.mp4?m=1528475245"
startNumber="178444" presentationTimeOffset="10395907501">
            <SegmentTimeline>
                <S t="10395907501" d="60060" r="29"/>
                <S t="10397709301" d="45045"/>
            </SegmentTimeline>
        </SegmentTemplate>
    </Representation>
  </AdaptationSet>
</Period>

```

このオリジン MPD の例:

- <scte35:TimeSignal> 要素が の代わりに使用される <scte35:SpliceInsert>
- <scte35:SegmentationDescriptor> は、広告表示に関する追加情報を提供します。

タイムシグナルの DASH パーソナライズされた MPD の例

AWS Elemental MediaTailor は、広告仕様を使用して広告表示をパーソナライズします。パーソナライズには、プレーヤーから受信した視聴者データと現在進行中の広告キャンペーンが反映されます。

次の例は、 が AWS Elemental MediaTailor パーソナライズした後の広告表示を示しています。

```

<Period id="178443_1" start="PT96H15M30.25S">
  <BaseUrl>http://111122223333.cloudfront.net/nbc_fallback_2/</BaseUrl>
  <AdaptationSet bitstreamSwitching="false" frameRate="30/1" mimeType="video/
mp4" segmentAlignment="true" startWithSAP="1" subsegmentAlignment="true"
subsegmentStartsWithSAP="1">
    <SegmentTemplate startNumber="1" timescale="90000"/>
    <Representation bandwidth="100000000" codecs="avc1.640028" height="1080" id="1"
width="1920">
        <SegmentTemplate initialization="nbc_fallback_ad_2_1080p_10init.mp4"
media="nbc_fallback_ad_2_1080p_10_$Number%09d$.mp4" startNumber="1" timescale="90000">
            <SegmentTimeline>

```

```
<S d="180000" r="13" t="0"/>
<S d="176940" t="252000"/>
</SegmentTimeline>
</SegmentTemplate>
</Representation>
</AdaptationSet>
</Period>
```

時間信号のパーソナライズされた MPD は、スプライス挿入の MPD と似ており、MediaTailor は広告コンテンツの新しい期間を作成します。

VOD DASH MPD の例

このセクションでは、ビデオオンデマンド (VOD) DASH MPDs。各例では、オリジンサーバーから受信した MPD と、MediaTailor が広告で MPD をパーソナライズした後の MPD を一覧表示します。

VOD DASH MPDs ライブ MPDs と同じ構造に従いますが、通常は MPD 要素に `type="static"` 属性があり、異なるコンテンツセグメントに複数の期間が含まれる場合があります。

VOD DASH MPDs の例については、の MediaTailor ドキュメントを参照してください [DASH 広告マーカ](#)。

パーソナライズMPDs の主な違い

MediaTailor が DASH MPDs をパーソナライズすると、いくつかの重要な変更が行われます。

期間処理

- 広告コンテンツ用に新しい期間が作成されます
- タイムラインの継続性を維持するために、期間の開始時刻が調整されます
- SCTE-35 マーカーを持つ EventStream 要素が処理および削除されます

AdaptationSet と表現の処理

- 広告期間の AdaptationSets は、コンテンツ AdaptationSets と一致するように作成されます。
- 広告コンテンツのさまざまな品質レベルに対して表現が作成されます。
- SegmentTemplate 要素は、広告コンテンツを指すように更新されます

これらの変更を理解することは、MediaTailor ワークフローの問題をトラブルシューティングし、CDN とプレイヤーの適切な設定を確保するのに役立ちます。

関連トピック

DASH MPDs と MediaTailor の詳細については、以下のトピックを参照してください。

- [DASH マニフェストタイプ](#) - DASH マニフェストタイプの詳細な説明
- [CDN を使用して MediaTailor の広告パーソナライゼーションとコンテンツ配信を最適化する](#) - MediaTailor での CDN の使用に関する情報
- [MediaTailor 広告挿入の仕組み](#) - MediaTailor 広告挿入の仕組みの概要
- DASH マニフェスト構造と MediaPackage 設定に関する包括的な情報については、DASH 概要の MediaPackage ユーザーガイドセクションを参照してください。

DASH のロケーション機能

このセクションでは、DASH のロケーション機能に関する情報を提供します。この機能は AWS Elemental MediaTailor でデフォルトで有効になっています。MediaTailor マニフェストにアクセスするためのコンテンツ配信ネットワーク (CDN) ルーティングルールを作成する場合は、このセクションをお読みください。スティッキー HTTP リダイレクトをサポートしていないプレイヤーでサーバー側のレポートを使用する場合も、このセクションをお読みください。

ロケーション機能とは

ロケーション機能を使用すると、スティッキー HTTP リダイレクトをサポートしていないプレイヤーでも、マニフェスト更新リクエストでスティッキー動作が可能になります。

AWS Elemental MediaTailor はセッションレス初期化を使用し、プレイヤーからのスティッキー HTTP リダイレクト動作が必要です。サーバー側のレポートでは、プレイヤーが MediaTailor に対してマニフェスト更新のリクエストを行うと、サービスが一時的な 302 リダイレクトを発行して、プレイヤーをパーソナライズされたマニフェスト用のエンドポイントに誘導します。MediaTailor は、セッション ID をクエリパラメータとしてレスポンスに含めます。この目的は、セッション全体でプレイヤーにこの URL を使用させることです。スティッキー HTTP リダイレクトをサポートしていないプレイヤーはリダイレクトをドロップして元の URL に戻ります。プレイヤーが元の URL に戻ると、MediaTailor は元のセッションを維持するのではなく、新しいリクエストごとに新しいセッションを作成します。これは、マニフェストが破損する原因になる場合があります。

DASH 仕様は、AWS Elemental MediaTailor 設定でデフォルトで有効になっているロケーション機能でこの問題の解決策を提供します。この機能が有効になっていると、MediaTailor がマニフェストの <Location> タグに絶対 URL を追加します。スティッキー HTTP リダイレクトをサポートして

いないプレイヤーは、<Location> で指定されている URL を使用してマニフェストの更新をリクエストできます。

設定でロケーション機能を無効にする必要がありますか？

ロケーション機能は、AWS Elemental MediaTailor マニフェストへのアクセス用に設定した CDN ルーティングルールを上書きするため、無効にすることが必要になる場合があります。ロケーション機能はコンテンツまたは広告セグメントの CDN キャッシュに影響しません。

以下のリストから該当する状況を見つけて、設定でロケーション機能を無効にする必要があるかどうか、その機能をどのように扱うかを判断します。

- AWS Elemental MediaTailor マニフェストにアクセスするための CDN ルーティングルールが設定されていない場合は、ロケーション設定を有効のままにします。
- それ以外の場合は、以下のルールを使用します。
 - サーバー側のレポートを使用しない場合、またはすべてのプレイヤーがスティッキー HTTP リダイレクトをサポートしている場合は、ロケーション機能を無効にします。コンソールでこの操作を行う方法については、「[the section called “設定の作成”](#)」を参照してください。
 - それ以外については、[AWS Support](#) までお問い合わせください。

ロケーション機能を使用する必要がありますか？

スティッキー HTTP リダイレクトをサポートしていないプレイヤーには、ロケーション機能を使用する必要があります。<Location> タグに指定されている URL をすべてのマニフェスト更新リクエストに使用します。

例

URL と <Location> タグの例:

- Example例: 初期リクエスト URL

```
https://777788889999.mediatailor.us-east-1.amazonaws.com/v1/  
dash/5ca4c1892b1f213a1247fad47b3e34c454a7d490/testLocationTag/index.mpd
```

- Example例: リダイレクト 302 レスポンス

```
/v1/dash/5ca4c1892b1f213a1247fad47b3e34c454a7d490/testLocationTag/index.mpd?  
aws.sessionId=0e5d9b45-ae97-49eb-901b-893d043e0aa6
```

- Example例: マニフェストのロケーションタグ

```
<Location>https://777788889999.mediatailor.us-east-1.amazonaws.com/v1/dash/5ca4c1892b1f213a1247fad47b3e34c454a7d490/testLocationTag/index.mpd?aws.sessionId=0e5d9b45-ae97-49eb-901b-893d043e0aa6</Location>
```

SigV4 との AWS Elemental MediaTailor オリジンインタラクションの保護

署名バージョン 4 (SigV4) は、HTTPS 経由でサポートされているオリジンへの MediaTailor リクエストを認証するために使用される署名プロトコルです。MediaTailor は HTTPS 通信のみをサポートし、HTTP 接続は許可しません。SigV4 署名では、MediaTailor は MediaTailor Channel Assembly、Amazon S3、および AWS Elemental MediaPackage バージョン 2 への HTTPS オリジンリクエストに署名付き認可ヘッダーを含めます。

オリジンで SigV4 を使用して、マニフェストリクエストが MediaTailor から送信され、署名付き認可ヘッダーが含まれている場合にのみ満たされるようにできます。これにより、不正な MediaTailor 再生設定がオリジンコンテンツにアクセスできなくなります。署名付きの認可ヘッダーが有効である場合は、オリジンがリクエストに対応します。有効でない場合は、リクエストが失敗します。

以下のセクションでは、サポートされているオリジンに MediaTailor SigV4 署名を使用するための要件について説明します。

MediaTailor チャネルアセンブリの要件

SigV4 を使用して MediaTailor Channel Assembly オリジンを保護する場合、MediaTailor がマニフェストにアクセスするには、次の要件を満たす必要があります。

- MediaTailor 設定のオリジンベース URL は、次の形式のチャネルアセンブリチャネルである必要があります。channel-assembly.mediatailor.*region*.amazonaws.com
- オリジンは HTTPS を使用するように設定する必要があります。MediaTailor は HTTPS 通信のみをサポートし、HTTP 接続は許可しません。オリジンで HTTPS が有効になっていない場合、MediaTailor はリクエストに署名しません。
- チャネルには、以下を含むオリジンアクセスポリシーが必要です。
 - MediaTailor がチャネルにアクセスするためのプリンシパルアクセス。mediataylor.amazonaws.com へのアクセスを許可します。
 - MediaTailor 設定で参照されるすべての多変量プレイリストを読み取るための IAM アクセス許可 mediataylor:GetManifest。

チャンネルでポリシーを設定する方法については、「」を参照してください[MediaTailor コンソールを使用してチャンネルを作成する](#)。

Example MediaTailor 設定アカウントを対象とする Channel Assembly のオリジンアクセスポリシー

```
{
  "Effect": "Allow",
  "Principal": {"Service": "mediatailor.amazonaws.com"},
  "Action": "mediatailor:GetManifest",
  "Resource": "arn:aws:mediatailor:us-west-2:777788889999:channel/ca-origin-channel",
  "Condition": {
    "StringEquals": {"AWS:SourceAccount": "777788889999"}
  }
}
```

Example MediaTailor 再生設定を対象とする Channel Assembly のオリジンアクセスポリシー

```
{
  "Effect": "Allow",
  "Principal": {"Service": "mediatailor.amazonaws.com"},
  "Action": "mediatailor:GetManifest",
  "Resource": "arn:aws:mediatailor:us-west-2:777788889999:channel/ca-origin-channel",
  "Condition": {
    "StringEquals": {"AWS:SourceArn": "arn:aws:mediatailor:us-west-2:777788889999:playbackConfiguration/test"}
  }
}
```

Amazon S3 の要件

SigV4 を使用して Amazon S3 オリジンを保護する場合、MediaTailor がマニフェストにアクセスするには、次の要件を満たす必要があります。

- MediaTailor 設定のオリジンベース URL は、次の形式の S3 バケットである必要があります。
`s3.region.amazonaws.com`
- オリジンは HTTPS を使用するように設定する必要があります。MediaTailor は HTTPS 通信のみをサポートし、HTTP 接続は許可しません。オリジンで HTTPS が有効になっていない場合、MediaTailor はリクエストに署名しません。
- チャンネルには、以下を含むオリジンアクセスポリシーが必要です。

- MediaTailor がバケットにアクセスするためのプリンシパルアクセス。mediatailor.amazonaws.com へのアクセスを許可します。

IAM でアクセスを設定する方法については、AWS Identity and [Access Management](#) ユーザーガイドの「アクセス管理」を参照してください。AWS Identity and Access Management

- MediaTailor 設定で参照されるすべての最上位マニフェストを読み取るための IAM アクセス許可 s3:GetObject。

Amazon S3 向けの SigV4 に関する一般的な情報については、Amazon S3 API リファレンスの「[リクエストの認証 \(AWS 署名バージョン 4\)](#) トピック」を参照してください。

Example MediaTailor アカウントを対象とする Amazon S3 のオリジンアクセスポリシー

```
{
  "Effect": "Allow",
  "Principal": {"Service": "mediatailor.amazonaws.com"},
  "Action": "s3:GetObject",
  "Resource": "arn:aws:s3:::mybucket/*",
  "Condition": {
    "StringEquals": {"AWS:SourceAccount": "111122223333"}
  }
}
```

Example MediaTailor 再生設定を対象とする Amazon S3 のオリジンアクセスポリシー

```
{
  "Effect": "Allow",
  "Principal": {"Service": "mediatailor.amazonaws.com"},
  "Action": "s3:GetObject",
  "Resource": "arn:aws:s3:::mybucket/*",
  "Condition": {
    "StringEquals": {"AWS:SourceArn": "arn:aws:mediatailor:us-west-2:111122223333:playbackConfiguration/test"}
  }
}
```

MediaPackage の要件

SigV4 を使用して MediaPackage v2 オリジンを保護する場合、MediaTailor がマニフェストにアクセスするには、次の要件を満たす必要があります。

- MediaTailor 設定のオリジンベース URL は、次の形式の MediaPackage v2 エンドポイントである必要があります。mediapackagev2.*region*.amazonaws.com
- オリジンは HTTPS を使用するように設定する必要があります。MediaTailor は HTTPS 通信のみをサポートし、HTTP 接続は許可しません。オリジンで HTTPS が有効になっていない場合、MediaTailor はリクエストに署名しません。
- チャンネルには、以下を含むオリジンアクセスポリシーが必要です。
 - MediaTailor がエンドポイントにアクセスするためのプリンシパルアクセス。mediatailor.amazonaws.com へのアクセスを許可します。
 - MediaTailor 設定で参照されるすべての多変量プレイリストを読み取るための IAM アクセス許可 mediapackagev2:GetObject。

MediaPackage v2 用 SigV4 の一般的な情報については、MediaPackage v2 API リファレンスの [「リクエストの認証 \(AWS 署名バージョン 4\)」](#) トピックを参照してください。

Example MediaTailor アカウントを対象とする MediaPackage v2 のオリジンアクセスポリシー

```
{
  "Effect": "Allow",
  "Principal": {"Service": "mediatailor.amazonaws.com"},
  "Action": "mediapackagev2:GetObject",
  "Resource": "arn:aws:mediapackagev2:us-west-2:444455556666:channelGroup/emp-origin-channel-group/channel/emp-origin-channel/originEndpoint/emp-origin-endpoint",
  "Condition": {
    "StringEquals": {"AWS:SourceAccount": "444455556666"}
  }
}
```

Example MediaTailor 再生設定を対象とする MediaPackage v2 のオリジンアクセスポリシー
MediaTailor

```
{
  "Effect": "Allow",
  "Principal": {"Service": "mediatailor.amazonaws.com"},
  "Action": "mediapackagev2:GetObject",
  "Resource": "arn:aws:mediapackagev2:us-west-2:444455556666:channelGroup/emp-origin-channel-group/channel/emp-origin-channel/originEndpoint/emp-origin-endpoint",
  "Condition": {
    "StringEquals": {"AWS:SourceArn": "arn:aws:mediatailor:us-west-2:444455556666:playbackConfiguration/test"}
  }
}
```

```
}  
}
```

Google の広告プラットフォーム AWS Elemental MediaTailor との統合

MediaTailor はserver-to-server接続を自動的に認証します。DV360 MediaTailor は、広告リクエストまたは追跡イベントが Google 広告サーバー宛てであることを検出すると、必要な安全で認証された接続を自動的に確立します。

関連する Google 広告プラットフォームでアカウントを設定する必要があります。MediaTailor は、次の統合タイプをサポートしています。

- サーバー側の統合: MediaTailor は、Google の広告プラットフォームへのサーバー側の広告リクエストとインプレッション追跡リクエストを自動的に保護します。これには、以下が含まれます。
 - Google 広告マネージャー (GAM): パブリッシャー向けの Google の広告サーバーへの広告リクエストは自動的に保護されます。これは、Google のリアルタイム広告販売マーケットプレイスおよび広告交換である認定購入者にアクセスするために必要です。
 - Google Campaign Manager (GCM): 広告配信および追跡リクエストは、Google の認証されたエンドポイントを介して自動的にルーティングされ、セキュリティで保護されます。これにより、これらのプラットフォームでキャンペーンを実行する広告主は、より正確なレポートと拒否されたインプレッションが少なくなります。
 - Display & Video 360 (DV360): 広告リクエストは Google の認証済みエンドポイントを介して自動的にルーティングされ、セキュリティで保護されます。これにより、これらのプラットフォームでキャンペーンを実行する広告主は、より正確なレポートと拒否されたインプレッションがなくて済みます。
- クライアント側の統合: Google Programmatic Access Libraries (PAL) SDK を使用したプレイヤーの統合。この統合は、Google 広告マネージャーでオープン入札トランザクションタイプを使用する場合に必要です。

他のすべての広告リクエストは、変更なしで動作し続けます。

プログラムによるトランザクションタイプの Ad Manager のサポートは、使用している統合のタイプによって異なります。使用可能なオプションのリストについては、[「トランザクションタイプ」](#)を参照するか、Google アカウントチームにお問い合わせください。

トピック

- [Google の広告プラットフォームとのサーバー側の AWS Elemental MediaTailor 統合](#)
- [Google 広告マネージャーとのクライアント側の AWS Elemental MediaTailor 統合](#)

Google の広告プラットフォームとのサーバー側の AWS Elemental MediaTailor 統合

MediaTailor は、Google が発行した SSL 証明書を使用して、Google 広告マネージャー (GAM)、Google キャンペーンマネージャー (GCM)、Google Display & Video 360 (DV360) との server-to-server 接続を自動的に認証します AWS Elemental MediaTailor。MediaTailor は、Google 広告サーバー宛ての広告リクエストまたは追跡イベントを検出すると、必要な安全で認証された接続を自動的に確立します。追加の SSL 証明書のセットアップやサポートチケットは必要ありません。

前提条件: 関連する Google 広告プラットフォーム (Google 広告マネージャー、Google キャンペーンマネージャー、または Display & Video 360) を持つアクティブなアカウントが必要です。

MediaTailor は、次の送信先 URI ルールに基づいて Google が発行した SSL 証明書認証を適用します。

Google 広告マネージャー (GAM): サーバー側の広告リクエスト

- このルールは `serverside.doubleclick.net`、そのドメインで ADS テンプレート URL を設定する場合や、リダイレクト広告リクエストが解決される場合など、への広告リクエストでトリガーされます。
- MediaTailor は、`ssss=mediatailor` クエリパラメータが存在しない場合は追加し、Google が発行した SSL 証明書を使用して接続を保護します。
- これにより、Google のリアルタイムの広告販売市場と広告交換である認定購入者にアクセスできます。

Google 広告マネージャー (GAM): VMAP 内でリダイレクトされた広告リクエスト

- このルールは、MediaTailor が現在のリクエスト `serverside.doubleclick.net` でへの VAST pubads.g.doubleclick.net リダイレクトを以前に確認していて、以前の ADS レスポンスが VMAP レスポンスであった場合に、リダイレクトされた広告リクエストでトリガーされます。
- MediaTailor はドメインを に自動的に変更 `serverside.doubleclick.net` し、Google が発行した SSL 証明書を使用して認証された接続を確立します。
- これにより、VMAP ベースの広告ワークフローの認定購入者にアクセスできます。

Google Campaign Manager (GCM): 広告提供および追跡リクエスト

- このルールは、URI に `/ddm/pfadx/` または `/ddm/trackimp/` (エンコードまたはエンコードされていない) が含まれている場合、`ad.doubleclick.net` 広告配信リクエストと追跡リクエストでトリガーされます。
- MediaTailor はドメインを に自動的に変更 `serverside.doubleclick.net` し、Google が発行した SSL 証明書を使用して認証された接続を確立します。
- これにより、これらのプラットフォームでキャンペーンを実行する広告主は、より正確なレポートと拒否されたインプレッションが少なくなります。

Display & Video 360 (DV360): 広告リクエスト

- このルールは、URI に `/dbm/vast/` (エンコードまたはエンコードされていない) が含まれている場合、`bid.g.doubleclick.net` または `pagead2.googlesyndication.com` 広告リクエストでトリガーされます。
- MediaTailor はドメインを に自動的に変更 `serverside.doubleclick.net` し、Google が発行した SSL 証明書を使用して認証された接続を確立します。
- これにより、ディスプレイとビデオ 360 キャンペーンの認証済み広告配信が有効になります。

他のすべての広告リクエストは、Google が発行した SSL 証明書認証なしで処理されます。

Note

SSL 証明書認証が自動になりました。以前に SSL 証明書を有効にするサポートチケットを送信した場合、アクションは必要ありません。設定は引き続き機能します。

Google 広告マネージャーとのクライアント側の AWS Elemental MediaTailor 統合

Google 広告マネージャープログラムアクセスライブラリ (PAL) SDKs を使用し、Google 広告マネージャーのオープン入札トランザクションタイプを使用するには、MediaTailor クライアント側の統合が必要です。

PAL SDKs は、再生セッションのコンテンツ、デバイス、ユーザーデータに関する情報を提供します。PAL SDK を通じて、この情報を Google 広告マネージャーに提供できます。これにより、表示するターゲット広告をよりの確に判断できます。SDKs は、Android、iOS、HTML5、および Cast で使用できます。PAL SDKs [「Google 広告マネージャー PAL SDK」](#) を参照してください。

Ad Manager とのクライアント側の統合を作成するには

1. PAL SDK を使用してノンスを生成します。ノンスの設定については、[Google Ad Manager PAL SDK の SDK](#) ドキュメントを参照してください。

 Note

nonce は、PAL がストリームリクエスト用に生成する暗号化された文字列です。各リクエストには一意の nonce が必要です。

2. ADS リクエストで givn パラメータを使用して、nonce 値を渡します。これを行うには、ADS URL を更新して `&givn=[player_params.givn]` を含めます。手順については、「[クライアント側の追跡の有効化](#)」を参照してください。

Datazoom プレイヤー SDKs

MediaTailor は Datazoom と提携し、Google 広告マネージャー PAL で提供されているような SDKs との統合を容易にする無料プレイヤー SDKs を提供しています。Datazoom と MediaTailor のパートナーシップの詳細については、「」を参照してください [Datazoom 無料プレイヤー SDKs](#)。Datazoom プレイヤー SDKs にアクセスするには、[Datazoom with AWS](#) サイトの連絡先情報を使用します。

広告ブレイク抑制による広告ブレイク動作のカスタマイズ

で設定を作成するときに AWS Elemental MediaTailor、広告ブレイクの抑制を設定する機能など、広告ブレイクの動作を管理するオプションの広告ブレイク設定を指定できます。これにより、特定の要件を満たすように動画コンテンツの広告ブレイクエクスペリエンスを調整できます。

互換性の制限

以下では広告ブレイク抑制を使用できません。

- VOD ワークフローと live-to-VOD ワークフロー。ライブワークフローのみがサポートされています。
- サーバーガイド広告挿入 (SGAI) メソッド。サーバーガイド方式は広告決定を異なる方法で処理し、抑制設定を必要としません。

トピック

- [広告ブレイク抑制の設定](#)

広告ブレイク抑制の設定

ライブコンテンツ用の広告ブレイクのパーソナライゼーションをスキップするように MediaTailor を設定できます。これは、広告ブレイク抑制、または表示抑制と呼ばれます。このトピックでは、 の使用方法と、広告ブレイク抑制の設定の仕組みについても説明します。

広告ブレイク抑制は、次のユースケースに使用できます。

- 長いマニフェストルックバックウィンドウ - 視聴者がマニフェストのライブエッジで再生を開始するが、ルックバックウィンドウが長いという場合は、視聴者による視聴の開始後のみに広告を挿入することが推奨されます。または、マニフェストのルックバックウィンドウ全体の一部に広告を挿入します。広告抑制は、MediaTailor がライブエッジより前の指定時間、または指定時間範囲内の広告ブレイクをパーソナライズするように設定できます。
- ブレイク中の参加 - 視聴者が広告ブレイクの途中でライブビデオストリームの視聴を開始した場合、そのユーザーはチャンネルを変更して広告を視聴しない可能性が高くなります。広告抑制を使用すると、視聴者がストリームに参加する前に広告時間枠が開始された場合に、広告時間枠のパーソナライズをスキップできます。

広告抑制の設定

広告抑制を使用するには、次の方法で avail 抑制モード、avail 抑制値、avail 抑制フィルポリシーを設定します。

- MediaTailor コンソールで
- AWS Command Line Interface (AWS CLI) の使用
- MediaTailor API を使用するか、クライアントの再生セッションリクエストのパラメータとして使用する

パラメータを使用した設定の詳細については、「」を参照してください[広告抑制パラメータの設定 – 再生セッションリクエスト](#)。

広告抑制設定パラメータ

広告抑制のオンとオフを選択できます。広告抑制を有効にする場合は、その抑制がライブ再生エッジの後か、ライブストリームのライブ再生エッジの前かを指定します。いずれの場合も、MediaTailor

が広告をパーソナライズしないライブエッジを基準にした時間も指定します。表示抑制を有効にすると、セッションが中断の途中で開始したときに MediaTailor が部分的な広告ブレイクフィルに使用する表示抑制ポリシーを指定できます。

広告抑制の設定パラメータは次のとおりです。

- [Avail suppression mode] (Avail 抑制モード) – 広告抑制モードを設定します。広告抑制は、デフォルトで無効になっています。使用できる値: OFF、BEHIND_LIVE_EDGE、または AFTER_LIVE_EDGE。
 - OFF: 広告抑制はなく、MediaTailor はすべての広告時間枠をパーソナライズします。
 - BEHIND_LIVE_EDGE: MediaTailor は、ライブエッジより前に開始する広告ブレイクを、Avail 抑制値を引いてパーソナライズしません。これは、個々の広告表示だけでなく、広告時間枠全体に影響します。
 - AFTER_LIVE_EDGE: MediaTailor は、ライブエッジ内の広告ブレイクと Avail 抑制値をパーソナライズしません。これは、広告時間枠全体に影響を与えるか、広告表示の部分的なフィルを許可するように設定できます。
- Avail suppression value (Avail 抑制値) – ライブストリーム内のライブエッジに相対する時間。使用できる値: HH:MM:SS 形式の時間値。
- Avail suppression fill policy – MediaTailor が Avail suppression モードに適用するポリシー を定義します。許容値: PARTIAL_AVAIL、FULL_AVAIL_ONLY。
 - BEHIND_LIVE_EDGE モードは常にFULL_AVAIL_ONLY抑制ポリシーを使用します。
 - AFTER_LIVE_EDGE モードを使用すると、セッションがブレイクの途中で開始したときにPARTIAL_AVAIL広告ブレイクフィルを呼び出すことができます。

広告抑制設定の例

[広告抑制設定パラメータ](#)が相互にやり取りする方法により、ライブストリームのライブエッジの前、とき、または後に広告抑制と表示フィルを処理するいくつかの異なる方法を指定できます。このセクションでは、これらのやり取りの一部を示す例を示します。これらの例を使用して、特定の状況の設定パラメータを設定します。

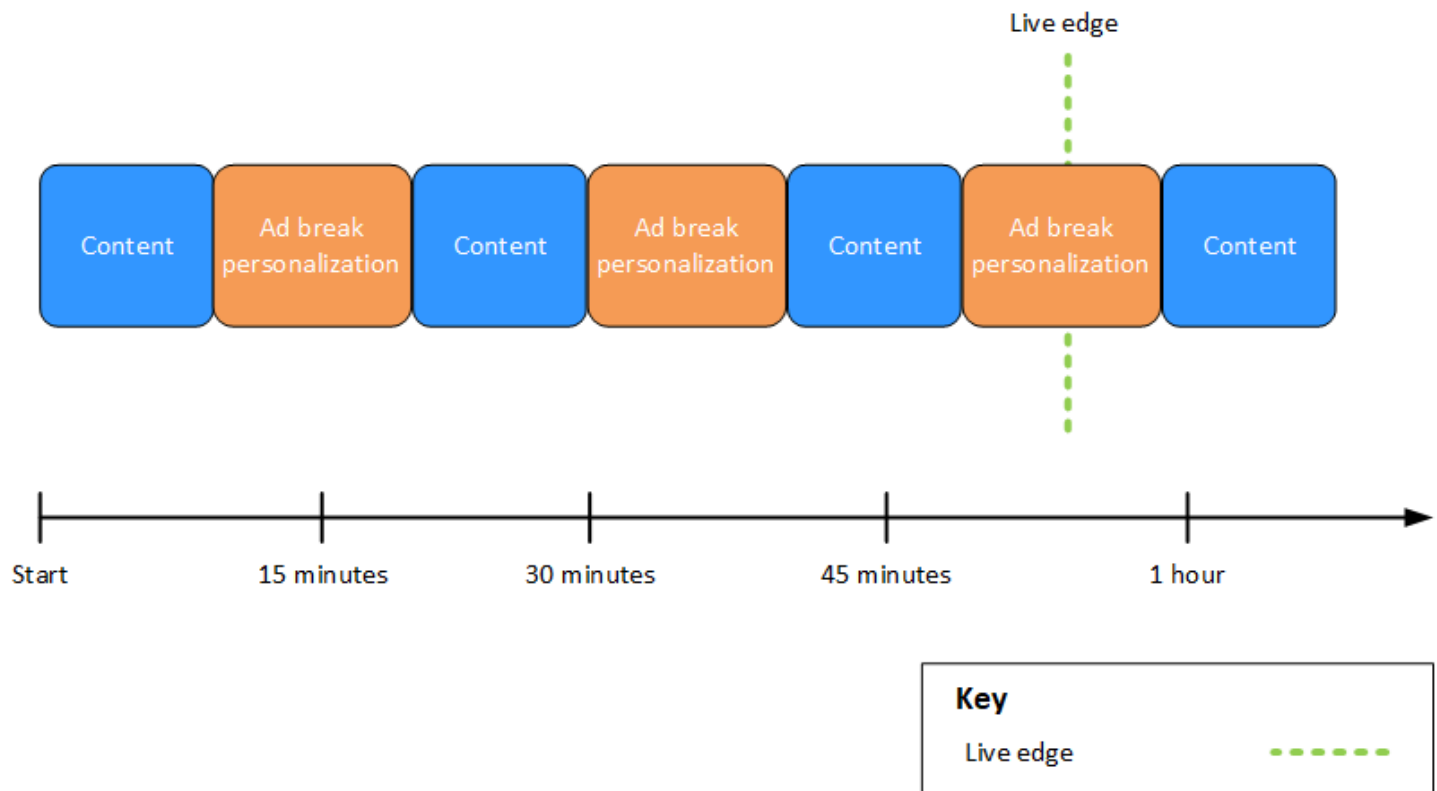
以下は、広告抑制設定の例です。

Example 1: 広告抑制なし

表示抑制モードが の場合OFF、広告抑制はなく、MediaTailor はすべての広告ブレイクをパーソナライズします。

次の図では、さまざまなブロックが、左から右に進行するタイムラインに沿って水平に配置されています。各ブロックは、ライブストリームまたはパーソナライズされた広告ブレークのコンテンツが再生される時間の一部を表します。点線は、ライブストリームの現在のライブエッジを表します。ライブエッジの前に 2 つの広告ブレークが発生し、ライブエッジで別の広告ブレークが進行中です。図に示すように、表示抑制モードが `OFF` の場合、MediaTailor はタイムラインのライブエッジの前に発生するすべての広告ブレークをパーソナライズします。MediaTailor は、ライブエッジで進行中の広告ブレークもパーソナライズします。

Avail suppression mode (default): `OFF`



Example 2: ライブエッジと同期した値を持つ `BEHIND_LIVE_EDGE` 広告抑制

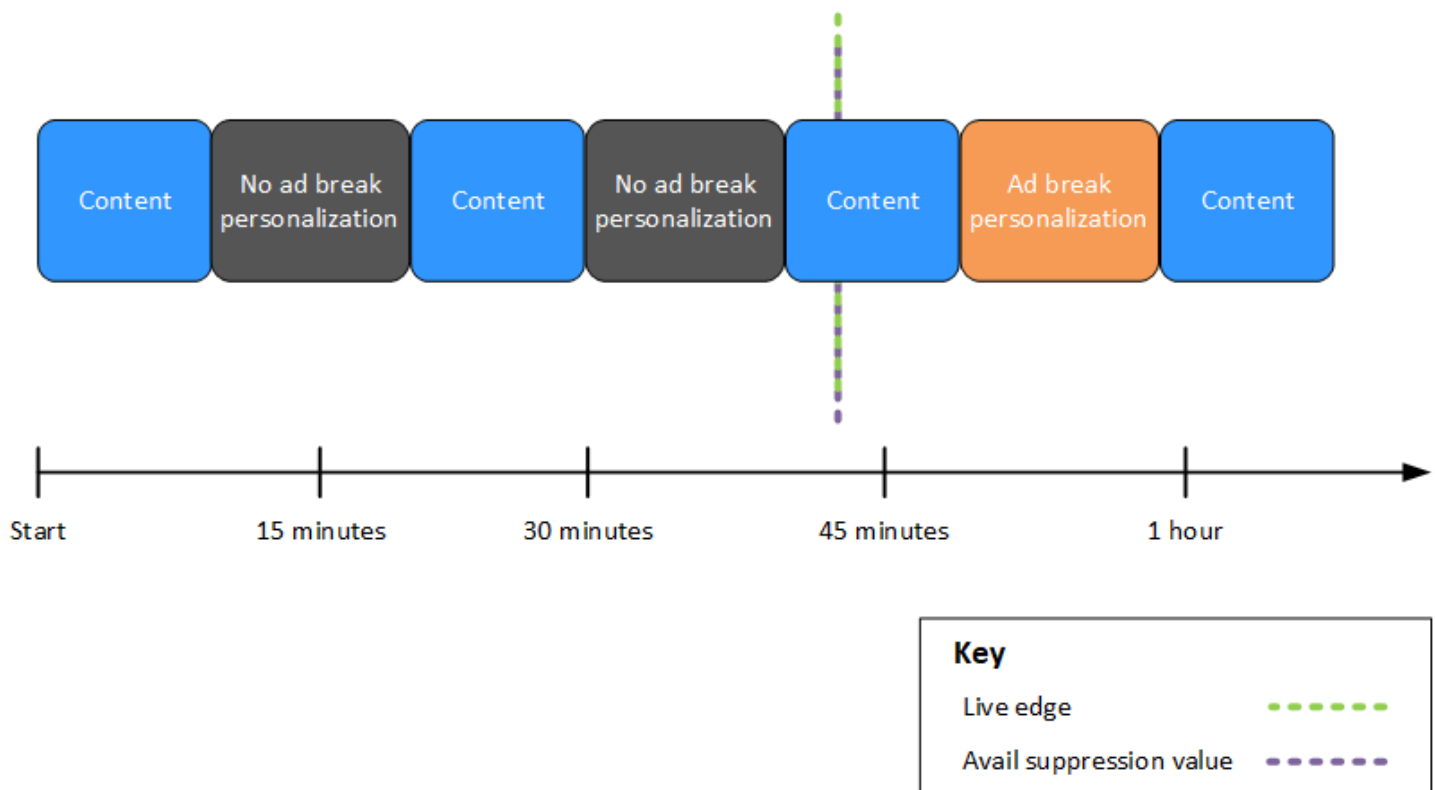
avail 抑制モードが `BEHIND_LIVE_EDGE` に設定され、avail 抑制値が `00:00:00` に設定されている場合、avail 抑制値はライブエッジと同期されます。MediaTailor は、ライブエッジまたはそれより前に開始される広告ブレークをパーソナライズしません。

次の図では、さまざまなブロックが、左から右に進行するタイムラインに沿って水平に配置されています。各ブロックは、ライブストリームのコンテンツ、パーソナライズされた広告時間枠、またはパーソナライズされていない広告時間枠が再生される時間の一部を表します。点線は、ライブストリームの現在のライブエッジを表します。に設定された表示抑制値を表す別の点線

は00:00:00、ライブエッジの点線と重複しています。ライブエッジの前に2つの広告ブレイクが発生し、ライブエッジの後に別の広告ブレイクが発生します。図に示すように、avail 抑制モードを に設定しBEHIND_LIVE_EDGE、avail 抑制値を に設定00:00:00してライブエッジと同期させると、MediaTailor はタイムラインのライブエッジの前に発生する広告ブレイクをパーソナライズしません。MediaTailor は、ライブエッジの後に発生する広告ブレイクをパーソナライズします。

Avail suppression mode: **BEHIND_LIVE_EDGE**

Avail suppression value: **00:00:00**



Example 3: ライブエッジの背後にある値を持つBEHIND_LIVE_EDGE広告抑制

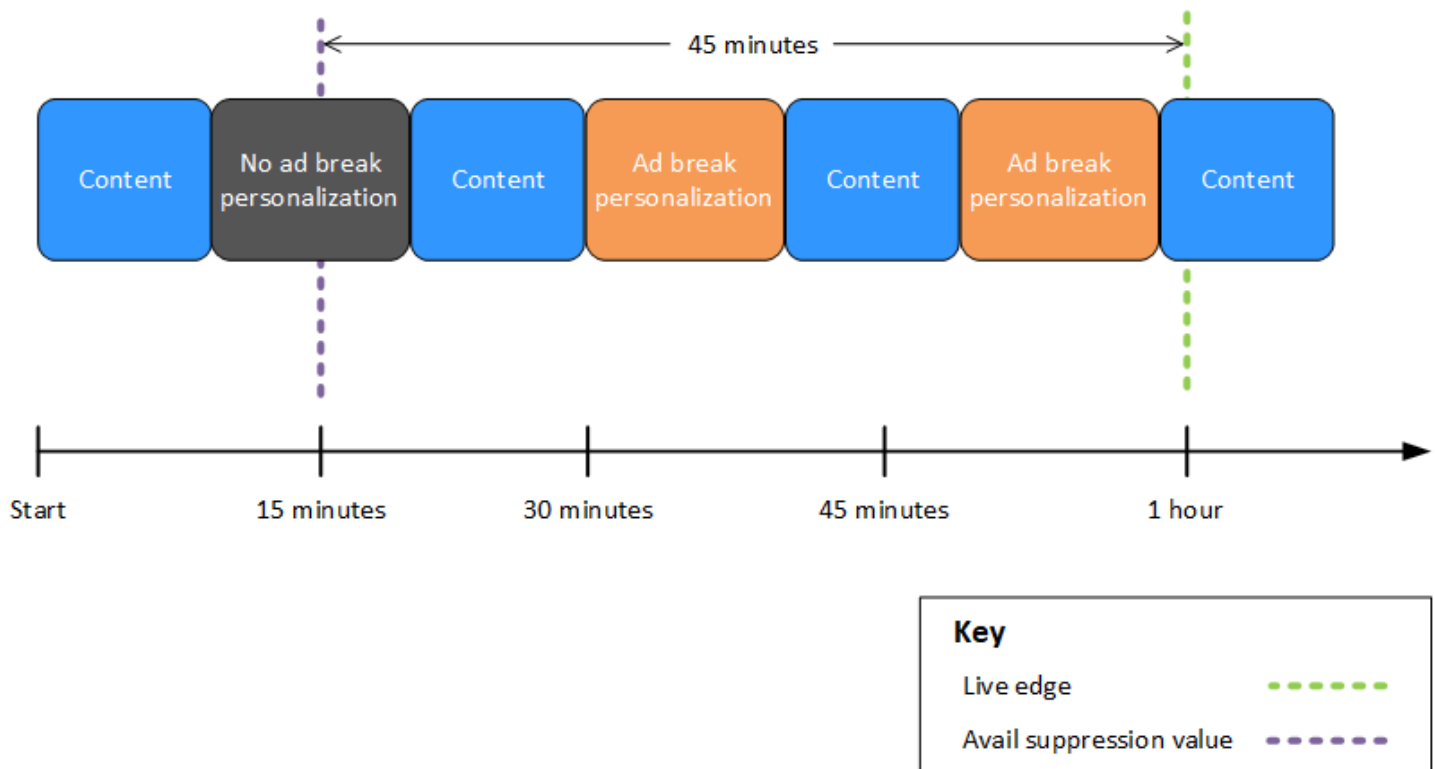
表示抑制モードが に設定されている場合BEHIND_LIVE_EDGE、MediaTailor はそれ以前に広告時間枠をパーソナライズしません。この例では、MediaTailor はライブエッジから最大 45 分遅れて開始する広告ブレイクをパーソナライズします。MediaTailor は、ライブエッジから 45 分以上遅れて開始する広告ブレイクをパーソナライズしません。

次の図では、さまざまなブロックが、左から右に進行するタイムラインに沿って水平に配置されています。各ブロックは、ライブストリームのコンテンツ、パーソナライズされた広告時間枠、またはパーソナライズされていない広告時間枠が再生される時間の一部を表します。点線は、ライブストリームの現在のライブエッジを表します。に設定された表示抑制値を表す別

の点線は、ライブエッジの点線に関して00:45:00タイムラインの 45 分前に発生します。点線の間の 45 分間は、表示抑制期間を表します。広告時間枠は、表示抑制期間の開始時に進行中です。その他の 2 つの広告時間枠は、表示抑制期間中に発生します。図に示すように、表示抑制モードを に設定しBEHIND_LIVE_EDGE、表示抑制値をライブエッジの00:45:00背後に設定すると、MediaTailor は表示抑制期間内に発生した広告ブレイクをパーソナライズします。MediaTailor は、表示抑制期間の開始時に進行中の広告ブレイクをパーソナライズしません。

Avail suppression mode: **BEHIND_LIVE_EDGE**

Avail suppression value: **00:45:00**



Example 4: 表示抑制期間中に**AFTER_LIVE_EDGE**広告中断が発生しない広告抑制

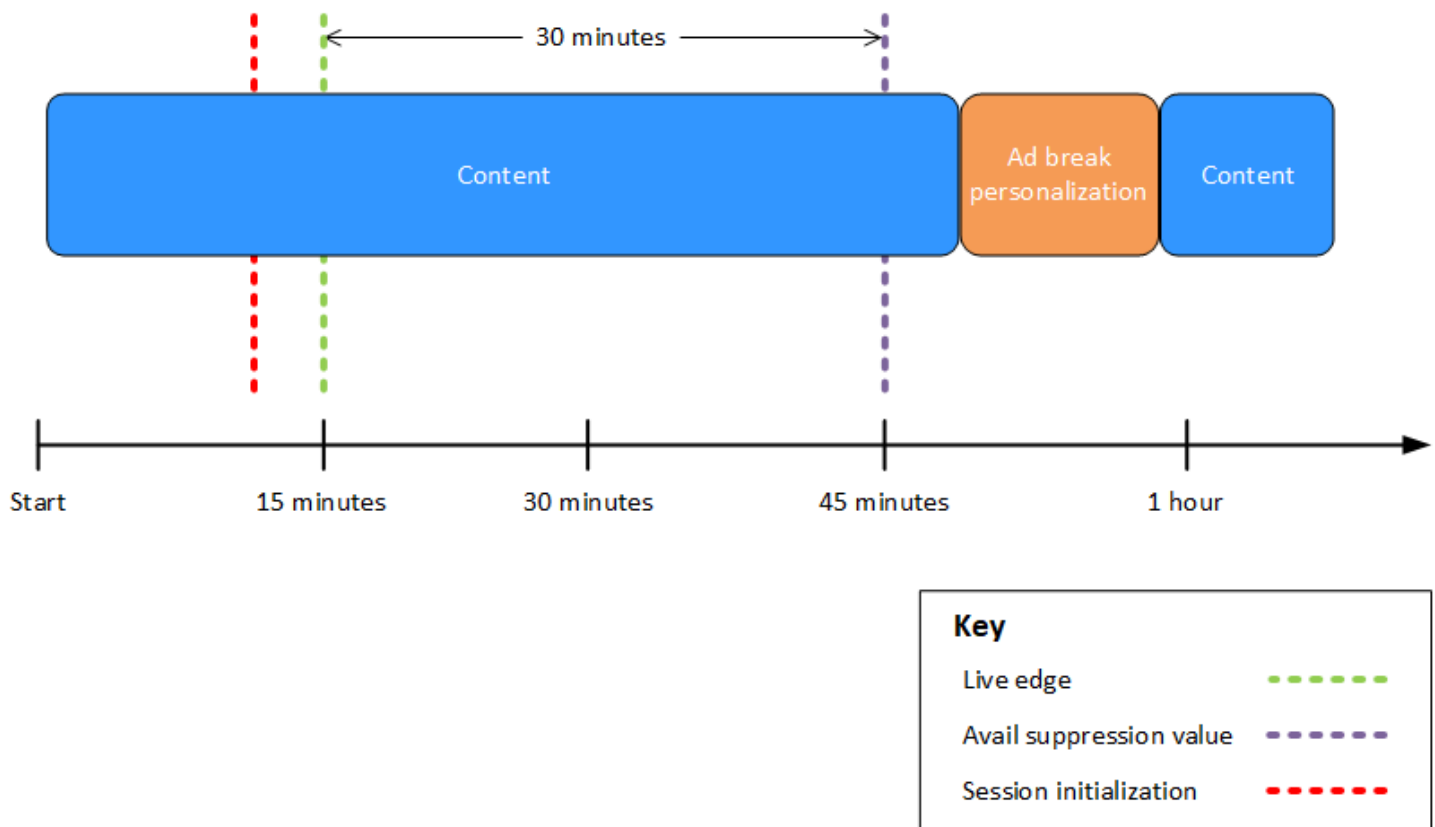
表示抑制モードが に設定**AFTER_LIVE_EDGE**され、表示抑制値が 0 より大きい場合、MediaTailor はセッションの経過時間がその値に達するまで広告ブレイクをパーソナライズしません。

次の図では、さまざまなブロックが、左から右に進行するタイムラインに沿って水平に配置されています。各ブロックは、ライブストリームまたはパーソナライズされた広告ブレイクのコンテンツが再生される時間の一部を表します。点線は、ライブストリームの現在のライブエッジを表します。に設定された表示抑制値を表すもう 1 つの点線は00:30:00、ライブエッジの点線に関してタイムラインの 30 分後に発生します。セッションの初期化を表す 3 番目の点線は、ライブエッ

ジの点線に関してタイムラインの前半で発生します。ライブエッジ時間と avail-suppression-value 時間の間の 30 分の期間は、avail 抑制期間を表します。広告ブレイクは、表示抑制期間の後に発生します。図に示すように、avail 抑制モードが `AFTER_LIVE_EDGE` に設定されている場合、avail 抑制値はライブエッジの 00:30:00 後に設定され、セッションの初期化はライブエッジの前に行われます。MediaTailor は、avail 抑制期間後に発生する広告ブレイクをパーソナライズします。

Avail suppression mode: `AFTER_LIVE_EDGE`

Avail suppression value: `00:30:00`



Example 5: **PARTIAL_AVAIL** フィルポリシーによる **AFTER_LIVE_EDGE** 広告抑制と、表示抑制期間の終了時に進行中の広告ブレイク

avail 抑制モードが `AFTER_LIVE_EDGE` に設定され、avail 抑制値が 0 より大きい場合、MediaTailor はセッションの経過時間がその値に達するまで広告ブレイクをパーソナライズしません。

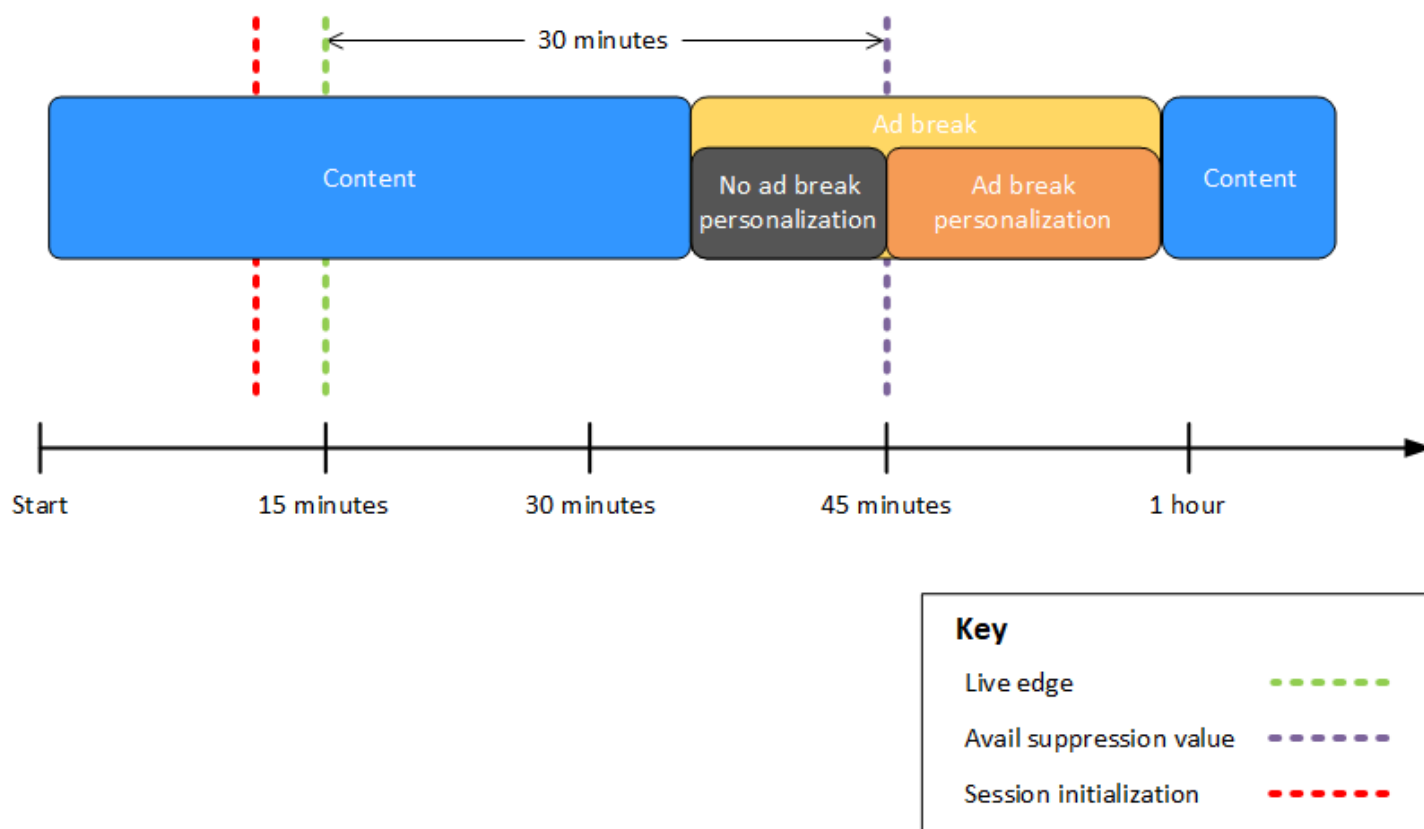
次の図では、さまざまなブロックが、左から右に進行するタイムラインに沿って水平に配置されています。各ブロックは、ライブストリームのコンテンツ、パーソナライズされた広告時間枠、またはパーソナライズされていない広告時間枠が再生される時間の一部を表します。

点線は、ライブストリームの現在のライブエッジを表します。に設定された表示抑制値を表すもう 1 つの点線は 00:30:00、ライブエッジの点線に関してタイムラインの 30 分後に発生します。セッションの初期化を表す 3 番目の点線は、ライブエッジの点線に関してタイムラインの前半で発生します。ライブエッジ時間と avail-suppression-value 時間の間の 30 分の期間は、avail 抑制期間を表します。広告時間枠は、表示抑制期間の終了時に進行中です。図に示すように、avail 抑制モードが に設定され AFTER_LIVE_EDGE、avail 抑制値がライブエッジ 00:30:00 の後に に設定され、avail 抑制フィルポリシーが に設定され PARTIAL_AVAIL、セッションの初期化がライブエッジの前に行われると、MediaTailor は、avail 抑制期間後に発生する広告ブレイクをパーソナライズします。表示抑制期間の終了時に進行中の広告ブレイクの場合、MediaTailor は表示抑制期間後に発生する広告ブレイクの部分をパーソナライズしますが、表示抑制期間中に発生する広告ブレイクの部分はパーソナライズしません。

Avail suppression mode: AFTER_LIVE_EDGE

Avail suppression value: 00:30:00

Avail suppression fill policy: PARTIAL_AVAIL



Example 6: **PARTIAL_AVAIL** フィルポリシーによる **AFTER_LIVE_EDGE** 広告抑制と、セッションの初期化前から表示抑制期間終了後まで進行中の広告ブレイク

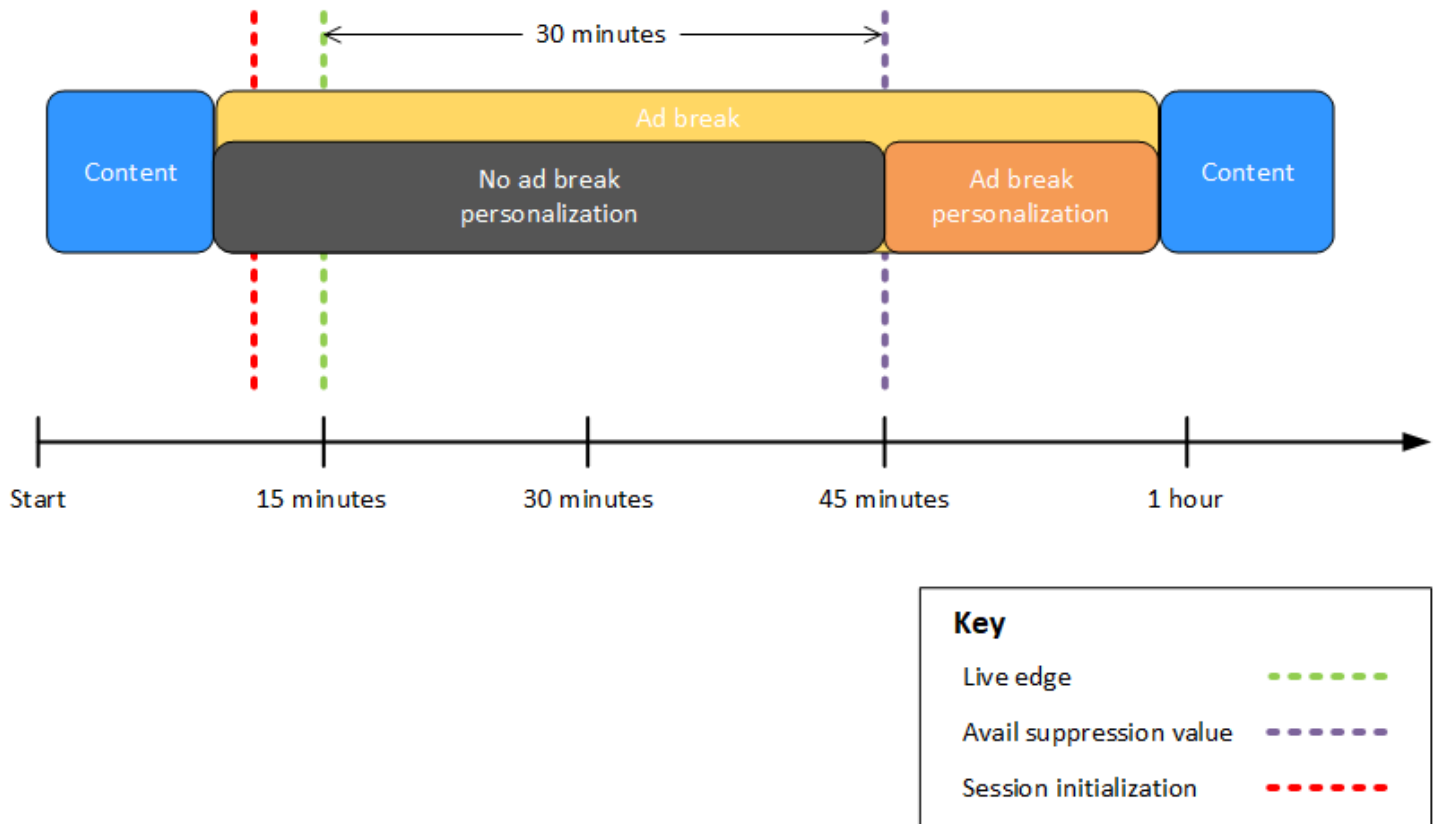
avail 抑制モードが に設定 **AFTER_LIVE_EDGE** され、avail 抑制値が 0 より大きい場合、MediaTailor はセッションの経過時間がその値に達するまで広告ブレイクをパーソナライズしません。

次の図では、さまざまなブロックが、左から右に進行するタイムラインに沿って水平に配置されています。各ブロックは、ライブストリームのコンテンツ、パーソナライズされた広告時間枠、またはパーソナライズされていない広告時間枠が再生される時間の一部を表します。点線は、ライブストリームの現在のライブエッジを表します。に設定された表示抑制値を表すもう 1 つの点線は 00:30:00、ライブエッジの点線に関してタイムラインの 30 分後に発生します。セッションの初期化を表す 3 番目の点線は、ライブエッジの点線に関してタイムラインの前半で発生します。ライブエッジ時間と avail-suppression-value 時間の間の 30 分の期間は、avail 抑制期間を表します。広告ブレイクは、セッションの初期化前から表示抑制期間の後まで進行中です。図に示すように、avail 抑制モードが に設定され **AFTER_LIVE_EDGE**、avail 抑制値がライブエッジ 00:30:00 の後に に設定され、avail 抑制フィルポリシーが に設定され **PARTIAL_AVAIL**、セッションの初期化がライブエッジの前に行われると、MediaTailor は、avail 抑制期間後に発生する広告ブレイクをパーソナライズします。表示抑制期間の前後に進行中の広告ブレイクの場合、MediaTailor は表示抑制期間の後に発生する広告ブレイクの部分をパーソナライズしますが、表示抑制期間の前後に発生する広告ブレイクの部分はパーソナライズしません。

Avail suppression mode: **AFTER_LIVE_EDGE**

Avail suppression value: **00:30:00**

Avail suppression fill policy: **PARTIAL_AVAIL**



Example 7: 表示抑制期間の開始時に**AFTER_LIVE_EDGE**広告ブレイクが進行中の広告抑制

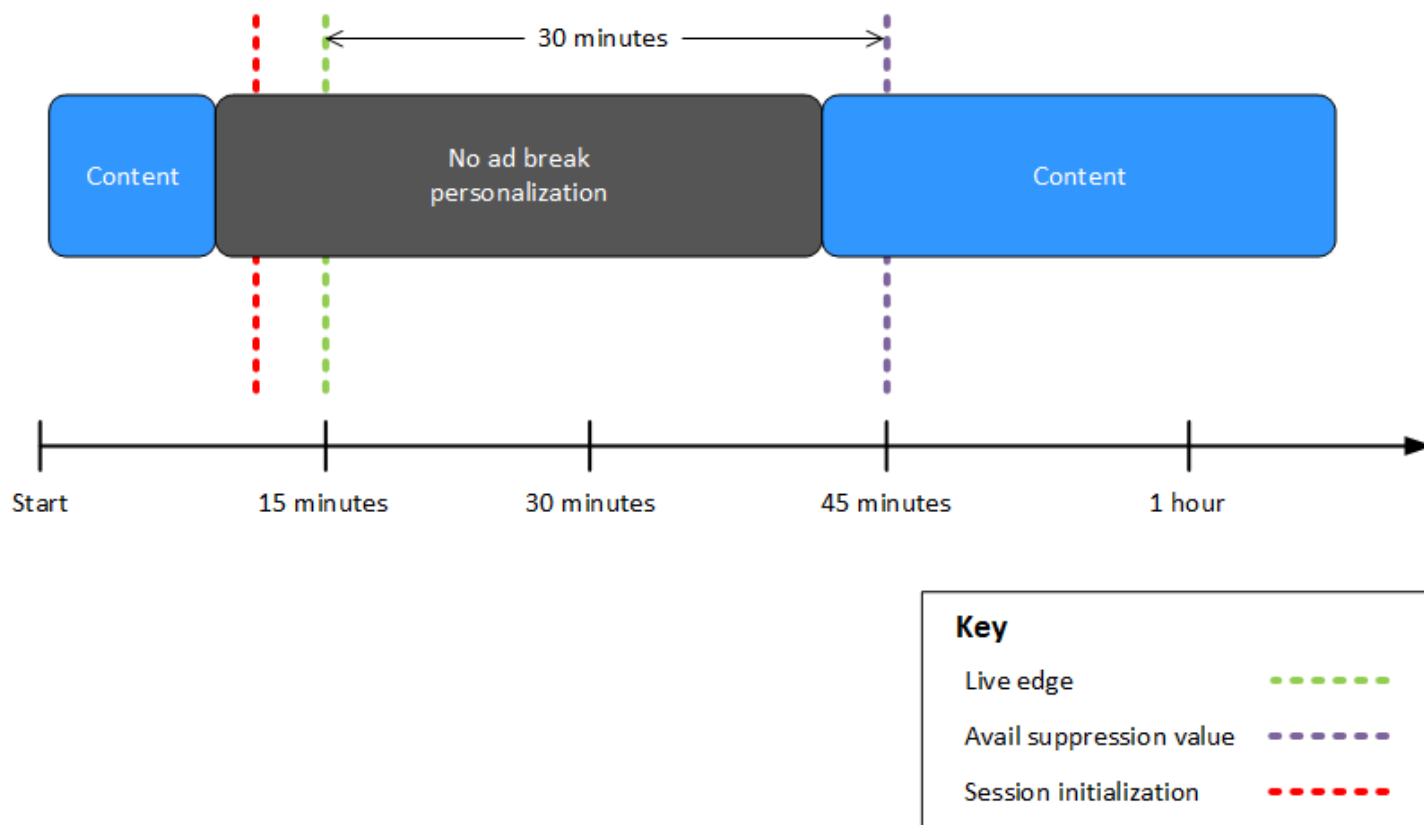
avail 抑制モードが に設定**AFTER_LIVE_EDGE**され、avail 抑制値が 0 より大きい場合、MediaTailor はセッションの経過時間がその値に達するまで広告ブレイクをパーソナライズしません。

次の図では、さまざまなブロックが、左から右に進行するタイムラインに沿って水平に配置されています。各ブロックは、ライブストリームまたはパーソナライズされていない広告ブレイクのコンテンツが再生される時間の一部を表します。点線は、ライブストリームの現在のライブエッジを表します。に設定された表示抑制値を表すもう 1 つの点線は00:30:00、ライブエッジの点線に関してタイムラインの 30 分後に発生します。セッションの初期化を表す 3 番目の点線は、ライブエッジの点線に関してタイムラインの前半で発生します。ライブエッジ時間と avail-suppression-value 時間の間の 30 分の期間は、avail 抑制期間を表します。広告時間枠は、セッションの初期化前の時点から表示抑制期間内の時点まで進行中です。図に示すように、avail サプレッションモードを に設定した場合**AFTER_LIVE_EDGE**、avail サプレッション値はライブエッ

00:30:00の後に設定され、セッションの初期化はライブエッジ時間より前ですが、広告ブレイクの開始後に行われるため、MediaTailor はその広告ブレイクをパーソナライズしません。

Avail suppression mode: **AFTER_LIVE_EDGE**

Avail suppression value: **00:30:00**



広告抑制パラメータの設定 – 再生セッションリクエスト

広告抑制は、MediaTailor に対するサーバー側またはクライアント側の最初の再生セッションリクエスト内のパラメータを使用して設定できます。MediaTailor コンソールまたは AWS Elemental MediaTailor API を介して広告抑制設定を既に設定している場合、これらのパラメータはそれらの設定を上書きします。

広告抑制を機能させるには、avail 抑制モードと avail 抑制値の両方が必要です。これらのパラメータを異なるソースから設定することはできません。例えば、MediaTailor コンソールで 1 つのパラメータを設定し、クエリパラメータで別のパラメータを設定することはできません。

MediaTailor は、以下の広告抑制パラメータをサポートします。

名前	説明	使用できる値
<code>availSuppressionMode</code>	広告抑制のモードを設定します。デフォルトでは、広告抑制は <code>OFF</code> 。に設定すると <code>BEHIND_LIVE_EDGE</code> 、MediaTailor は <code>aws.availSuppressionValue</code> 時間の経過に伴う広告時間枠を埋めません。に設定すると <code>AFTER_LIVE_EDGE</code> 、MediaTailor は表示抑制期間の前後の広告時間枠を埋めません。表示抑制期間は、ライブエッジ時間から <code>aws.availSuppressionValue</code> 時間、および追加のバッファ時間まで及びます。	• <code>OFF</code> • <code>BEHIND_LIVE_EDGE</code> • <code>AFTER_LIVE_EDGE</code>
<code>availSuppressionValue</code>	ライブストリームのライブエッジからの相対的な時間。	の UTF-8 URL エンコードされたタイムコード <code>HH:MM:SS</code> 。例えば、1 時間と 30 分は <code>01%3A30%3A00</code> になります。
<code>availSuppressionFallbackPolicy</code>	<code>avail</code> 抑制モードに適用するポリシーを定義します。 <code>BEHIND_LIVE_EDGE</code> は常に完全な <code>avail</code> 抑制ポリシーを使用します。 <code>AFTER_LIVE_EDGE</code> は、セッションが中断の途中で開始したときに部分的な広告ブレイクフィルを呼び出すために使用できます。	• <code>PARTIAL_AVAILABLE_BEFORE_LIVE_EDGE</code> 抑制モードでは使用できません • <code>FULL_AVAILABLE_ONLY_AFTER_LIVE_EDGE</code> 抑制モードのデフォルト値

サーバー側の設定

基本となるクエリパラメータは `aws.availSuppression` で、その後にオプションのパラメータ名と値のペアが続きます。クエリを作成するには、MediaTailor に対する再生セッションリクエストの最後に `aws.availSuppression=` を付加し、その後にパラメータ名と値を追加します。サーバー側の再生セッションリクエストを作成する方法の詳細については、「[MediaTailor サーバー側の広告の追跡とレポート](#)」を参照してください。

例: HLS

```
GET <mediatailorURL>/v1/master/<hashed-account-id>/<origin-id>/index.m3u8?
aws.availSuppressionMode=BEHIND_LIVE_EDGE&aws.availSuppressionValue=00%3A00%3A21
```

以下の表は、サーバー側のクエリ構文のリストです。

クエリ文字列コンポーネント	説明
?	クエリの先頭を示す制限された文字。
aws.	名前と値のペアで構成されたパラメータが続くベースクエリ。すべての使用可能なパラメータのリストについては、「 広告抑制パラメータの設定 - 再生セッションリクエスト 」を参照してください。
=	パラメータ名を値に関連付けます。例えば、 <code>aws.availSuppressionMode= BEHIND_LIVE_EDGE</code> などです。
&	クエリパラメータを連結します。例えば、 <code>aws.availSuppressionMode= BEHIND_LIVE_EDGE</code> および <code>aws.availSuppressionValue= 00:30:00&aws.availSuppressionFillPolicy= FULL_AVAIL_ONLY</code> などです。

クライアント側の設定

MediaTailor に対するクライアントの POST リクエストに `availSuppression` パラメータを含めます。クライアント側の再生セッションリクエストを作成する方法の詳細については、「[クライアント側の広告追跡](#)」を参照してください。

例: HLS

```
POST parent.m3u8
{
  "availSuppression": {
    "mode": "BEHIND_LIVE_EDGE",
    "value": "00:00:21",
    "fillPolicy": "FULL_AVAIL_ONLY"
  }
}
```

MediaTailor バンパー広告挿入

バンパーは、広告時間枠の開始時または終了前に再生される、スキップ不可能な短いビデオクリップまたはオーディオクリップです AWS Elemental MediaTailor。

バンパーには、以下の条件が適用されます。

- バンパーは 10 秒以下である必要があります。
- バンパーは、広告ブレークの開始時、広告ブレークの終了直前、またはその両方に挿入できます。
- バンパーは、プレロールが設定されている場合を除き、再生セッションのすべての広告ブレーク中に再生されます。プレロールが設定されている場合、バンパーはプレロールブレーク中に再生されませんが、プレロール後のすべての後続ブレークで再生されます。
- HLS の場合、SCTE-35 EXT-X-CUE-OUT タグのそれぞれに duration 属性を含める必要があります。
- バンパーは、ソースコンテンツに一致するようにトランスコードされます。
- バンパーに対する料金は発生しません。

バンパーの設定

バンパーを使用するには、MediaTailor コンソール、MediaTailor API、または AWS Command Line Interface () MediaTailor を使用してバンパー URLs を設定しますAWS CLI。スタートバンパー、エンドバンパー、またはその両方を設定できます。バンパーは、Amazon Simple Storage Service (Amazon S3) などのサーバーに保存されます。バンパー URL は、バンパーアセットが保存されている場所を示します。

スタートバンパーとエンドバンパーの URL 例:

スタートバンパー URL: <https://s3.amazonaws.com/startbumperad>

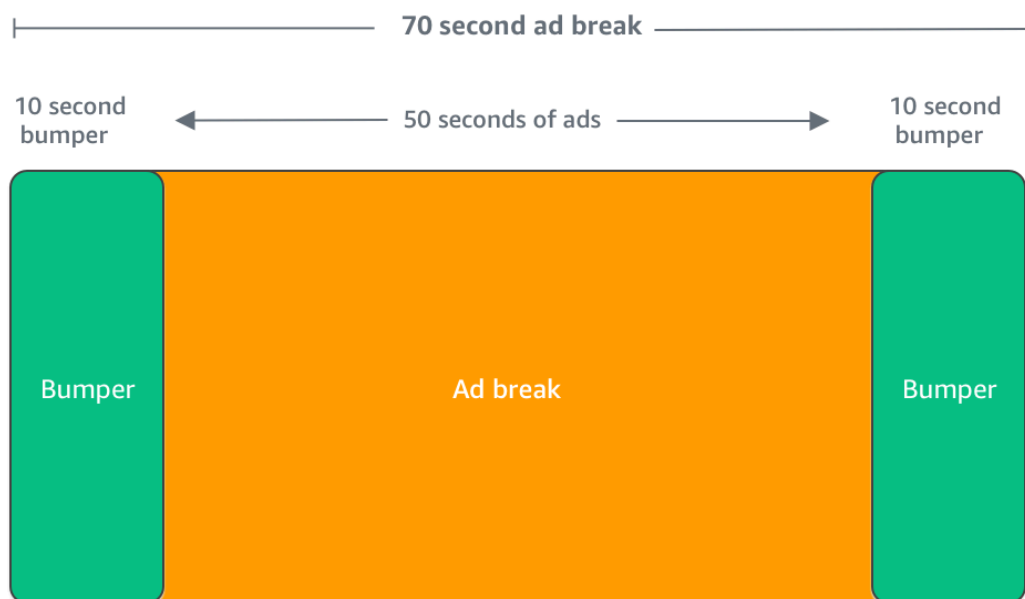
エンドバンパー URL: <https://s3.amazonaws.com/endbumperad>

例

以下は、バンパー広告動作の例です。

Example例 1: 開始バンパーと終了バンパー

この例では、スタートバンパーとエンドバンパーが有効になっています。広告決定サーバーには、70 秒の広告ブレイクを埋めるための 50 秒のパーソナライズされた広告があります。広告ブレイクの開始時に 10 秒のスタートバンパーが再生され、続いて 50 秒の広告が再生されから、10 秒のエンドバンパーが再生されます。



MediaTailor プリロール広告挿入

Note

設定可能なプレロール広告は、ライブワークフローでのみ使用できます。VOD での広告挿入 (プリロールを含む) の仕組みの詳細については、「」を参照してください[VOD に対する広告 ステッチ動作](#)。

AWS Elemental MediaTailor は、メインコンテンツを開始する前に、再生セッションの開始時に広告を挿入できます。これらがプレロール広告です。

プレロール広告を挿入するには、[オプションの設定](#)の説明にあるように、設定の [Additional (追加) 設定] で [Live pre-roll ad decision server] (ライブプレロール広告決定サーバー) と [Live pre-roll maximum allowed duration] (ライブプレロールの最大許容継続時間) の各フィールドに入力します。

- MediaTailor は、再生リクエストを受信すると、MediaTailor 再生設定の次のフィールドに基づいて、プリロール広告のリクエストを ADS に送信します。
 - [Live pre-roll ad decision server] (ライブプレロール広告決定サーバー) は、MediaTailor がプレロール広告のリクエストを送信する広告決定サーバー (ADS) の URL です。
 - [Live pre-roll maximum allowed duration] (ライブプレロールの最大許容継続時間) は、プレロール広告の最大合計時間です。MediaTailor は、最大許容継続時間に基づいて以下のアクションを実行します。
 - ADS レスポンス内の広告の合計継続時間が [Live pre-roll maximum allowed duration] (ライブプレロールの最大許容継続時間) で指定された値よりも短い場合、MediaTailor はすべての広告を挿入します。最後の広告の再生が終了すると、MediaTailor は直ちに基盤となるコンテンツに戻ります。
 - ADS レスポンス内の広告の合計継続時間が [Live pre-roll maximum allowed duration] (ライブプレロールの最大許容継続時間) で指定された値よりも長い場合、MediaTailor は、その継続時間を超過することなく時間内に収まる一連の広告を挿入します。MediaTailor は、これらの広告をトリミングしたりカットしたりすることなく挿入します。MediaTailor は、選択された最後の広告の再生が終了すると、基盤となるコンテンツに戻ります。
- MediaTailor が ADS からプレロールレスポンスを受信すると、マニフェストを操作してプレロール広告へのリンクを追加します。MediaTailor は、プレロール広告ブレイクの開始時間を以下のよう

- DASH の場合、式は $(publishTime - availabilityStartTime) - \max(suggestedPresentationDelay, minBufferTime)$ です。
 - HLS の場合、式は $\max(2 * EXT-X-TARGETDURATION, EXT-X-START:TIMEOFFSET)$ です。
3. プレロールではない広告ブレイクに対して実行するアクションは、MediaTailor が決定します。プレロールが他の広告ブレイクと重複する場合、MediaTailor は広告ブレイクの重複部分をパーソナライズしません。

MediaTailor スレート広告挿入

Note

Slate は、ライブワークフローと live-to-VOD ワークフローでのみ使用できます。

を使用すると AWS Elemental MediaTailor、広告時間枠のスレート広告を指定できます。スレートは、ストリームに挿入されるデフォルトの MP4 アセット (静止画像やループビデオなど) で、ライブコンテンツの代わりに再生されます。

AWS Elemental MediaTailor は、以下の特定の状況で広告時間枠中にスレートを表示します。

- パーソナライズされた広告に置き換えられていない休憩の残り時間を埋めるには
- 広告決定サーバー (ADS) が空の VAST または VMAP レスポンスで応答する場合
- ADS タイムアウトや ADS からの HTTP 500 エラーなどのエラー条件の場合
- MediaTailor が広告を挿入できない場合 (トランスコードが完了していない場合など)

スレートを設定しない場合、MediaTailor は上記の条件のいずれかが満たされると、デフォルトで基盤となるコンテンツストリームになります。

スレートの設定

スレートは、[MediaTailor コンソール](#)の [additional configuration] (追加の設定) ペインで指定します。MediaTailor は、ユーザー指定の URL からスレートをダウンロードして、それをコンテンツと同じレンディションにトランスコードします。スレートが表示される最大時間は、MediaTailor コンソールの [personalization threshold] (パーソナライゼーションのしきい値) 設定オプションを使用して制御できます。詳細については、「[the section called “パーソナライゼーションしきい値の仕組み”](#)」を参照してください。

スレートは、音声と動画の両方が含まれた高品質 MP4 アセットである必要があります。スレートの設定は VPAID 以外の設定ではオプションですが、VPAID ワークフローでは必須です。

Note

スレートをホストするサーバーが HTTPS を使用している場合、その証明書は既知の認証機関からのものであることが必要です。自己署名証明書にすることはできません。自己署名証明書を使用する場合、AWS Elemental MediaTailor はコンテンツオリジンからマニフェストにスレートを取得してステッチすることはできません。

パーソナライゼーションしきい値の仕組み

パーソナライゼーションしきい値は、広告時間枠で許容される不足広告時間 (秒単位) の最大期間を定義します。この機能は、基盤となるコンテンツストリームに依存するため、広告挿入ではなくライブストリームと VOD ストリームの広告置換に特に適用されます。

動作は 3 つのシナリオによって異なります。

1. パーソナライゼーションが無効になっている場合:
 - a. スレートは、未入力時間の全期間挿入されます。
 - b. 開始/終了バンパーは設定時に挿入されます (詳細については、「」を参照してください [MediaTailor バンパー広告挿入](#))。
 - c. 広告は通常どおり挿入されます
2. パーソナライゼーションが有効で、しきい値がブレイク期間より短い場合:
 - a. 未入力時間がパーソナライゼーションのしきい値を超えた場合:
 - i. MediaTailor が広告時間枠のパーソナライゼーションを中止する
 - ii. 基盤となるコンテンツが表示されます
 - iii. 広告、スレート、バンパーが挿入されない
 - b. 未入力時間がパーソナライゼーションのしきい値より短い場合:
 - i. 広告とスレートが挿入される
 - ii. バンパーは設定されている場合に挿入されます
3. パーソナライゼーションが有効で、しきい値がブレイク期間より大きい場合:
 - a. 広告が挿入されます
 - b. スレートは広告時間枠の残り時間の間挿入されます

c. 設定されている場合、バンパーは挿入されます

次の表は、HLS プロトコルと DASH プロトコルの両方における特定のシナリオの詳細な動作を示しています。

詳細な動作マトリックス

シナリオ	パーソナライゼーションが無効	パーソナライゼーションがブレイク期間未満で有効	パーソナライゼーションがブレイク期間よりも長い時間有効
空の VAST または VMAP	スレートが挿入されました	スレートが挿入されていない	スレートが挿入されました
ADS タイムアウト	スレートが挿入されました	スレートが挿入されていない	スレートが挿入されました
広告は利用できません (Vast 404)	スレートが挿入されました	スレートが挿入されていない	スレートが挿入されました
広告の期間が広告時間枠よりも長い	挿入された広告	広告が挿入されていない	挿入された広告
広告置換で完全に使用されていない時間を入力する	スレートが挿入されました	パーソナライゼーションのしきい値が未入力時間より大きい場合: 広告/スレートは挿入されず、それ以外の場合: 広告とスレートは挿入されます	スレートが挿入されました
空の VAST でプリロールする	スレート挿入、プリロール挿入なし	スレートは挿入されず、プリ	スレート挿入、プリロール挿入なし

シナリオ	パーソナライゼーションが無効	パーソナライゼーションがブレイク期間未満で有効	パーソナライゼーションがブレイク期間よりも長い時間有効
		ロールは挿入されません	
ADS タイムアウトによるプリロール	スレート挿入、プリロール挿入なし	スレートは挿入されず、プリロールは挿入されません	スレート挿入、プリロール挿入なし
広告が利用できない場合のプリロール (Vast 404)	スレート挿入、プリロール挿入なし	スレートは挿入されず、プリロールは挿入されません	スレート挿入、プリロール挿入なし
VAST が空のバンパー	スレート挿入、バンパー挿入	スレートは挿入されず、バンパーは挿入されません	スレート挿入、バンパー挿入
ADS タイムアウトのバンパー	スレート挿入、バンパー挿入	スレートは挿入されず、バンパーは挿入されません	スレート挿入、バンパー挿入
広告が利用できない場合のバンパー (Vast 404)	スレート挿入、バンパー挿入	スレートは挿入されず、バンパーは挿入されません	スレート挿入、バンパー挿入

シナリオ	パーソナライゼーションが無効	パーソナライゼーションがブレイク期間未満で有効	パーソナライゼーションがブレイク期間よりも長い時間有効
広告置換で完全に使用されていない時間を入力するバンパー	スレート挿入、バンパー挿入、広告挿入	パーソナライゼーションしきい値が未入力時間より大きい場合: bumpers/ads/slidesは挿入されず、それ以外の場合: bumpers/ads/slides挿入されません	スレート挿入、バンパー挿入、広告挿入

重要な考慮事項

スレートとパーソナライゼーションのしきい値を使用する場合は、次の点に注意してください。

- この動作は、HLS プロトコルと DASH プロトコルの両方で一貫しています。
- パーソナライゼーションしきい値機能は、スレートが設定されている場合にのみ適用されます。
- スレートが設定されていて、パーソナライゼーションのしきい値が設定されていない場合、スレートは完全なオリジン表示期間または広告の入力後の残り時間のいずれかで再生されます。
- VPAID 広告の場合、MediaTailor は VPAID 広告の期間中スレートを挿入し、ビデオプレーヤーが挿入する広告のスペースを保持します。

スレート設定と VPAID

Important

VPAID 広告を使用する場合、スレート設定は必須です。MediaTailor はスレートを挿入して、ビデオプレーヤーが挿入する VPAID 広告のスペースを保持します。ユーザーのインタラクティブ性に対応するため、スレート期間は VPAID 広告期間よりもわずかに長くなる場合があります。

次に、ビデオプレーヤーは、「」で説明されているように、MediaTailor が返すクライアント側のレポートメタデータに基づいて VPAID 広告を処理します [VPAID 要件](#)。クライアント側のレポートについては、「[the section called “クライアント側の追跡”](#)」を参照してください。

での広告トランスコーディング AWS Elemental MediaTailor

AWS Elemental MediaTailor は、オリジンビデオストリームの形式と仕様に合わせて広告、バンパー、スレートをトランスコードし、広告時間枠中にシームレスに再生できるようにします。MediaTailor は、動的トランスコード (デフォルト) とカスタムトランスコードプロファイルの 2 つのトランスコードアプローチをサポートしています。これらのアプローチの違いを理解することは、ワークフローに適した設定を決定するのに役立ちます。

トピック

- [ダイナミックトランスコーディング](#)
- [カスタムトランスコードプロファイル](#)

ダイナミックトランスコーディング

動的トランスコーディングは、MediaTailor のデフォルトの広告トランスコーディング動作です。広告、バンパー、またはスレートをトランスコードする必要がある場合、MediaTailor は実行時にオリジンマニフェストを自動的に検査し、ユーザー側の設定なしでトランスコード設定を取得します。

動的トランスコーディングが処理するもの

動的トランスコーディングは、HLSおよび DASH ワークフローに対して以下を自動的に処理します。

- **標準コーデックマッチング** – MediaTailor は、オリジンストリームのコーデックプロファイルと一致します。コーデックレベルのマッチングは保証されていないことに注意してください。
- **オーディオ正規化** – MediaTailor は、デフォルトのターゲットレベルを使用して、トランスコードされた広告にオーディオ正規化を自動的に適用します -24 LKFS。これにより、広告オーディオレベルは、追加の設定なしでオリジンコンテンツと一致します。ワークフローでデフォルトの 以外のカスタムオーディオ正規化ターゲットが必要な場合は -24 LKFS、カスタムトランスコードプロファイルが必要です。
- **標準セグメント長マッチング** – デフォルトでは、MediaTailor は 2 秒のセグメントを使用して広告をトランスコードします。プレーヤーがセグメントの長さの一致を必要とする場合は、

calculateSegmentLengthフラグを有効にできます。これにより、MediaTailor はオリジンから最初の 10 個のセグメントを分析し、セグメントの長さとして最も一般的な期間を選択することで、正しいセグメントの長さを自動的に計算します。このフラグはデフォルトで無効になっており、AWS サポートで有効にする必要があります。AWS サポートに連絡して、アカウントまたは特定の設定calculateSegmentLengthの のアクティベーションをリクエストしてください。

- 標準WebVTT字幕 – HLS ワークフローの場合、MediaTailor は、オリジンマニフェストに字幕が存在する場合、パーソナライズされたマルチバリエーションプレイリストにWebVTT字幕グループの参照を保持します。広告セグメント中、MediaTailor は字幕トラックの継続性を維持するために空白WebVTTセグメントを挿入します。これは、広告クリエイティブには通常字幕トラックが含まれないためです。DASH ワークフローの場合、MediaTailor では、オリジンマニフェストの字幕適応セットをセグメント化する必要があります (動画やオーディオなど)。MediaTailor によって構築された広告期間には、広告クリエイティブによって提供される適応セットのみが含まれます。広告クリエイティブには通常、字幕トラックが含まれないため、広告期間には字幕 は含まれません AdaptationSet。コンテンツ期間は、字幕適応セットを通常どおり保持します。
- CMAF 形式のサポート – MediaTailor は、HLSオリジンストリームが CMAF (fMP4) セグメントまたは標準TSセグメントを使用しているかどうかを検出し、それに応じて広告をトランスコードします。の場合DASH、CMAFパッケージ化は透過的にサポートされます。はDASH本質的に ISO-BMFF (fMP4) をコンテナ形式として使用します。どちらのプロトコルにも追加の設定は必要ありません。

Note

MediaTailor HLS は、TSセグメントを含むコーデックとして HVC1 (HEVC) を使用するストリームをサポートしていません。Apple デバイスの HLS オーサリング仕様のセクション 1.5 では、HEVCビデオのコンテナ形式は である必要がありますfMP4。したがってHLS、を使用するストリームHEVCはフラグメント化された MP4 セグメントCMAFで を使用する必要があります。

- トリックプレイパーソナライゼーション – MediaTailor は、トリックプレイバリエーションと関連するイメージストリーム (I フレームマニフェスト、サムネイル、イメージストリームを含む) を使用してHLSDASH、オリジンコンテンツの仕様に一致する広告を自動的にトランスコードします。この機能では、カスタムトランスコードプロファイルが不要になりました。
- コンパクトDASHマニフェスト – MediaTailor はコンパクトマニフェストを自動的にサポートします。これにより、SegmentTemplate要素を個々のRepresentation要素からAdaptationSetレベルに引き上げ、全体的なマニフェストサイズを減らし、配信効率を向上させます。この機能では、カスタムトランスコードプロファイルが不要になりました。

動的トランスコードでは不十分な場合

動的トランスコーディングは、ほとんどの広告トランスコーディングシナリオを処理します。ただし、追加の設定やカスタムトランスコードプロファイルが必要になる場合があります。

- ビットレートマッチング – 動的トランスコーディングは、デフォルトで正確なビットレートマッチングを保証するものではありません。ワークフローで広告ビットレートをオリジンビットレートと密接に一致させる必要がある場合は、`enableBitrateRounding` フラグを有効にできます。これにより、最大 5% のビットレートの四捨五入でオリジン値をより適切に一致させることができます。このフラグは AWS サポートで有効にする必要があります。AWS サポートに連絡して、アカウントまたは特定の設定 `enableBitrateRounding` の アクティベーションをリクエストしてください。ワークフローで 5% を超えるビットレートマッチングが必要な場合は、カスタムトランスコードプロファイルが必要です。
- MediaTailor のデフォルト 以外のカスタムオーディオ正規化ターゲット -24 LKFS
- 整数以外のセグメントの長さ (1.92 または 3.84 など)
- Dolby Vision オーディオ
- ダウンストリームシステムまたはプレイヤーに必要な特定の PMT、ビデオ、またはオーディオ PID 値
- カスタムオーディオサンプルレート (44.1 kHz など)
- カスタムピクセルアスペクト比 (PAR) – 例えば、平方ピクセルの場合は 1:1、D1/DV NTSC ワイドスクリーンの場合は 40:33
- 特定の名前付きオーディオトラックまたは字幕トラック

カスタムトランスコードプロファイル

カスタムトランスコードプロファイル (CTP) は、MediaTailor に登録する明示的に作成された AWS Elemental MediaConvert ジョブ設定です。CTP が MediaTailor 設定に関連付けられている場合、MediaTailor は定義された設定を使用してすべての広告、バンパー、スレートをトランスコードし、コーデック、ビットレート、オーディオ正規化レベル、セグメント長、PID 値、サンプルレートなどを正確に制御できます。CTP は動的トランスコーディングを完全に上書きするため、トリックプレイ機能 (I フレームマニフェスト、サムネイル、イメージストリーム) など、必要なすべての機能をプロファイル内で明示的に設定する必要があります。

CTP は、特定のオリジン設定と一致するように設計されています。ソースコンテンツが CTP が最初に作成された設定と異なる場合、`NO_VARIANT_MATCH` エラーにより広告が適切にトランスコード

されない可能性があります。これは、CTP 出力がオリジンマニフェストに存在するバリエーションと一致しない場合に発生します。これは、ライブワークフローとVODワークフローの両方に適用されます。例えば、異なるソースからのコンテンツを含むVODライブラリや、異なるエンコードプロファイル (さまざまな解像度、ビットレート、コーデック、セグメント長など) でエンコードされたライブラリでは、CTP がライブラリに存在するすべてのバリエーションの組み合わせを考慮していない場合にNO_VARIANT_MATCHエラーが発生することがあります。複数のオリジンをサポートする単一のCTP または混合エンコーディングプロファイルを持つVODライブラリが必要な場合は、すべてのターゲットオリジンのバリエーションを含むスーパーセット CTP を作成できます。ただし、スーパーセット CTPs には 36 個のビデオバリエーションのハード制限が適用されます。これにより、1 つのプロファイルで対応できるオリジンまたはバリエーションの組み合わせの数が制限されます。

Important

エンコーダーの設定を変更したり、新しいVODコンテンツのソースを引き受けたりするときは、常に CTP を同期させる必要があります。そうしないと、NO_VARIANT_MATCHエラーが発生し、広告トランスコードが失敗します。これには、解像度の追加や削除、ビットレートの調整、コーデックの変更、セグメントの長さの変更、異なるエンコードプロファイルを持つ新しいプロバイダーからのVODコンテンツのオンボーディングなどの変更が含まれます。

CTP を更新するときは、次の動作に注意してください。

- 既存の CTP を更新すると、新しくトランスコードされた広告にのみ新しい設定が適用されます。以前に古い CTP 設定でトランスコードされた広告は再トランスコードされません。
- 新しい CTP を作成して MediaTailor 設定に関連付けると、MediaTailor が新しいプロファイルを使用してすべての広告を再トランスコードするように求められ、広告インベントリ全体で完全な整合性が確保されます。

このため、エンコーダーの設定が変更されたり、新しいコンテンツソースをオンボードしたりする場合は、ベストプラクティスとして、既存の CTP を更新するのではなく、新しい CTP を作成して、以前にトランスコードされた広告を含むすべての広告に更新された設定が反映されるようにします。

カスタムトランスコードプロファイルを使用するタイミング

ワークフローに以下が 1 つ以上必要な場合は、カスタムトランスコードプロファイルを使用します。

- MediaTailor のデフォルト 以外のカスタムオーディオ正規化ターゲット -24 LKFS

- 整数以外のセグメントの長さ (1.92 や 3.84 など)
- Dolby Vision オーディオ設定
- ダウンストリームシステムまたはプレイヤーに必要な特定の PMT、ビデオ、またはオーディオ PID 値
- カスタムオーディオサンプルレート (44.1 kHz など)
- 設定フラグを使用した動的トランスコーディングでサポートされる 5% のしきい値を超えるビットレートマッチング
- カスタムピクセルアスペクト比 (PAR) – 例えば、平方ピクセルの場合は 1:1、D1/DV NTSC ワイドスクリーンの場合は 40:33
- 特定の名前付きオーディオまたは字幕トラック

カスタムトランスコードプロファイルの開始方法

CTP の登録に必要な MediaTailor Transcoding API へのアクセスは、許可リストに登録されたアカウントに制限されます。開始するには、次のいずれかのオプションを使用して AWS サポートにお問い合わせください。

- 許可リストアクセスのリクエスト – 独自の CTPs を作成および管理するには、AWS サポートに連絡して、アカウントを許可リストに追加するようリクエストしてください。アカウントが許可リストに追加されると、Transcoding API にアクセスして、既存の MediaTailor 設定がある任意の AWS リージョンでプロファイルを登録および管理できます。
- AWS サポートによる CTP 作成のリクエスト – AWS サポートがユーザーに代わって CTP を生成して登録するには、MediaTailor がアクセスできる non-DRM-protected オリジンエンドポイント URL を使用して AWS サポートにお問い合わせください。AWS サポートはこのエンドポイントを使用してオリジン設定を分析し、アカウントに適した CTP を生成します。

動的トランスコードとカスタムトランスコードプロファイル

次の表を使用して、ワークフローに適したアプローチを決定します。

動的トランスコードとカスタムトランスコードプロファイルの比較

考慮事項	ダイナミックトランスコーディング	カスタムトランスコードプロファイル
セットアップの労力	なし	手動で作成された AWS Elemental MediaConvert ジョブ設定
Trickplay サポート (HLS および DASH)	自動	CTP で明示的に設定する必要があります
コンパクトマDASHニフェスト	自動	CTP で明示的に設定する必要があります
CMAF 形式のサポート	自動	CTP で明示的に設定する必要があります
オリジンの変更処理	自動	手動 CTP 更新が必要
マルチリージョンの複雑さ	なし	リージョンごとに登録する必要があります
オーディオ正規化	-24 LKFS (デフォルト)	完全に設定可能
ビットレートマッチング	enableBitrateRounding フラグによる最大 5% の四捨五入 (AWS サポートに連絡して有効にしてください)	オリジンビットレートと一致するように CTP で明示的に設定
非整数セグメントの長さ	2 秒のデフォルト。非整数セグメント長はサポートされていません (AWS サポートに連絡して、整数セグメント長の自動計算 calculateSegmentLength を有効にします)	オリジンと一致するようにセグメントの長さを手動で設定する必要があります
カスタム PID 値	サポート外	サポート

考慮事項	ダイナミックトランスコーディング	カスタムトランスコードプロファイル
Dolby Vision オーディオ	サポート外	サポート
カスタムオーディオサンプルレート	サポート外	サポート
マルチオリジンのサポート	自動	スーパーセット CTP が必要。 ビデオバリエーションは 36 個に制限
MediaTailor の改善	自動	手動 CTP 更新が必要
以下に最適:	標準コーデックマッチング、 一般的なビットレートプロファイル、トリックプレイ、 コンパクト DASH	動的トランスコーディング では満たされない特殊なエンコーディング要件

広告のプリフェッチ

ライブストリームの AWS Elemental MediaTailor 広告プリフェッチを使用すると、広告決定サーバー (ADS) のピーク負荷を軽減し、各広告ブレイクの開始時にマニフェスト配信レイテンシーを短縮できます。プリフェッチスケジュールを定義すると、MediaTailor はスケジュールに従って ADS から広告を取得し、広告時間枠に必要な前に広告挿入の準備をします。ライブストリーム中、プリフェッチは、広告リクエストやトランスコーディングのタイムアウトやその他のネットワーク遅延による広告フィル率の低下や収益機会の損失を軽減するのに役立ちます。

Note

広告プリフェッチは、従来のサーバーガイドおよび HLS インタースティシャルを含むサーバーガイド広告挿入 (SGAI) メソッドでは機能しません。SGAI メソッドでは、プレイヤーが再生する広告のみを取得するため、プリフェッチは必要ありません。また、MediaTailor が個々のセッションリクエストを表示せずに CDNs でマニフェストを提供できます。

広告のプリフェッチをセットアップするには、再生設定で 1 つ、または複数のプリフェッチスケジュールを作成します。プリフェッチスケジュールは、今後の広告ブレイクのための広告をいつ、どのように取得して準備するのかを MediaTailor に伝えます。

- イベントに予測可能なスケジュールの広告表示がある場合は、1 つのプリフェッチスケジュールを使用します。1 つのプリフェッチスケジュールごとに、MediaTailor が 1 つの広告表示に配置する 1 つの広告セットを定義します。1 つのプリフェッチスケジュールを使用するときに複数の広告表示の広告をプリフェッチするには、各広告表示に関連する複数のプリフェッチスケジュール (広告表示の 24 時間前まで) を作成する必要があります。
- イベントに予測可能なスケジュールにない広告表示がある場合は、定期的なプリフェッチスケジュールを使用します。定期的なプリフェッチスケジュールは、スケジュールを自動的に作成し、イベントの各広告時間枠の前に広告をプリフェッチします。定期的なプリフェッチスケジュールは、定義された期間 (イベントが終了する最大 24 時間前) 内のすべての広告表示の広告を取得します。広告表示ごとにスケジュールを作成する必要はありませんが、単一のプリフェッチが提供するタイミング制御の一部が失われることはありません。

以下のトピックでは、広告プリフェッチについて詳しく説明します。

トピック

- [プリフェッチの仕組み](#)
- [プリフェッチスケジュールの作成](#)
- [TPS ベースのトラフィックシェーピング](#)
- [プリフェッチスケジュールの削除](#)

プリフェッチの仕組み

クライアントが MediaTailor に対してマニフェストリクエストを行うと、サービスが再生設定に関連付けられているすべてのプリフェッチスケジュールを評価します。MediaTailor に一致するプリフェッチスケジュールがない場合、サービスは通常の広告挿入に戻り、広告をプリフェッチしません。

MediaTailor が一致するプリフェッチスケジュールを検出すると、サービスは取得と消費の 2 つのコンポーネントに基づいてスケジュールを評価します。各コンポーネントの設定は、以下のセクションで説明するように、単一のプリフェッチスケジュールと定期的なプリフェッチスケジュールによって異なります。

単一のプリフェッチスケジュールフロー

取得

これは、MediaTailor が ADS から広告をプリフェッチする時間範囲である取得時間枠を定義します。このウィンドウは、必ず広告時間枠の前にスケジュールしてください。以下は、MediaTailor が単一のプリフェッチスケジュールを処理する方法の概要です。

コンソールで単一のプリフェッチスケジュールを作成する手順については、「」を参照してください [プリフェッチスケジュールの作成](#)。API の手順については、API AWS Elemental MediaTailor リファレンスの [PrefetchSchedules](#)」を参照してください。

指定された取得ウィンドウ中に、MediaTailor は ADS にリクエストを送信して、後で再生セッションに挿入するための広告を取得して準備します。

- MediaTailor は、オプションでトラフィックシェーピングを使用して、一度に ADS へのリクエストの数を制限します。次の 2 つのアプローチから選択できます。

タイムウィンドウトラフィックシェーピング - MediaTailor は、すべてのセッションのリクエストを一度に送信するのではなく、指定された秒数にわたってリクエストを分散します。この分散トラフィック分散は、ADS が過負荷になり、タイムアウトが発生し、広告フィルレートが低下するのを防ぐのに役立ちます。

TPS ベースのトラフィックシェーピング - MediaTailor は、1 秒あたりのトランザクション (TPS) と同時ユーザーに基づいてリクエストを制限します。このアプローチは、ADS 容量制限に基づいてより直感的な設定を提供します。詳細については、「[TPS ベースのトラフィックシェーピング](#)」を参照してください。

- 動的変数を設定すると、MediaTailor は ADS へのリクエストにこれらの変数を含めます。MediaTailor は、これらの変数を使用して広告表示を照合し、消費ウィンドウ中にスケジュールをプリフェッチします。詳細については、次の消費セクションを参照してください。

Example

ライブイベントは午前 7 時 45 分から 10AM 時まで続き、広告時間枠は午前 8 時 15 分です。MediaTailor は、午前 7 時 45 分から 8AM 時までの広告を取得するように設定し、トラフィックシェーピングウィンドウは 60 秒です。同時ユーザー数が 500,000 人の場合、MediaTailor は ADS リクエストを分散して、すべてのリクエストを同時に送信するのではなく、平均 1 秒あたり約 8,333 件のトランザクションを 60 秒間 ($500,000 \text{ ユーザー} / 60 \text{ 秒} = 1 \text{ 秒あたり } 8,333 \text{ 件のリクエスト}$) 達成します。

取得設定には、動的変数キー `scte.event` と値が含まれます 1234。MediaTailor は ADS へのリクエストにこの変数を含め、特定のアドバタイザーをイベント ID 1234 にターゲットにするために使用できます。

消費

MediaTailor は、消費ウィンドウ中に SCTE-35 広告ブレイクマーカを検出すると、プリフェッチされた広告を広告ブレイクに配置します。

- 表示一致基準を設定しなかった場合、MediaTailor は消費ウィンドウの最初の時間枠に広告を挿入します。
- 可用性マッチング基準に動的変数キーを設定した場合、MediaTailor は取得ウィンドウで設定した動的変数に対してこれらの基準を評価します。広告ブレイクは、表示マッチング基準を満たしている場合にのみ、プリフェッチされた広告挿入の対象となります。MediaTailor は、条件を満たす最初の時間枠に広告を挿入します。

サポートされている avail-matching 条件のリストについては、の表の「広告プリフェッチに使用可能」列を参照してください [ADS リクエストの MediaTailor セッション変数](#)。

Example continued

消費の開始時刻を午前 8 時 15 分に設定し、終了時刻を午前 8 時 17 分に設定します。
キー `scte.event_id` の を有効一致条件に含めます。

MediaTailor が午前 8 時 15 分から午前 8 時 17 分まで遭遇する広告時間枠ごとに、各広告時間枠の SCTE イベント ID を評価します。各再生セッションで、MediaTailor はイベント ID が 1234 (取得動的変数で定義) の最初の広告ブレイクにプリフェッチされた広告を挿入します。正しいイベント ID が含まれていない広告時間枠の場合、MediaTailor は標準広告挿入を実行します。

定期的なプリフェッチスケジュールフロー

取得

これは、MediaTailor がライブイベントの広告をプリフェッチして挿入する時間範囲 (最大 24 時間) である定期的な取得ウィンドウを定義します。以下は、MediaTailor が定期的なプリフェッチスケジュールを処理する方法の概要です。

コンソールで定期的なプリフェッチスケジュールを作成する手順については、「」を参照してください [プリフェッチスケジュールの作成](#)。API の手順については、API AWS Elemental MediaTailor リファレンスの [PrefetchSchedules](#)」を参照してください。

指定された定期的なプリフェッチウィンドウ中に、MediaTailor は最大 24 時間のライブイベントの広告を取得して挿入します。ウィンドウ内の広告時間枠ごとに、MediaTailor は次の広告時間枠の広告を自動的に取得します。

- 表示終了後に遅延を設定すると、MediaTailor は次の広告時間枠の次の広告セットを取得する前に、指定された時間待機します。
- MediaTailor は、オプションでトラフィックシェーピングを使用して、一度に ADS へのリクエストの数を制限します。次の 2 つのアプローチから選択できます。

タイムウィンドウトラフィックシェーピング - MediaTailor は、すべてのセッションのリクエストを一度に送信するのではなく、指定された秒数にリクエストを分散します。この分散トラフィック分散は、ADS が過負荷になり、タイムアウトが発生し、広告フィルレートが低下するのを防ぐのに役立ちます。

TPS ベースのトラフィックシェーピング - MediaTailor は、1 秒あたりのトランザクション (TPS) と同時ユーザーに基づいてリクエストを制限します。このアプローチは、ADS 容量制限に基づいてより直感的な設定を提供します。詳細については、「[TPS ベースのトラフィックシェーピング](#)」を参照してください。

- 動的変数を設定すると、MediaTailor は ADS へのリクエストにこれらの変数を含めます。MediaTailor は、これらの変数を使用して広告表示を照合し、消費ウィンドウ中にスケジュールをプリフェッチします。詳細については、次の消費セクションを参照してください。

Example

ライブイベントは 7PM 時から午後 8 時 45 分まで続き、その間に 4 回の広告時間枠があります。広告時間枠は予測可能なスケジュールではありません。定期的なプリフェッチを 7PM 時から午後 8 時 45 分まで設定し、10 分の遅延と 60 秒のトラフィックシェーピングウィンドウを設定します。各表示の後、MediaTailor は次の広告時間枠の広告を取得します。表示が終了してから 10 分後、MediaTailor は ADS への取得リクエストの送信を開始します。トラフィックシェーピングウィンドウが 60 秒、同時ユーザー数が 500,000 人の場合、MediaTailor は ADS リクエストを分散して、すべてのリクエストを同時に送信するのではなく、1 秒あたり約 8,333 トランザクション ($500,000 \text{ ユーザー} / 60 \text{ 秒} = 1 \text{ 秒あたり } 8,333 \text{ リクエスト}$) の平均レートを実現します。

取得設定には、動的変数キー `scte.event` と値 が含まれます 1234。MediaTailor は ADS へのリクエストにこの変数を含め、特定のアドバタイザーをイベント ID 1234 にターゲットにするために使用できます。

消費

MediaTailor は SCTE-35 広告ブレイクマーカを検出すると、プリフェッチされた広告を広告ブレイクに配置します。

- 取得した広告の有効期限を設定した場合、プリフェッチされた広告は、指定された有効期限まで挿入できます。
- 表示一致基準を設定しなかった場合、MediaTailor は消費ウィンドウの最初の時間枠に広告を挿入します。
- 可用性マッチング基準に動的変数キーを設定した場合、MediaTailor は取得ウィンドウで設定した動的変数に対してこれらの基準を評価します。広告ブレイクは、表示マッチング基準を満たしている場合にのみ、プリフェッチされた広告挿入の対象となります。MediaTailor は、条件を満たす最初の時間枠に広告を挿入します。

サポートされている avail-matching 条件のリストについては、の表の「広告プリフェッチに使用可能」列を参照してください[ADS リクエストの MediaTailor セッション変数](#)。

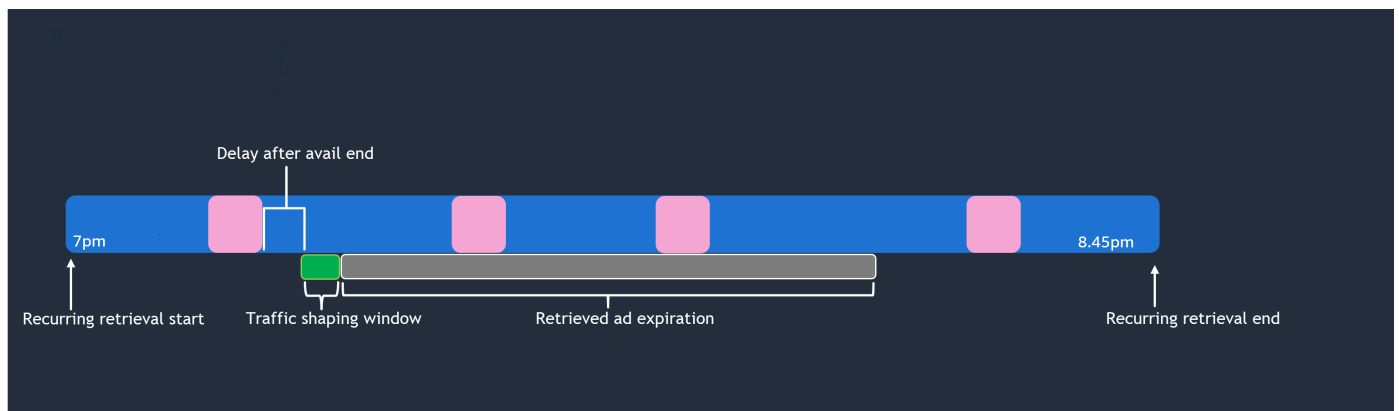
Example continued

消費では、キー `scte.event_id` の を有効一致条件に含めます。

MediaTailor が遭遇する広告時間枠ごとに、各広告時間枠の SCTE イベント ID を評価します。各再生セッションで、MediaTailor はイベント ID が 1234 (取得動的変数で定義) の各広告ブレイクにプリフェッチされた広告を挿入します。正しいイベント ID が含まれていない広告時間枠の場合、MediaTailor は標準広告挿入を実行します。

取得した広告を 45 分間挿入できるように、広告の有効期限を 2700 秒に設定します。

次の図は、広告時間枠を表す小さな四角形の例を示しています。定期的なプリフェッチスケジュール設定は、イベントタイムラインに沿って示されています。



プリフェッチコストについて

広告取り出しリクエストの作成にコストはかかりません。ただし、プリフェッチ広告の取得には、MediaTailor がトランスコードするプリフェッチ広告の標準トランスコードレートで課金されます。プリフェッチ広告の消費については、MediaTailor が広告ブレイクに配置するプリフェッチされた広告の広告挿入に対する標準料金が請求されます。トランスコードと広告挿入のコストについては、[AWS Elemental MediaTailor 「料金表」](#)を参照してください。

プリフェッチスケジュールの作成

以下は、MediaTailor コンソールを使用してプリフェッチスケジュールを作成する方法を説明する手順です。MediaTailor API を使用してプログラムでプリフェッチスケジュールを作成および管理する方法の詳細については、AWS Elemental MediaTailor API リファレンスの[PrefetchSchedules](#)を参照してください。

Note

MediaTailor でプリフェッチスケジュールを設定するときは、さまざまなタイプの変数がどのように処理されるかを理解することが重要です。

利用可能な一致基準

スケジュールで avail 一致基準を使用する場合は、まず再生設定の ADS URL テンプレートを[動的セッション変数](#)で設定してください。そうしないと、avail 一致基準は効果がありません。動的セッション変数の操作については、「MediaTailor 広告挿入の開始方法」トピックの[ステップ 3: ADS リクエスト URL とクエリパラメータを設定する](#)「」を参照してください。

プリフェッチスケジュールのプレイヤー変数

プリフェッチスケジュールを作成するときは、プレイヤー変数をプリフェッチ設定で動的変数として定義しないでください。代わりに、セッション開始時と同じようにプレイヤー変数を渡します。変数が ADS テンプレート URL にマッピングされている場合、MediaTailor はこれらの変数をプリフェッチ広告リクエストに自動的に含めます。

コンソールを使用して新しいプリフェッチスケジュールを作成する

1. MediaTailor コンソール (<https://console.aws.amazon.com/mediatailor/>) を開きます。

2. ナビゲーションペインで [Configurations] (設定) をクリックします。プリフェッチスケジュールを作成する再生設定を選択します。
3. [Prefetch schedules] (プリフェッチのスケジュール) タブで、[Add prefetch schedule] (プリフェッチスケジュールを追加) をクリックします。
4. [Prefetch schedule details] (プリフェッチスケジュールの詳細) ペインで、以下を実行します。
 - [Name] (名前) には、プリフェッチスケジュールの識別子 (my-prefetch-schedule など) を入力します。
 - [Stream ID] (ストリーム ID) には、オプションで一意的 ID を入力します。オリジンに複数の再生ストリームが含まれている場合は、この ID を使用して、特定のストリームに広告を配置するよう MediaTailor に指示することができます。たとえば、再生設定にスポーツストリームとテレビ番組ストリームがある場合、ストリーム ID を使用してプリフェッチスケジュールを作成し、スポーツストリームを対象とした広告を挿入できます。ストリーム ID 値は、クライアントのセッション開始またはマニフェストリクエストで MediaTailor に渡します。詳細については、以下の例を参照してください。
 - サーバー側の追跡の場合は、MediaTailor エンドポイントに対するクライアントの GET HTTP リクエストに ?aws.streamId クエリパラメータと値を含めます。サーバー側の追跡に関する一般情報については、「[MediaTailor サーバー側の広告の追跡とレポート](#)」を参照してください。ストリーム ID が含まれた HLS エンドポイントに対するマニフェストリクエストは以下のようになります。*myStreamId* はストリーム ID の名前です。

```
GET <mediatailorURL>/v1/master/<hashed-account-id>/<origin-id>/<asset-id>?  
aws.streamId=myStreamId
```

- クライアント側の追跡の場合は、MediaTailor /v1/Session エンドポイントに対するクライアントの POST HTTP セッション開始リクエストボディに streamId キーと値を含めます。クライアント側の追跡に関する一般情報については、「[クライアント側の広告追跡](#)」を参照してください。ストリーム ID が含まれたセッション開始リクエストは以下のようになります。*myStreamId* はストリーム ID の名前です。

```
POST <mediatailorURL>/v1/session/<hashed-account-id>/<origin-id>/<asset-id>  
{  
  'streamId': 'myStreamId',  
  'reportingMode': 'client'  
}
```

5. プリフェッチタイプでは、選択し、対応するセクションを選択して、追加のフィールドをサポートします。

- イベントの 1 つの広告時間枠に 1 つのプリフェッチスケジュールを作成する場合は、「シングル」を選択します。
- イベントの各広告時間枠の前に広告を自動的にプリフェッチするスケジュールを作成する場合は、繰り返しを選択します。

単一のプリフェッチスケジュール

イベントで 1 つの広告表示の前に広告をプリフェッチするスケジュールを作成するには。

1. [Retrieval] (取得) ペインで、使用する取得設定を指定します。これらの設定は、MediaTailor が ADS から広告をプリフェッチするタイミングを決定します。また、ADS へのリクエストに含める動的セッション変数があれば決定します。
 - [Start time] (開始時間) には、MediaTailor がこの広告ブレイク用のプリフェッチ取得を開始できる時刻を入力します。MediaTailor は、この時刻以降にクライアントによって行われるマニフェストリクエストのための広告のプリフェッチを試みます。デフォルト値は現在の時刻です。値を指定しない場合、サービスは可能な限り速やかにプリフェッチ取得を開始します。
 - [End time] (終了時間) には、MediaTailor がこの広告ブレイク用のプリフェッチ取得を停止する時刻を入力します。MediaTailor は、この時刻以前に行われるマニフェストリクエストのための広告のプリフェッチを試みます。取得時間枠は、消費時間枠と重複させることができます。
 - 必要に応じて、一度に ADS へのリクエスト数を制限するようにトラフィックシェーピングを設定します。次のいずれかのアプローチを選択します。

タイムウィンドウアプローチ: トラフィックシェーピングウィンドウ期間には、MediaTailor が ADS にリクエストを配信する秒数を入力します。詳細については、[「シングルプリフェッチスケジュールの取得の説明」](#)を参照してください。

TPS ベースのアプローチ: ピーク TPS とピーク同時ユーザーを設定し、1 秒あたりのトラザクションと同時ユーザーに基づいてリクエストを制限します。詳細については、[「TPS ベースのトラフィックシェーピング」](#)を参照してください。

- [動的変数](#)セクションには、最大 100 個の動的セッション変数を入力します。MediaTailor は、ADS に送信するプリフェッチリクエストでの代入にこれらの変数を使用します。動的セッション変数を入力しない場合、MediaTailor は [ADS URL](#) に含まれる動的変数の値を補間するよう最善を尽くします。
 - [Add dynamic variable] (動的変数を追加) をクリックします。

- Key には、などの動的セッション変数キーを入力します `scte.event_id`。これには、MediaTailor がサポートする任意の動的変数を使用できます。動的セッション変数の詳細については、「」を参照してください [ADS リクエストの MediaTailor セッション変数](#)。
 - [Value] (値) には、*my-event* などの動的変数値を入力します。
 - 別の動的変数を追加するには、[Add dynamic variable] (動的変数を追加) をクリックします。
2. [Consumption] (消費) ペインで、消費時間枠に使用する設定を指定します。これらの設定は、MediaTailor が広告ブレイクに広告を配置するタイミングを決定します。また、使用する avail 使用条件も決定します。
- [Start time] (開始時間) には、MediaTailor がプリフェッチされた広告の広告ブレイクへの配置を開始する時刻を入力します。デフォルト値は現在の時刻です。時刻を指定しない場合、サービスは可能な限り速やかにプリフェッチ消費を開始します。
 - [End time] (終了時間) には、MediaTailor がプリフェッチされた広告の広告ブレイクへの配置を停止する時刻を入力します。MediaTailor は、この時刻以前に行われるクライアントのマニフェストリクエストのための広告のプリフェッチを試みます。終了時刻は、開始時刻の後、かつ現時点から 1 日未満の時刻にする必要があります。消費時間枠は、取得時間枠と重複させることができます。
 - [可用性一致基準](#) セクションで、可用性基準を追加を選択し、スケジュールに最大 5 つの可用性一致基準を追加します。その後、[Dynamic variable key] (動的変数キー) で、`scte.event_id` などの動的変数キーを追加します。MediaTailor は、クライアントが MediaTailor に渡す動的変数値、または MediaTailor がセッションデータなどの情報から推測する動的変数値で定義された基準を満たしている場合にのみ MediaTailor、プリフェッチされた広告を広告ブレイクに配置します。広告時間枠が指定された一致基準を満たしていない場合、MediaTailor はその時間枠のプリフェッチをスキップします。詳細については、「[Single prefetch schedule consumption explanation](#)」を参照してください。

 Important

プリフェッチスケジュールごとに最大 5 つの有効な一致基準を指定できます。複数の条件を指定すると、MediaTailor は論理 AND を使用して条件を組み合わせます。論理 OR の組み合わせをモデル化するには、複数のプリフェッチスケジュールを作成します。

広告時間枠ごとに変化する変数 (`scte.event_id`、`scte.segmentation_upid`、など `avail.index`) のみを有効な一致基準として使用します。プレイヤー変数 (など `player_params.*`) はセッション全体で一定に保たれ、広告時間枠を区別しない

ため、表示マッチング基準に使用しないでください。変数が ADS テンプレート URL にマッピングされている場合、MediaTailor はプリフェッチ広告リクエストにプレイヤー変数を自動的に含めます。

3. [Add avail criteria] (Avail 条件を追加) をクリックします。

プリフェッチスケジュールは、消費時間枠の終了時間後、自動的に期限が切れます。これらは、診断目的のために少なくとも 7 日間表示された後、MediaTailor によって自動的に削除されます。また、プリフェッチスケジュールはいつでも手動で削除できます。プリフェッチスケジュールを手動で削除する方法については、以下の「[the section called “プリフェッチスケジュールの削除”](#)」セクションを参照してください。

クライアントが CreatePrefetchSchedule API を呼び出す頻度の決定

広告ブレイクが配信される正確なタイミングをユーザーが把握している場合、クライアントは 1 日 1 回 [CreatePrefetchSchedule](#) API をプログラマ的に呼び出して、取得と消費をセットアップできます。クライアントは、一日中何度も API を呼び出して、取得と消費を定義することもできます。API コールの頻度を選択するときは、[アクティブなプリフェッチスケジュールの最大数](#)と、プリフェッチスケジュールの作成後に広告時間枠スケジュールが変更される可能性を考慮してください (複数可)。プリフェッチスケジュールの作成後に広告ブレイクスケジュールが変更される可能性が高い場合は、API をより頻繁に呼び出すことをお勧めします。

定期的なプリフェッチスケジュール

イベントの各広告表示の前に広告をプリフェッチするスケジュールを作成するには。

1. 定期的な取り出しペインで、使用する取り出し設定を指定します。これらの設定は、MediaTailor が ADS から広告をプリフェッチするタイミングを決定します。また、ADS へのリクエストに含める動的セッション変数があれば決定します。
 - 定期的なプリフェッチウィンドウには、MediaTailor がこの広告時間枠のプリフェッチ取得を開始できる時間を入力します。MediaTailor は、この時刻以降にクライアントによって行われるマニフェストリクエストのための広告のプリフェッチを試みます。デフォルト値は現在の時刻です。値を指定しない場合、サービスは可能な限り速やかにプリフェッチ取得を開始します。
 - 表示終了後の遅延には、次の表示の広告をプリフェッチする前に MediaTailor が表示終了後に待機する秒数を入力します。値を指定しない場合、MediaTailor はデフォルトで遅延なしになります。

- 必要に応じて、一度に ADS へのリクエスト数を制限するようにトラフィックシェーピングを設定します。次のいずれかのアプローチを選択します。

タイムウィンドウアプローチ: トラフィックシェーピングウィンドウ期間には、MediaTailor が ADS にリクエストを配信する秒数を入力します。詳細については、[「定期的なプリフェッチスケジュールの取得の説明」](#)を参照してください。

TPS ベースのアプローチ: ピーク TPS とピーク同時ユーザーを設定し、1 秒あたりのランザクションと同時ユーザーに基づいてリクエストを制限します。詳細については、[「TPS ベースのトラフィックシェーピング」](#)を参照してください。

- [動的変数](#) セクションには、最大 100 個の動的セッション変数を入力します。MediaTailor は、ADS に送信するプリフェッチリクエストでの代入にこれらの変数を使用します。動的セッション変数を入力しない場合、MediaTailor は [ADS URL](#) に含まれる動的変数の値を補間するよう最善を尽くします。
 - [Add dynamic variable] (動的変数を追加) をクリックします。
 - Key には、などの動的セッション変数キーを入力します `scte.event_id`。これには、MediaTailor がサポートする任意の動的変数を使用できます。動的セッション変数の詳細については、「」を参照してください [ADS リクエストの MediaTailor セッション変数](#)。
 - [Value] (値) には、*my-event* などの動的変数値を入力します。
 - 別の動的変数を追加するには、[Add dynamic variable] (動的変数を追加) をクリックします。
- 2. [Consumption] (消費) ペインで、消費時間枠に使用する設定を指定します。これらの設定は、MediaTailor が広告ブレイクに広告を配置するタイミングを決定します。また、使用する avail 使用条件も決定します。
 - 取得済み広告の有効期限については、取得済み広告を挿入できる期間を指定します。
 - [可用性一致基準](#) セクションで、可用性基準を追加を選択し、スケジュールに最大 5 つの可用性一致基準を追加します。その後、[Dynamic variable key] (動的変数キー) で、`scte.event_id` などの動的変数キーを追加します。MediaTailor は、クライアントが MediaTailor に渡す動的変数値、または MediaTailor がセッションデータなどの情報から推測する動的変数値で定義された基準を満たしている場合にのみ MediaTailor、プリフェッチされた広告を広告ブレイクに配置します。広告時間枠が指定された一致基準を満たしていない場合、MediaTailor はその時間枠のプリフェッチをスキップします。詳細については、[「定期的なプリフェッチスケジュールの消費の説明」](#)を参照してください。

⚠ Important

プリフェッチスケジュールごとに最大 5 つの有効な一致基準を指定できます。複数の条件を指定すると、MediaTailor は論理 AND を使用して条件を組み合わせます。論理 OR の組み合わせをモデル化するには、複数のプリフェッチスケジュールを作成します。

広告時間枠ごとに変化する変数 (scte.event_id、scte.segmentation_upid、など avail.index) のみを有効な一致基準として使用します。プレイヤー変数 (など player_params.*) はセッション全体で一定に保たれ、広告時間枠を区別しないため、表示マッチング基準に使用しないでください。変数が ADS テンプレート URL にマッピングされている場合、MediaTailor はプリフェッチ広告リクエストにプレイヤー変数を自動的に含めます。

3. [Add avail criteria] (Avail 条件を追加) をクリックします。

プリフェッチスケジュールは、消費時間枠の終了時間後、自動的に期限が切れます。これらは、診断目的のために少なくとも 7 日間表示された後、MediaTailor によって自動的に削除されます。また、プリフェッチスケジュールはいつでも手動で削除できます。プリフェッチスケジュールを手動で削除する方法については、以下の「[the section called “プリフェッチスケジュールの削除”](#)」セクションを参照してください。

TPS ベースのトラフィックシェーピング

AWS Elemental MediaTailor には、一度に ADS へのリクエスト数を制限するための 2 つのオプションのトラフィックシェーピングアプローチが用意されています。TPS ベースのトラフィックシェーピングは、プリフェッチスケジュールのタイムウィンドウベースのトラフィックシェーピングに代わる方法を提供します。このアプローチは、時間計算ではなく、1 秒あたりのトランザクション (TPS) と予想される同時ユーザー数の観点から広告決定サーバー (ADS) 容量を指定できるようにすることで、より直感的な設定を提供します。

TPS ベースのトラフィックシェーピングの仕組み

取得ウィンドウの期間を指定する代わりに、次のパラメータを指定します。

ピーク TPS

ADS が処理できる 1 秒あたりのリクエストの最大数。このパラメータにはデフォルト値はありません。

ピーク同時ユーザー

コンテンツの同時ビューワーの予想されるピーク数。このパラメータにはデフォルト値はありません。

MediaTailor は、同時セッションの数に関係なく、指定した TPS 制限内に収まるように、プリフェッチリクエストを一定期間にわたって自動的に分散します。

Example TPS ベースの設定例

ADS は 500 TPS を処理でき、ピーク時に 100,000 人の同時視聴者を想定します。以下を設定します。

- ピーク TPS: 500
- ピーク同時ユーザー: 100,000

MediaTailor は、同時セッションの数に関係なく、指定した TPS 制限内に収まるように、プリフェッチリクエストを一定期間にわたって自動的に分散します。

プリフェッチスケジュールの削除

以下は、MediaTailor コンソールを使用してプリフェッチスケジュールを削除する方法を説明する手順です。MediaTailor API を使用してプログラムでプリフェッチスケジュールを削除する方法については、API AWS Elemental MediaTailor リファレンスの [DeletePrefetchSchedule](#)」を参照してください。

Note

削除はリアルタイムでは行われません。MediaTailor がプリフェッチスケジュールを削除する間に遅延が発生する場合があります、その間はプリフェッチの取得と消費がバックグラウンドで引き続き実行されます。

コンソールを使用してスケジュールされたアクションを削除する

1. MediaTailor コンソール (<https://console.aws.amazon.com/mediatailor/>) を開きます。
2. ナビゲーションペインで [Configurations] (設定) をクリックします。削除するプリフェッチスケジュールが含まれている再生設定を選択します。

3. [Prefetch schedules] (プリフェッチスケジュール) タブで、削除するプリフェッチスケジュールを選択します。その後、[Delete] (削除) をクリックします。

での前提条件付き広告の使用 AWS Elemental MediaTailor

[一般的な広告挿入ワークフロー](#)では、MediaTailor はコンテンツストリームに合わせて広告を動的にトランスコードし、保存して、ライブストリームに広告をステッチします。このプロセスは、MediaTailor が広告決定サーバー (ADS) から VAST レスポンスで広告を受信した後にのみ行われるため、広告がステッチ可能になるまでに遅延が発生します。広告ステッチングワークフローに追加のレイテンシーが発生した場合 (ADS タイムアウトやその他のコンテンツやネットワークの問題が原因)、MediaTailor は部分的に表示を埋めたり、広告時間枠を完全に見逃したりする可能性があります。

コンテンツに広告をステッチするのに必要な時間を短縮するために、前提条件付き広告を使用できます。前提条件付き広告は、MediaTailor 広告挿入で使用する前にトランスコードする広告です。ADS に無条件広告URLs を提供する代わりに、前提条件付き広告の URLs を指定します。MediaTailor リクエストに対する VAST レスポンスで、ADS には前提条件付き広告への直接リンクが含まれています。広告ステッチングのトランスコード部分を削除することで、MediaTailor は広告を保存してコンテンツストリームにステッチするだけで済みます。事前条件付き広告を使用した広告ステッチングプロセスにより、MediaTailor が VAST レスポンスを通じて広告を認識してから、広告がコンテンツにステッチされるまでの時間が短縮されます。

または、広告プリフェッチを使用することもできます。これは、広告時間枠が必要になる前に、スケジュールされた時刻に広告ステッチングプロセスを実行するように MediaTailor を設定する場合です。広告プリフェッチの詳細については、「」を参照してください[広告のプリフェッチ](#)。

前提条件付き広告の要件

以下は、前提条件付き広告で広告ステッチングワークフローを設定する際に考慮すべき要件です。

MediaFiles の要件

広告サーバーが MediaTailor に送信する VAST レスポンスには MediaFiles、以下の要件を満たすものを含める必要があります。

広告 (Creative) には、コンテンツストリームのビットレートバリエーションに準拠するバリエーションが必要です。VAST レスポンスで適切な広告バリエーションを使用してテンプレートマニフェストを照合するのはお客様の責任です。

前提条件付き広告を使用すると、広告挿入の効率を高めることができますが、MediaTailor では、広告のメディアファイルがコンテンツマニフェスト仕様と互換性があることを保証するトランスコードプロセスを管理する機能はありません。広告がコンテンツストリームと一致しない場合、MediaTailor が挿入を見逃したり、一致しないと再生デバイスがエラーになる可能性があります。

さらに、MediaTailor トランスコードなしでコンテンツストリームにステッチインするには、が次の要件を満たしているMediaFile必要があります。

- MediaTailor がダウンロードできるようにするには、パブリックインターネットでアクセス可能である必要があります。
- VAST レスponsdelivery="streaming"で として示されているストリーミング配信を使用する必要があります。
- (HLS .m3u8の場合) または .mpd (DASH の場合) ファイルである必要があります。

Example VAST レスpons

次の VAST レスponsの例から、MediaTailor は次の URL MediaFileで を挿入します。URLs

- HLS ストリームの場合、MediaTailor は を使用しますhttps://example-ad-origin.amazonaws.com/ad1/index_low.m3u8。これは、ストリーミング配信とサポートされているファイル拡張子 (.m3u8) MediaFileを持つ最初の です。
- DASH ストリームの場合、MediaTailor は を使用しますhttps://example-ad-origin.amazonaws.com/ad1/index.mpd。これは、ストリーミング配信とサポートされているファイル拡張子 (.mpd) MediaFileを持つ最初の です。

```
<?xml version="1.0" encoding="UTF-8"?>
<VAST xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="3.0">
  <Ad id="ad1">
    <Inline>
      <AdSystem>ExampleAdSystem</AdSystem>
      <AdTitle>ad1</AdTitle>
      <Impression><![CDATA[https://example-impression.amazonaws.com]]></
Impression>
      <AdServingId>de8e0d33-9c72-4d77-bb3a-f7e566fffc605</AdServingId>
      <Creatives>
        <Creative id="creativeId1" sequence="1">
          <Linear skipoffset="00:00:05">
```

```

        <Duration>00:00:30</Duration>
        <MediaFiles>
            <MediaFile delivery="progressive" width="1280" height="720"
type="video/mp4" bitrate="533" scalable="true" maintainAspectRatio="true"><![
CDATA[https://example-ad-origin.amazonaws.com/ad1/ad1.mp4]]></MediaFile>
            <MediaFile delivery="streaming" width="1280"
height="720" type="application/dash+xml" bitrate="533" scalable="true"
maintainAspectRatio="true"><![CDATA[https://example-ad-origin.amazonaws.com/ad1/
index.mpd]]></MediaFile>
            <MediaFile delivery="streaming" width="640"
height="360" type="application/x-mpegURL" bitrate="262" scalable="true"
maintainAspectRatio="true"><![CDATA[https://example-ad-origin.amazonaws.com/ad1/
index_low.m3u8]]></MediaFile>
            <MediaFile delivery="streaming" width="2560"
height="1440" type="application/x-mpegURL" bitrate="1066" scalable="true"
maintainAspectRatio="true"><![CDATA[https://example-ad-origin.amazonaws.com/ad1/
index_high.m3u8]]></MediaFile>
        </MediaFiles>
    </Linear>
</Creative>
</Creatives>
</InLine>
</Ad>
</VAST>

```

広告マニフェストの要件

前提条件付き広告を使用するには、親広告マニフェストと子広告マニフェストが次の要件を満たしている必要があります。

- VAST レスポンスの Creative セクションにリンクされているマニフェストは、親広告マニフェストである必要があります。
- 子広告マニフェスト URLs は相対パスである必要があります。
- 子広告マニフェストは、親マルチバリエーションプレイリストと同じディレクトリに同じレベルで存在する必要があります。子マニフェストをサブディレクトリやその他の場所に配置することはできません。

Example サポートされている親多変量プレイリスト

次の親広告マルチバリエーションプレイリストには、子広告メディアプレイリストの相対 URLs が含まれています。子プレイリストも親マルチバリエーションプレイリストと同じディレクトリにあります。

```
#EXTM3U
#EXT-X-STREAM-INF:BANDWIDTH=150000,RESOLUTION=416x234,CODECS="avc1.42e00a,mp4a.40.2"
index_1.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=440000,RESOLUTION=416x234,CODECS="avc1.42e00a,mp4a.40.2"
index_2.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=640000,RESOLUTION=640x360,CODECS="avc1.42e00a,mp4a.40.2"
index_3.m3u8
```

Exampleサポートされていない親マルチバリエーションプレイリスト: サブディレクトリ

次の親広告多変量プレイリストには、親多変量プレイリストに関連するサブディレクトリにある子プレイリストが含まれています。これは、前提条件付き広告でサポートされているプレイリストではありません。

```
#EXTM3U
#EXT-X-STREAM-INF:BANDWIDTH=150000,RESOLUTION=416x234,CODECS="avc1.42e00a,mp4a.40.2"
child/index_1.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=440000,RESOLUTION=416x234,CODECS="avc1.42e00a,mp4a.40.2"
child/index_2.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=640000,RESOLUTION=640x360,CODECS="avc1.42e00a,mp4a.40.2"
child/index_3.m3u8
```

Exampleサポートされていない親多変量プレイリスト: URLs

次の親広告マルチバリエーションプレイリストには、絶対 URLs を持つ子プレイリストが含まれています。これは、前提条件付き広告でサポートされているプレイリストではありません。

```
#EXTM3U
#EXT-X-STREAM-INF:BANDWIDTH=150000,RESOLUTION=416x234,CODECS="avc1.42e00a,mp4a.40.2"
https://example.mediataylor.us-west-2.amazonaws.com/index_1.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=440000,RESOLUTION=416x234,CODECS="avc1.42e00a,mp4a.40.2"
https://example.mediataylor.us-west-2.amazonaws.com/index_2.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=640000,RESOLUTION=640x360,CODECS="avc1.42e00a,mp4a.40.2"
https://example.mediataylor.us-west-2.amazonaws.com/index_3.m3u8
```

前提条件付き広告ワークフロー

以下は、MediaTailor での広告ステッチングワークフローにおける前提条件付き広告の仕組みの基本的な説明です。ワークフローの最初の部分は、前提条件付き広告を使用するようにセットアップするために実行する必要があるアクションです。2 番目のパートでは、MediaTailor が広告を処理する方法について説明します。

パート 1: 前提条件付き広告のセットアップ

MediaTailor で前提条件付き広告を使用するワークフローを設定するには、次のステップを実行します。

1. などのトランスコーダーサービスを使用して、テンプレートマニフェストのさまざまなビットレート、解像度、コーデックをサポートするバリエーションにクリエイティブを AWS Elemental MediaConvert 条件付けします。
2. VAST レスポンスで使用するため、事前にトランスコードされたメディアファイルの URLs を ADS に提供します。
3. MediaTailor [で再生設定を作成します](#)。事前条件付き広告を使用するには、設定のストリーミングメディアファイルの条件設定で「なし」を選択します。
4. 通常どおり、コンテンツ配信のセットアップを続行します。

パート 2: MediaTailor 広告処理

MediaTailor の広告ステッチングは、「」の説明に従って完了します [MediaTailor 広告挿入の仕組み](#)。MediaTailor は、ADS から VAST レスポンスを受信すると、次のロジックを使用して、広告に対して実行するアクションを決定します。このロジックは、再生設定のストリーミングメディアファイルの条件設定によって決まります。

- ストリーミングメディアファイルのコンディショニングが Transcode に設定されている場合、MediaTailor はメディアファイルを progressive 配信でトランスコードし、マニフェストにステッチします。progressive 配信メディアファイルを含む十分な広告がない場合、MediaTailor はそれらをトランスコードして streaming 配信に使用します。
- ストリーミングメディアファイルのコンディショニングを None に設定すると、MediaTailor は streaming、配信メディアファイルをトランスコードせずにマニフェストに広告をステッチします。streaming 配信メディアファイルを含む十分な広告がない場合、MediaTailor はそれらをトランスコードして progressive 配信に使用します。

MediaTailor 収益化関数の使用

MediaTailor 収益化関数 (関数) を使用すると、 がセッションデータ AWS Elemental MediaTailor を管理し、広告挿入中に広告リクエストを構築する方法をカスタマイズできます。関数を使用すると、外部 APIs 呼び出し、式を使用したデータの変換、広告決定サーバー (ADS) リクエストパラメータまたはプレイヤーパラメータの変更を行うことができます。カスタムインフラストラクチャをデプロイまたは管理する必要はありません。

関数は、JSON データの軽量クエリおよび変換言語である JSONata を使用して式を評価します。セッションデータの読み取り、値の変換、出力の定義を行う JSONata 式を記述します。サポートされている JSONata 関数の完全なリストについては、「」を参照してください[JSONata 式リファレンス](#)。

以下の必要がある場合は、関数を使用します。

- ビューワーデータで広告リクエストを強化します。セッションの開始時に ID サービスを呼び出し、パーソナライズされた広告ターゲティングのためにすべての ADS リクエストに解決された ID を含めます。
- ADS リクエストパラメータをカスタマイズします。セッションデータ、SCTE-35 シグナル、または外部 API レスポンスに基づいて、ADS URL、ヘッダー、または本文を動的に設定します。
- 広告サーバー間で A/B テストを実行します。セッション属性またはランダム割り当てに基づいて、異なる ADS エンドポイント間でトラフィックを分割します。
- マルチステップエンリッチメントパイプラインを構築します。複数の関数を連結して 1 つの API からデータを取得し、変換して、結果を次のステップに渡します。

関数が MediaTailor ワークフローにどのように適合するか

視聴者が再生セッションを開始すると、MediaTailor はライフサイクルフックと呼ばれる広告挿入フローの特定の時点で関数を実行します。

1. 関数を定義し、関数マッピングを通じて再生設定にアタッチします。
2. MediaTailor は、指定されたライフサイクルフックで関数を実行します。
3. この関数は、セッションデータを読み取り、オプションで外部 API を呼び出し、MediaTailor が再生処理中に使用する結果を書き込みます。

Tip

Functions を初めて使用する場合は、から始め「[クイックスタートガイド](#)」で、を読んでください[ライフサイクルフック](#)。式の構文と完全な例については、[JSONata 式リファレンス](#)「」および「」を参照してください[関数の例](#)。

トピック

- [関数クイックスタートガイド](#)

- [関数ライフサイクルフック](#)
- [関数のタイプと構成](#)
- [関数の作成と管理](#)
- [Functions の JSONata 式リファレンス](#)
- [関数の例](#)
- [Functions のトラブルシューティングとモニタリング](#)
- [関数の制限](#)

関数クイックスタートガイド

このガイドでは、最初の関数を作成し、それを再生設定にアタッチして、実行されたことを確認する手順を説明します。最終的には、各ビューワーのデバイスタイプ (ctv、または desktop) を分類し mobile、すべての ADS リクエストで使用できるプレイヤーパラメータとして保存する作業関数が作成されます。

前提条件

開始する前に、既存の MediaTailor 再生設定があることを確認してください。アカウントをお持ちでない場合は、「[MediaTailor の使用開始](#)」を参照してください。

ステップ 1: 関数を作成する

このステップでは、ユーザーエージェント文字列に基づいてビューワーのデバイスタイプを分類し、その結果をプレイヤーパラメータに保存する関数を作成します。関数は、カスタム出力タイプ (外部 API コールなし) と [JSONata 式リファレンス](#) 式を使用してユーザーエージェントを評価します。

1. MediaTailor コンソール (<https://console.aws.amazon.com/mediatailor/>) を開きます。
2. ナビゲーションペインで、[Functions] (関数) を選択します。
3. [関数の作成] を選択してください。
4. ウィザードの作成モдалで、最初から作成を選択し、続行を選択します。
5. 関数タイプで、カスタム出力タイプを選択します。
6. 関数の詳細で、次のように入力します。
 - 関数 ID: myFirstFunction
 - [Description] (説明): Classify device type from user agent

7. カスタム出力設定の出力セクションで、1 行を追加します。

- [Key] (キー): `player_params.deviceType`
- 値: `{% $contains(session.user_agent, 'CTV') ? 'ctv' : $contains(session.user_agent, 'Mobile') ? 'mobile' : 'desktop' %}`

8. [関数の作成] を選択してください。

成功通知は、関数が作成されたことを確認し、関数の詳細ページにリダイレクトされます。

結果の関数設定は次のとおりです。

```
{
  "FunctionId": "myFirstFunction",
  "FunctionType": "CUSTOM_OUTPUT",
  "Description": "Classify device type from user agent",
  "CustomOutputConfiguration": {
    "Runtime": "JSONATA",
    "Output": {
      "player_params.deviceType": "{% $contains(session.user_agent, 'CTV') ? 'ctv' : $contains(session.user_agent, 'Mobile') ? 'mobile' : 'desktop' %}"
    }
  }
}
```

ステップ 2: 関数を再生設定にアタッチする

関数を再生設定のライフサイクルフックにマッピングします。マッピングは、いつ関数を実行するかを MediaTailor に伝えます。

1. ナビゲーションペインで [Configurations] (設定) をクリックします。
2. 更新する再生設定を選択します。
3. [編集] を選択します。
4. Functions 設定セクションを展開します。
5. セッション初期化フックで、ドロップダウン `myFirstFunction` から を選択します。
6. [保存] を選択します。

これにより、[PRE_SESSION_INITIALIZATION](#) ライフサイクルフック `myFirstFunction` にアタッチされます。結果の関数マッピングは次のとおりです。

```
{
  "FunctionMapping": {
    "PRE_SESSION_INITIALIZATION": "myFirstFunction"
  }
}
```

MediaTailor は、この再生設定のすべての新しいセッションの開始時に関数を 1 回実行します。

ステップ 3: セッションを開始し、関数が実行されたことを確認する

新しい再生セッションを開始して、関数をトリガーします。再生設定のセッション初期化エンドポイントにセッション初期化リクエストを使用します。

MediaTailor は、関数の実行ごとに CloudWatch メトリクスを自動的に発行します。オプトインは必要ありません。セッションを開始したら、AWS/MediaTailor 名前空間で次のメトリクスを確認して、関数が実行されたことを確認します。

- `PreSessionInitHook.Invocations` — フックが発射されたことを確認します。
- `PreSessionInitHook.Errors` — 関数が成功した場合は 0 になります。
- `Function.Invocations` — 実行された個々の関数を確認します。このメトリクスには、`FunctionId`、`FunctionType`、および `HookType` デイメンションが含まれているため、`myFirstFunction` 特定の にフィルタリングできます。

関数が失敗した場合、MediaTailor はデフォルトでエラーログイベントをマニフェストログに出力します (設定は必要ありません)。

- `PRE_SESSION_INIT_HOOK_ERROR` — `errorType` と によるフックレベルの障害 `cause`。
- `PRE_SESSION_INIT_FUNCTION_ERROR` — 特定の `functionId` エラーの詳細を含む関数レベルの障害。

次の例は、関数式の構文エラーの `PRE_SESSION_INIT_FUNCTION_ERROR` イベントを示しています。

```
{
  "eventTimestamp": "2024-01-01T12:00:00.076000000Z",
  "eventType": "PRE_SESSION_INIT_FUNCTION_ERROR",
  "eventDescription": "Function execution failed",
  "awsAccountId": "123456789012",
```



```
{
  "originId": "my-config",
  "sessionId": "session-123",
  "requestId": "req-abc",
  "eventId": "5dc6f040-0f72-4e8c-a64e-25eeef62708c",
  "functionId": "myFirstFunction",
  "functionType": "CUSTOM_OUTPUT",
  "executionTimeMs": 2,
  "errorType": "SYNTAX_ERROR",
  "cause": "Expected \"\\)\" before end of expression",
  "input": {}
}
```

eventId フィールドを使用して、同じ実行のフックと関数のエラーイベントを関連付けます。errorType フィールドは失敗のクラスを示します。エラータイプと修正の完全なリスト [トラブルシューティングとモニタリング](#) については、「」を参照してください。

Note

詳細なログ記録を成功させるには、マニフェストログ設定で PRE_SESSION_INIT_HOOK_SUMMARY および PRE_SESSION_INIT_FUNCTION_COMPLETED イベントにオプトインします。概要イベントは、すべての実行のフック結果を表示します。完了したイベントには、各関数の入力、出力、HTTP リクエスト/レスポンスの詳細が表示されます。これらは、ログコストを最小限に抑えるためにデフォルトで無効になっています。詳細については、「[トラブルシューティングとモニタリング](#)」を参照してください。

舞台裏で何が起こるか

先ほど作成した 関数の完全なリクエストフローは次のとおりです。

1. プレイヤーは MediaTailor とのセッションを開始します。
2. MediaTailor は PRE_SESSION_INITIALIZATION ライフサイクルフックを起動し、 を実行します myFirstFunction。
3. 関数は session.user_agent フィールドを評価し、ctv、mobile または desktop に書き込みます player_params.deviceType。
4. MediaTailor はセッションを作成し、マニフェストをプレイヤーに返します。
5. プレイヤーは再生中に広告ブレイクに遭遇します。

6. MediaTailor はPRE_ADS_REQUESTライフサイクルフックを起動し、ADS リクエストを構築します。deviceType はプレイヤーパラメータに保存されるため、動的変数置換を使用して ADS リクエスト URL に含めることができます。
7. ADS はデバイスタイプを使用して、ターゲットを絞った広告クリエイティブを返します。
8. MediaTailor は広告をマニフェストにステッチし、プレイヤーに返します。

何らかの理由で関数が失敗した場合、MediaTailor は出力を破棄し、関数がアタッチされていないかのように続行します。ビューワーには、デバイスタイプのターゲティングなしで広告が表示されます。

推奨されるトピック

これで、再生設定に作業関数がアタッチされました。ここから:

- 各ライフサイクルフックで利用できる入力フィールドと出力名前空間については、「」を参照してください[ライフサイクルフック](#)。
- さまざまな関数タイプとそれらを連鎖させる方法については、「」を参照してください[関数のタイプと構成](#)。
- 完全な作業例については、「」を参照してください[関数の例](#)。

関数ライフサイクルフック

ライフサイクルフックは、MediaTailor が再生中に関数を実行するタイミングを定義します。このページは、入力フィールド、出力名前空間、および各フックのデータフローを管理するルールの完全なリファレンスです。

概要

MediaTailor は 2 つのライフサイクルフックをサポートしています。

- **PRE_SESSION_INITIALIZATION** ビューワーが新しいセッションを開始すると、 が 1 回発生します。対象者セグメントの取得などの 1 回限りの設定作業に使用します。この時点では広告時間枠は発生していないため、広告時間枠コンテキストは使用できません。
- **PRE_ADS_REQUEST** は、ストリーム内の広告ブレイクごとに 1 回、すべての広告決定サーバー (ADS) リクエストの前に起動します。これを使用して、ターゲットデータを使用して ADS リクエストをカスタマイズしたり、ADS URL を変更したり、ヘッダーを追加したりできます。

主な違いはタイミングです。は 1 回PRE_SESSION_INITIALIZATION実行され、セッション全体で保持されるデータを設定しますが、は繰り返しPRE_ADS_REQUEST実行され、各 ADS リクエストを特定の広告時間枠に合わせて調整できます。

PRE_SESSION_INITIALIZATION

起動時

MediaTailor は、新しい再生セッションの開始時に、にマッピングされた関数を 1 PRE_SESSION_INITIALIZATION回実行します。この関数は、MediaTailor が最初のマニフェストレスポンスを構築する前に実行されます。

Input

session.*、player_params.*、および event.*。使用可能なすべてのフィールドについては、「」を参照してください[入力フィールドリファレンス](#)。

許可される出力名前空間

Namespace	使用できるタイプ
player_params.*	文字列、数値、ブール値

に書き込まれた値はplayer_params.*セッションに保持されます。これらを使用できます。

- を介したPRE_ADS_REQUESTライフサイクルフックの入力として player_params.*
- 経由URLs [ADS リクエストの MediaTailor 動的広告変数](#) (例: [player_params.deviceType])
- すべての広告時間枠にわたるセッションの存続期間中

Note

すべてのplayer_params出力キーと値の合計シリアル化サイズは、1,000 文字を超えることはできません。合計がこの制限を超えると、関数出力は破棄されます。詳細については、「[関数の制限](#)」を参照してください。

一般的なユースケース

- 外部サービスから ID または対象者データを取得し、後の ADS リクエストで使用するプレイヤーパラメータに保存します。
- ユーザーエージェントに基づいてデバイスタイプを分類し、分類をプレイヤーパラメータに書き込みます。
- ダウンストリーム広告ブレイク処理が依存するデフォルトのプレイヤーパラメータ値を設定します。
- を通じて ADS URL に含まれるプレイヤーパラメータに値を保存します [ADS リクエストの MediaTailor 動的広告変数](#)。

失敗動作

にアタッチされた関数何らかの理由でPRE_SESSION_INITIALIZATION失敗した場合、MediaTailor は関数の出力を破棄し、関数がアタッチされていないかのように続行します。セッションは通常、関数のプレイヤーパラメータ値なしで開始されます。

PRE_ADS_REQUEST

起動時

MediaTailor は、ADS にリクエストを送信する直前に、広告時間枠PRE_ADS_REQUESTごとに マッピングされた関数を実行します。関数は、マニフェスト処理中に広告オポチュニティが発生するたびに実行されます。

Input

PRE_SESSION_INITIALIZATION、`avail.*`、`scte.*`、`asset.*`および `adsRequest.*` (URL、メソッド、ヘッダー、本文) のすべてのフィールド。使用可能なすべてのフィールドについては、「」を参照してください [入力フィールドリファレンス](#)。

許可される出力名前空間

Namespace	使用できるタイプ	出力の使用方法
<code>player_params.*</code>	文字列、数値、ブール値	この広告ブレイクのセッションプレイヤーパラメータを上書きします。を通じて ADS リクエスト URL で使用で

Namespace	使用できるタイプ	出力の使用方法
		きます ADS リクエストの MediaTailor 動的広告変数 。
session.*	文字列、数値、ブール値	この広告ブレークのセッション変数を上書きします。動的変数置換を使用して ADS リクエスト URL で使用できます。
avail.*	文字列、数値、ブール値	この広告ブレークの avail 変数を上書きします。動的変数置換を使用して ADS リクエスト URL で使用できます。
scte.*	文字列、数値、ブール値	この広告ブレークの SCTE 変数を上書きします。動的変数置換を使用して ADS リクエスト URL で使用できます。
asset.*	文字列、数値、ブール値	この広告ブレークのアセットメタデータ変数 (EXT-X-ASSET タグから) を上書きします。動的変数置換を使用して ADS リクエスト URL で使用できます。HLS 広告マーカの詳細については、 「HLS 広告マーカ」 を参照してください。

Namespace	使用できるタイプ	出力の使用方法
adsRequest.*	String	この広告時間枠の ADS リクエストのみを上書きします。サポートされているフィールド: url、method、headers.<name>、body。url 値はテンプレートとして扱われ、関数の実行 ADS リクエストの MediaTailor 動的広告変数 後にサポートします。永続化されません。

Note

PRE_ADS_REQUEST フックからの出力はすべて一時的なオーバーライドです。これらは現在の広告時間枠の ADS リクエストにのみ適用され、セッションには保持されません。

例 — ADS リクエストの書き換え:

```
{
  "Output": {
    "adsRequest.url": "{%'https://ads.example.com/v1/vast?sid=' & session.id &
'&genre=' & player_params.genre%}",
    "adsRequest.headers.X-Custom-Token": "{%player_params.auth_token%}"
  }
}
```

これにより、ADS URL が置き換えられ、現在の広告時間枠のカスタムヘッダーが追加されます。

一般的なユースケース

- ADS リクエスト URL を書き換えて、A/B テストのために異なる広告サーバー間でトラフィックをルーティングします。
- ADS リクエスト URL またはヘッダーにエンリッチメントデータ (オーディエンスセグメント、ID トークン) を追加します。

- SCTE-35 シグナルデータまたは表示インデックスに基づいて ADS リクエストパラメータを条件付きで変更します。

失敗動作

にアタッチされた関数何らかの理由でPRE_ADS_REQUEST失敗した場合、MediaTailor は関数の出力を破棄し、関数がアタッチされていないかのように続行します。ADS リクエストは、変更せずに元のセッションとリクエストパラメータを使用して送信されます。

入力フィールドリファレンス

フィールド	タイプ	PRE_SESSION_INITIALIZATION	PRE_ADS_REQUEST
session.id	Long	✓	✓
session.uuid	String	✓	✓
session.client_ip	文字列	✓	✓
session.user_agent	文字列	✓	✓
session.referer *	String	✓	✓
session.avail_duration_secs	Long	✗	✓
session.avail_duration_ms	Long	✗	✓
player_params.*	String	✓	✓
event.id	文字列	✓	✓
event.hook	文字列	✓	✓

フィールド	タイプ	PRE_SESSI ON_INITIALIZATION	PRE_ADS_REQUEST
event.timestamp	String	✓	✓
avail.index	Int	✗	✓
avail.random	Long	✗	✓
avail.sou rce_conte nt_time_e poch_ms	Long	✗	✓
scte.event_id	Int	✗	✓
scte.avail_num	Int	✗	✓
scte.segm entation_ event_id	Int	✗	✓
scte.segm entation_ type_id	Int	✗	✓
scte.segm entation_upid	String	✗	✓
scte.segm entation_ upid.assetId	文字列	✗	✓
scte.segm entation_ upid.cueD ata.key	文字列	✗	✓

フィールド	タイプ	PRE_SESSI ON_INITIALIZATION	PRE_ADS_REQUEST
scte.segm entation_ upid.cueD ata.value	String	✗	✓
scte.uniq ue_program_id	Int	✗	✓
scte.arch ive_allow ed_flag	ブール値	✗	✓
scte.deliv ery_not_ restricte d_flag	ブール値	✗	✓
scte.devi ce_restri ctions	Int	✗	✓
scte.no_r egional_b lackout_flag	ブール値	✗	✓
scte.segm ent_num	Int	✗	✓
scte.segm ents_expected	Int	✗	✓
scte.sub_ segment_num	Int	✗	✓

フィールド	タイプ	PRE_SESSI ON_INITIALIZATION	PRE_ADS_REQUEST
scte.sub_ segments_ expected	Int	X	✓
scte.avai ls_expected	Long	X	✓
asset.*	String	X	✓
adsRequest.url	文字列	X	✓
adsReques t.method	文字列	X	✓
adsReques t.headers .<key>	文字列	X	✓
adsRequest.body	String	X	✓

* session.refererは、Referer ヘッダーがセッション初期化リクエストに含まれている場合にのみ存在します。アクセスする前に `$exists(session.referer)` を使用して確認します。

関数のタイプと構成

AWS Elemental MediaTailor には 3 つの関数タイプがあり、それぞれ異なるカテゴリの作業用に設計されています。関数の作成時にタイプを選択すると、そのタイプによって実行時に関数が実行できる操作が決まります。このページでは、各タイプの実行方法、使用するタイミング、関数を複数ステップのパイプラインに構成する方法について説明します。

関数タイプの概要

タイプ	Category	目的
CUSTOM_OUTPUT	データ変換	現在のセッション状態に対して式を評価し、出力を生成し

タイプ	Category	目的
		ます。外部呼び出しはありません。
HTTP_REQUEST	外部統合	外部サービスへの HTTP 呼び出しを行い、レスポンスを参照できる出力式を評価します。
SEQUENTIAL_EXECUTOR	オーケストレーション	一連の関数を順番に実行し、ステップ間でデータを一時データに渡します。

各タイプは個別のロールを提供します。関数を作成するときにタイプを選択します。このタイプは、実行時に関数が実行できる操作を決定します。

CUSTOM_OUTPUT

どのようなときに使うか

関数ロジックが純粋に計算CUSTOM_OUTPUT的である場合に使用します。一般的なユースケースには、ユーザーエージェント文字列に基づくデバイスの分類、A/B テストロジックを使用した広告決定サーバー URLs の選択、広告決定サーバーに到達する前のプレイヤーパラメータの再フォーマットなどがあります。

CUSTOM_OUTPUT 関数は、ネットワーク呼び出しなしでメモリ内の式を評価するため、最速のタイプです。外部サービスに連絡せずに目標を達成できる場合は、常にこのタイプを選択します。

設定フィールド

CUSTOM_OUTPUT 関数には 2 つのフィールドがあります。

- ランタイム — 式言語。これを に設定しますJSONATA。
- 出力 — キーと値のペアのマップ。各キーは出力バインディング名 (などplayer_params.device_type) で、各値は MediaTailor が実行時に評価する式です。

Note

{%...%} を使用して、JSONata 式を文字列フィールドにラップします。これらの区切り文字のない値は、静的文字列として扱われます。たとえば、は式 "{%session.id%}" を評価し、"session.id" はリテラルテキストです。

ランタイム動作

MediaTailor が CUSTOM_OUTPUT 関数を実行すると、出力ブロック内のすべての式を現在のセッション状態に対して評価します。すべての式に同じ入力スナップショットが表示されます。評価された結果は関数の出力になります。

いずれかの式が評価に失敗した場合、MediaTailor は関数の出力全体を破棄し、関数がアタッチされていないかのように続行します。関数は結果を部分的にコミットしません。

例: A/B トラフィック分割

次の関数は、各広告ブレイクを 2 つの広告決定サーバー URLs。PRE_ADS_REQUEST ライフサイクルフック用に設計されています。

```
{
  "FunctionId": "abTestAdsUrl",
  "FunctionType": "CUSTOM_OUTPUT",
  "CustomOutputConfiguration": {
    "Runtime": "JSONATA",
    "Output": {
      "adsRequest.url": "{%$random() < 0.5 ? 'https://ads.example.com/v1' : 'https://ads.example.com/v2'%}"
    }
  }
}
```

この例の完全なチュートリアルについては、「」を参照してください [関数の例](#)。

HTTP_REQUEST

どのようなときに使うか

関数 HTTP_REQUEST が外部サービスを呼び出す必要がある場合に使用します。一般的なユースケースには、解決プロバイダーからの ID データの取得、データ管理プラットフォームからの対象者セグメントの取得、ログ記録エンドポイントへのセッション情報の送信などがあります。

設定フィールド

HTTP_REQUEST 関数には次のフィールドがあります。

- ランタイム — 式言語。これを に設定しますJSONATA。
- MethodType — HTTP メソッド。サポートされている値は GET および POST です。
- URL — リクエストを送信する URL。静的 URL または URL を動的に構築する JSONata 式を使用できます。
- ヘッダー — リクエストに含める HTTP ヘッダー。ヘッダー名と値のペアとして指定されます。動的ヘッダー値には{%...%}式構文を使用します。静的な値は文字列として直接指定できます。
- 本文 — 送信するリクエスト本文。POST リクエストで使用されます。JSONata 式を使用して、本文を動的に構築できます。
- RequestTimeoutMilliseconds (必須) — レスポンスを待機する時間。
- 出力 — HTTP 呼び出しの完了後に生成する値を定義します。各エントリは、responseオブジェクトを参照できる式に出力キー (などplayer_params.envelope_id) をマッピングします。

これらのフィールドに適用されるサイズ制限と制限については、「」を参照してください[制限](#)。

リクエストの処理方法

MediaTailor は、2 つのステップでHTTP_REQUEST関数を処理します。

1. リクエストを構築する — MediaTailor はUrl、Headers、および Body式を現在のセッション状態に対して評価します。これらの評価された値は、アウトバウンド HTTP リクエストを形成します。
2. レスポンスの処理 — HTTP 呼び出しが完了すると、MediaTailor は出力ブロック内の式を評価します。これらの式は、元のセッション状態と呼び出しによって返されるresponseオブジェクトの両方を参照できます。

レスポンスフィールド

HTTP 呼び出しが完了したら、出力式で次のフィールドを参照できます。

フィールド	タイプ	説明
response.body	オブジェクトまたは配列	JSON として解析されたレスポンス本文。本文nullが

フィールド	タイプ	説明
		20,000 文字を超える場合、または有効な JSON でない場合は、 に設定します。
<code>response.statusCode</code>	整数	外部サービスによって返される HTTP ステータスコード。ネットワーク障害null時に を に設定します。
<code>response.text</code>	String	20,000 文字に切り捨てられた文字列としての raw レスポンス本文。ネットワーク障害"Internal Error"時に を に設定します。

⚠ Important

`response.body` フィールドはnull、レスポンスが有効な JSON であっても、レスポンスが 20,000 文字を超える場合です。

ℹ Note

レスポンスオブジェクトは、HTTP_REQUEST関数の出力ブロックでのみ使用できます。URL、ヘッダー、または本文フィールドでレスポンスフィールドを参照することはできません。ではSEQUENTIAL_EXECUTOR、各HTTP_REQUEST関数は独自のレスポンスにのみアクセスできます。

値 は、データが使用できないnullことを意味します。これは、HTTP 呼び出しが失敗した場合(ネットワークエラーまたはタイムアウト)、またはレスポンス本文が 20,000 文字を超えているか、有効な JSON でない場合に発生します。

ネットワーク障害の動作

ネットワークエラーまたはタイムアウトが原因で HTTP 呼び出しが失敗、`response.statusCode` が `200` に設定され、`response.body` が `null`、`response.text` が `"Internal Error"` に設定されている場合、Output 式は引き続き実行されるため、レスポンスデータを使用する `response.statusCode` 前に必ず確認してください。

Tip

条件式を使用して、障害を適切に処理します。 `{%response.statusCode = 200 ? response.body.value : 'default'%}`

例: ID データの取得

次の関数は、セッションの開始時に ID 解決 API を呼び出し、その結果をプレイヤーパラメータに保存します。PRE_SESSION_INITIALIZATION ライフサイクルフック用に設計されています。

```
{
  "FunctionId": "fetchIdentityEnvelope",
  "FunctionType": "HTTP_REQUEST",
  "HttpRequestConfiguration": {
    "Runtime": "JSONATA",
    "MethodType": "GET",
    "Url": "{%'https://identity.example.com/v1/resolve?ip=' &
$encodeURIComponent(session.client_ip)%}",
    "Headers": {
      "Authorization": "{%'Bearer my_api_token'%}",
      "Accept": "application/json"
    },
    "RequestTimeoutMilliseconds": 2000,
    "Output": {
      "player_params.identity_envelope": "{%response.statusCode = 200 ?
response.body.envelope : ''%}"
    }
  }
}
```

同様の例の完全なチュートリアルについては、「」を参照してください [関数の例](#)。

SEQUENTIAL_EXECUTOR

どのようなときに使うか

は、一連の関数を一度に 1 つずつ順番にSEQUENTIAL_EXECUTOR実行します。各ステップは、その前のステップの結果を使用できます。

ロジックが相互に結果に依存する複数のステップを必要とするSEQUENTIAL_EXECUTOR場合に使用します。一般的なユースケースには、アイデンティティデータを取得し、それを使用して対象者セグメントを取得し、分類ステップを実行して、結果に基づいてさまざまな外部サービスを条件付きで呼び出し、複数のデータソースから複雑な広告リクエスト URLsを構築することが含まれます。

設定フィールド

SEQUENTIAL_EXECUTOR 関数には次のフィールドがあります。

- ランタイム — 式言語。これを に設定しますJSONATA。
- FunctionList — 1～10 ステップの順序付きリスト。各ステップでは、実行する関数FunctionIdの を指定します。必要に応じて、ステップを実行するかスキップするかを制御するRunCondition式を追加できます。
- 出力 — すべてのステップの完了後に生成する値を定義します。各エントリは、シーケンス内の任意のステップによって生成されたデータを参照できる式に出力キー (などplayer_params.envelope) をマッピングします。省略すると、シーケンス内の個々の関数からのすべての出力が使用されます。
- TimeoutMilliseconds (必須) — シーケンス全体が完了するまでの最大時間。シーケンスがこのタイムアウトを超えると、MediaTailor はシーケンスからのすべての出力を破棄します。

順序付けられた実行とデータフロー

MediaTailor は、最初から最後まで順序で各ステップを実行します。各ステップが完了すると、生成される値は実行中の結果のセットにマージされます。以降のステップでは、元のセッションデータと、前のステップで生成されたすべての値にアクセスできます。

一時データは、ステップ間でデータを渡すための主要なメカニズムです。関数がtemp.*キーに書き込むと、次のステップでその値を読み取ることができます。前のステップで書き込まれたプレイヤーパラメータと広告リクエストフィールドは、後のステップにも表示されます。

Note

一時データは、オブジェクトや配列を含む任意のデータ型を受け入れます。プレイヤーパラメータと広告リクエストフィールドは、文字列、数字、ブール値、null のみを受け入れます)。

ステップごとの実行条件

シーケンスの各ステップには、オプションの `RunCondition` フィールドがあります。このフィールドには、`true`または `false` を返す式が含まれています。MediaTailor は、そのステップを実行する直前に `RunCondition` 式を評価します。

`RunCondition` 式が `true` と評価された場合 `false`、MediaTailor はステップを完全にスキップし、次のステップに進みます。`RunCondition` フィールドを省略すると、ステップは常に実行されます。

```
{ "FunctionId": "retryFetch", "RunCondition": "{%temp.statusCode = 500%}" }
```

このメカニズムにより、条件付きパイプラインを構築できます。たとえば、ステップ 1 で ID フェッチを実行し、ステップ 1 が有効な ID を返した場合にのみ、条件付きでステップ 2 でセグメントルックアップを実行できます。

出力ブロックの仕組み

の出力ブロックは、すべてのステップが完了した後にシーケンスが生成する内容 `SEQUENTIAL_EXECUTOR` を制御します。

- 出力ブロックが存在する — MediaTailor は、出力ブロック内の式を最終的な累積状態と照らし合わせて評価し、それらの出力のみを保存します。シーケンシャル出力ブロックで参照されていない、前のステップで生成された出力は破棄されます。
- 出力ブロックなし — MediaTailor は、すべてのステップから蓄積されたすべての出力を直接保存します。

Tip

すべての関数の出力を渡す場合は、出力ブロックを省略します。保存する前に累積された結果をフィルタリング、名前変更、または変換する必要がある場合は、出力ブロックを追加します。

タイムアウト設定

TimeoutMilliseconds フィールドは、シーケンス全体の期限を設定します。このタイムアウトは、関数によって行われた HTTP 呼び出しを含むすべてのステップをカバーします。シーケンスがタイムアウトを超えると、MediaTailor はシーケンスからのすべての出力を破棄し、関数がアタッチされていないかのように続行します。

個々のHTTP_REQUEST関数は、引き続き独自のRequestTimeoutMilliseconds設定を尊重します。シーケンスタイムアウトは、合計実行時間を制限する外部境界として機能します。

例: HTTP での再試行の失敗

この例では、ID API を呼び出し、最初の呼び出しがサーバーエラーを返すと自動的に再試行します。SEQUENTIAL_EXECUTOR によってオーケストレーションされた 2 つの HTTP_REQUEST 関数を使用します。

ステップ 1 — プライマリフェッチ (**fetchIdentity**):

```
{
  "FunctionId": "fetchIdentity",
  "FunctionType": "HTTP_REQUEST",
  "HttpRequestConfiguration": {
    "Runtime": "JSONATA",
    "MethodType": "GET",
    "Url": "{% 'https://identity.example.com/v1/resolve?ip=' & session.client_ip%}",
    "Headers": {
      "Accept": "application/json"
    },
    "RequestTimeoutMilliseconds": 1000,
    "Output": {
      "temp.statusCode": "{%response.statusCode%}",
      "temp.envelope": "{%response.statusCode = 200 ? response.body.envelope :
null%}"
    }
  }
}
```

ステップ 2 — 失敗時に再試行する (**retryIdentity**):

```
{
  "FunctionId": "retryIdentity",
  "FunctionType": "HTTP_REQUEST",
  "HttpRequestConfiguration": {
```

```

    "Runtime": "JSONATA",
    "MethodType": "GET",
    "Url": "{% 'https://identity-fallback.example.com/v1/resolve?ip=' &
session.client_ip%}",
    "Headers": {
        "Accept": "application/json"
    },
    "RequestTimeoutMilliseconds": 1000,
    "Output": {
        "temp.statusCode": "{%response.statusCode%}",
        "temp.envelope": "{%response.statusCode = 200 ? response.body.envelope :
null%}"
    }
}
}

```

シーケンス (**identityWithRetry**):

```

{
    "FunctionId": "identityWithRetry",
    "FunctionType": "SEQUENTIAL_EXECUTOR",
    "SequentialExecutorConfiguration": {
        "Runtime": "JSONATA",
        "TimeoutMilliseconds": 2000,
        "FunctionList": [
            { "FunctionId": "fetchIdentity" },
            { "FunctionId": "retryIdentity", "RunCondition": "{%temp.statusCode >=
500%}" }
        ],
        "Output": {
            "player_params.envelope": "{%temp.envelope%}"
        }
    }
}

```

仕組み:

1. `fetchIdentity` は ID API を呼び出し、ステータスコードとエンベロープを `temp.*` に書き込みます。
2. ステータスコードが 500 以上の場合、ステップ 2 `RunCondition` の `temp.statusCode >= 500` が true され、`retryIdentity` 実行されます。再試行レスポンス `temp.envelope` で `temp.statusCode` と `temp.envelope` が上書きされます。

- 最初の呼び出しが成功すると、ステップ 2 はスキップされます。
- シーケンス出力ブロックは `temp.envelope` に書き込みます `player_params.envelope`。

コンポジションルール

MediaTailor は、関数構成に次の制限を適用します。

ルール	制限
最大ネスト深度	2 (シーケンスには関数を含めることができますが、それらの子をシーケンス自体にすることはできません)
シーケンスあたりのステップ	1 ~ 10
ライフサイクルフックあたりの合計関数実行数	20
円形参照	許可されていません。関数は、直接または間接的にそれ自体を参照することはできません。
関数の存在	で参照されるすべての関数は、親シーケンスを作成または更新する前に存在 <code>FunctionList</code> する必要があります。

関数を作成すると、MediaTailor は式構文を検証し、制限された関数をチェックし、参照されるすべての関数が存在することを確認し、循環参照を検出します。関数を再生設定にアタッチすると、MediaTailor は関数ツリー全体のすべての出力キーが割り当てられたライフサイクルフックと互換性があることをさらに検証します。

関数の作成と管理

このページでは、MediaTailor コンソールを使用して Functions を作成、アタッチ、更新、削除する方法について説明します。関数を使用すると、再生中にキーポイントでロジックを実行して、セッション動作と広告リクエストをカスタマイズできます。

関数のセットアップ

コンソールを使用した関数の作成

1. MediaTailor コンソール (<https://console.aws.amazon.com/mediatailor/>) を開きます。
2. ナビゲーションペインで、[Functions] (関数) を選択します。
3. [関数の作成] を選択してください。
4. 関数 ID には、関数の一意の名前 (例:) を入力します `fetchIdentity`。
5. 関数タイプで、CUSTOM_OUTPUT、HTTP_REQUEST、または を選択します SEQUENTIAL_EXECUTOR。コンソールには、選択したタイプの設定フィールドが表示されます。
6. タイプ固有の設定フィールドに入力します。各関数タイプとそのフィールドの説明については、「」を参照してください [関数のタイプと構成](#)。
7. [関数の作成] を選択してください。

関数を再生設定にアタッチする

関数は、関数マッピング (ライフサイクルフックと関数間のリンク) を介して再生設定にアタッチするまで実行されません。関数マッピングは、関数をトリガーするライフサイクルフックを指定します。

1. MediaTailor コンソールを開きます。
2. ナビゲーションペインで [Configurations] (設定) をクリックします。
3. 更新する再生設定を選択します。
4. 関数マッピングセクションで、編集 を選択します。
5. ライフサイクルフックごとに、アタッチする関数を選択します。
 - セッション初期化フック — セッション開始時に 1 回実行する関数を選択します。
 - 広告リクエストフック — 各 ADS リクエストの前に実行する関数を選択します。
6. [保存] を選択します。

関数の管理

関数の表示

Functions に移動して、アカウント内のすべての関数を表示します。設定を表示するには、関数名を選択します。

既存の 関数の更新

関数を更新するには、関数に移動し、関数を選択して、設定を変更します。更新により、関数定義全体が置き換えられます。

Tip

変更を行う前に、関数設定のコピーを保存します。関数には組み込みのバージョンングやロールバックはありません。

関数の削除

再生設定から関数をデタッチする

関数を削除する前に、それを参照するすべての再生設定から削除します。

1. MediaTailor コンソールを開きます。
2. ナビゲーションペインで [Configurations] (設定) をクリックします。
3. 再生設定を選択します。
4. 関数マッピングセクションで、編集 を選択します。
5. ライフサイクルフックから 関数を削除します。
6. [保存] を選択します。

関数の削除

関数に移動し、関数を選択し、削除を選択します。

削除ブロックルール

MediaTailor では、まだ使用中の関数を削除することはできません。

Condition	結果
関数は、関数マッピングを介して再生設定にアタッチされます。	削除はブロックされます。最初に関数をデタッチします。
関数は <code>SEQUENTIAL_EXECUTOR</code> で参照されます	削除はブロックされます。最初に親関数から参照を削除します。
関数はどこにも参照されません	削除は成功しました。

検証ルール

MediaTailor は、関数を作成または更新するときに関数を検証します。次のチェックが実行されます。

- 式構文 — すべての式は有効な JSONata である必要があります。
- 制限された関数 — 式は制限された JSONata 関数を呼び出すことはできません。完全なリスト [JSONata 式リファレンス](#) については、「」を参照してください。
- 出力キープレフィックス — すべての出力キーは、認識された名前空間プレフィックスで始まる必要があります。使用できるプレフィックスのリストについては、「」を参照してください [ライフサイクルフック](#)。
- 関数参照 — 内のすべての関数 IDs 既存の関数を参照 FunctionList する必要があります。
- 循環参照 — 関数は、直接または間接的にそれ自体を参照することはできません。
- ネスト深度 — は他の関数を呼び出す SEQUENTIAL_EXECUTOR ことができますが、それらの関数自体を SEQUENTIAL_EXECUTOR にすることはできません。

特定の値とサイズ制限については、「」を参照してください [制限](#)。

API リファレンス

関数をプログラムで管理するには、 [AWS Elemental MediaTailor API リファレンス](#) を参照してください。

Functions の JSONata 式リファレンス

このページは、Functions で使用できる式の構文、演算子、関数の完全なリファレンスです。出力ブロック、URL フィールド、ヘッダー値、本文テンプレート、実行条件の式を記述するときに使用します。

式区切り文字

関数で定義するすべての値は定数または式のいずれかであり、両方を組み合わせたものではありません。MediaTailor は、区切り文字に基づいて 2 つのを区別します。

構文	タイプ	評価
<code>https://ads.example.com/vast</code>	定数	評価なしでそのまま返されます。
<code>{%session.client_ip%}</code>	式	実行時に評価されます。その結果、値全体が置き換えられます。
<code>GET</code>	定数	そのまま返されます。
<code>{%'https://ads.example.com/vast?ip=' & session.client_ip%'}</code>	式	実行時に評価されます。

Important

値は、完全に定数または完全に式です。2 つのを 1 つの値に混在させることはできません。たとえば、`hello {% 'world' %}` は無効です。静的テキストと動的値を組み合わせるには、式内で文字列連結を使用します `{%'hello ' & 'world' %}`。

言語の基本

パスナビゲーションのドット表記

ドット表記を使用して入力データをトラバースします。各ドットは 1 つのレベルをオブジェクト階層に降順します。


```
session.client_ip           # the viewer's IP address
response.body.envelope      # a field inside a parsed JSON response
player_params.campaign_id   # a player parameter
```

欠落しているフィールドはエラーを発生させnullずに返ります。

```
temp.nonExistent           # null
temp.nonExistent.deeply.nested # null
```

との文字列連結 &

& 演算子は 2 つの文字列値を結合します。文字列以外の値は、自動的に文字列に変換されます。

```
'https://ads.example.com/vast?ip=' & session.client_ip
# "https://ads.example.com/vast?ip=192.0.2.1"

'duration=' & 30
# "duration=30"
```

条件 (三項) 式

条件に基づいて 2 つの値のいずれかを返すには、3 項演算子を使用します。

```
condition ? value_if_true : value_if_false
```

例:

```
$exists(player_params.env) ? player_params.env : 'prod'
response.statusCode = 200 ? response.body.id : 'unknown'
$random() > 0.5 ? 'groupA' : 'groupB'
```

多方向分岐の 3 項式をネストできます。

```
$contains(session.user_agent, 'CTV') ? 'ctv'
  : $contains(session.user_agent, 'Mobile') ? 'mobile'
  : 'desktop'
```

を使用した変数バインディング :=

式内に中間値を割り当てるには、括弧内の := 演算子を使用します。境界変数は括弧で囲まれた括弧に限定され、式の外部には保持されません。

```
(  
  $base := 'https://ads.example.com';  
  $base & '/vast?ip=' & session.client_ip  
)
```

セミコロンは括弧内でステートメントを区切ります。最後のステートメントは、式の戻り値です。

```
(  
  $code := response.statusCode;  
  $code != null and $code >= 200 and $code < 300  
    ? response.body.value  
    : 'fallback'  
)
```

オペレータ

算術

演算子	説明	例	結果
+	加算	5 + 3	8
-	減算	10 - 4	6
*	乗算	6 * 7	42
/	除算	15 / 4	3.75
%	モジュロ	17 % 5	2

Important

プレイヤーパラメータとセッションデータの入力値は文字列として到着します。数値比較または算術の前に変換`$number()`するには、`number`を使用します。文字列を数値と比較すると、予期しない結果が生成されます。

比較

演算子	説明	例	結果
=	等しい	<code>response. statusCode = 200</code>	true
!=	等しくない	<code>player_pa rams.region != 'us-east-1'</code>	true us-east-1 でない場合
<	Less than	<code>avail.index < 3</code>	true 3 未満の場合
>	Greater than	<code>\$number(p layer_par ams.age) > 18</code>	true 18 を超える場合
<=	Less than or equal	<code>\$count(items) <= 10</code>	true 10 以下の場合
>=	Greater than or equal	<code>response. statusCode >= 400</code>	true エラーステータスの場合

ブール値

演算子	説明	例
and	論理 AND	<code>response.statusCod e = 200 and \$exists(r esponse.body.id)</code>
or	論理 OR	<code>player_params.regi on = 'us-east-1' or player_params.region = 'us-west-2'</code>

優先順位には括弧を使用します。

```
score > 0.5 and (tier = 'premium' or tier = 'gold')
```

Note

論理的な否定`$not()`に を使用します。not キーワード演算子はありません。

メンバーシップ (`in`)

`in` 演算子は、配列に値が存在するかどうかをテストします。

```
'premium' in segments      # true if segments contains 'premium'
player_params.region in ['us-east-1', 'us-west-2'] # true
```

連鎖 (`~>`)

連鎖演算子は、左側の式の結果を最初の引数として右側の関数に渡します。

```
session.user_agent ~> $lowercase ~> $trim
# equivalent to $trim($lowercase(session.user_agent))
```

許可された関数

MediaTailor は、以下の組み込み関数をサポートしています。ここでリストされていない関数はブロックされ、関数を作成または更新すると検証エラーが発生します。

型変換 (3)

関数	説明	例	結果
<code>\$string(value)</code>	文字列に変換する	<code>\$string(200)</code>	"200"
<code>\$number(value)</code>	数値に変換する	<code>\$number('42')</code>	42
<code>\$boolean(value)</code>	ブール値に変換する	<code>\$boolean(1)</code>	true

イントロスペクション (4)

関数	説明	例	結果
<code>\$length(string)</code>	文字列の長さ	<code>\$length('hello')</code>	5
<code>\$count(array)</code>	配列要素数	<code>\$count([1, 2, 3])</code>	3
<code>\$exists(value)</code>	値が存在するかどうかを確認する (未定義ではない)	<code>\$exists(temp.id)</code>	true、または false
<code>\$keys(object)</code>	オブジェクトキー名を取得する	<code>\$keys(response.body)</code>	<code>["id", "name"]</code>

数値 (7)

関数	説明	例	結果
<code>\$sum(array)</code>	配列の合計	<code>\$sum([1, 2, 3])</code>	6
<code>\$max(array)</code>	最大値	<code>\$max([10, 5, 20])</code>	20
<code>\$min(array)</code>	最小値	<code>\$min([10, 5, 20])</code>	5
<code>\$average(array)</code>	算術平均	<code>\$average([10, 20, 30])</code>	20
<code>\$abs(number)</code>	絶対値	<code>\$abs(-7)</code>	7
<code>\$floor(number)</code>	切り捨て	<code>\$floor(3.9)</code>	3
<code>\$round(number, precision)</code>	精度を四捨五入する	<code>\$round(3.456, 2)</code>	3.46

文字列 (7)

関数	説明	例	結果
<code>\$uppercase(string)</code>	大文字にするには	<code>\$uppercase('hello')</code>	"HELLO"
<code>\$lowercase(string)</code>	小文字にするには	<code>\$lowercase('Hello')</code>	"hello"
<code>\$trim(string)</code>	先頭/末尾の空白を削除する	<code>\$trim(' hi ')</code>	"hi"
<code>\$substring(string, start, length)</code>	部分文字列の抽出 (ゼロベース)	<code>\$substring('abcdef', 2, 3)</code>	"cde"
<code>\$contains(string, pattern)</code>	文字列にパターンが含まれているかどうかを確認する	<code>\$contains(session.user_agent, 'CTV')</code>	true、または false
<code>\$match(string, pattern)</code>	正規表現パターンと文字列を照合する	<code>\$match('abc-123', /[0-9]+/)</code>	{"match": "123", ...}
<code>\$replace(string, pattern, replacement)</code>	一致するパターンを置き換える	<code>\$replace('hello', 'l', 'r')</code>	"herro"

配列 (5)

関数	説明	例	結果
<code>\$append(arr1, arr2)</code>	配列を連結する	<code>\$append([1, 2], [3, 4])</code>	[1, 2, 3, 4]

関数	説明	例	結果
<code>\$reverse(array)</code>	逆順	<code>\$reverse([1, 2, 3])</code>	<code>[3, 2, 1]</code>
<code>\$sort(array)</code>	ソート配列	<code>\$sort([3, 1, 2])</code>	<code>[1, 2, 3]</code>
<code>\$distinct(array)</code>	重複を除去する	<code>\$distinct([1, 2, 2, 3])</code>	<code>[1, 2, 3]</code>
<code>\$map(array, func)</code>	関数を各要素に適用する	<code>\$map([1,2,3], function(\$v) {\$v*2})</code>	<code>[2, 4, 6]</code>

ブール値 (1)

関数	説明	例	結果
<code>\$not(value)</code>	論理 NOT	<code>\$not(false)</code>	<code>true</code>

ランダム (1)

関数	説明	例	結果
<code>\$random()</code>	0 (含む) から 1 (含む) までの乱数	<code>\$random() > 0.5 ? 'A' : 'B'</code>	"A"、または "B"

Note

`$random()` は、評価ごとに新しい値を生成します。複数の出力キーで同じランダム値が必要な場合は、最初に変数にバインドします(`$r := $random(); ...`)。

日付/時刻 (4)

関数	説明	例	結果
<code>\$now()</code>	ISO 8601 文字列としての現在のタイムスタンプ	<code>\$now()</code>	"2024-01-15T12:00:00.000Z"
<code>\$millis()</code>	エポックからのミリ秒単位の現在のタイムスタンプ	<code>\$millis()</code>	1705320000000
<code>\$toMillis(string)</code>	ISO 8601 文字列をミリ秒に変換する	<code>\$toMillis('2024-01-15T12:00:00.000Z')</code>	1705320000000
<code>\$fromMillis(number)</code>	ミリ秒を ISO 8601 文字列に変換する	<code>\$fromMillis(1705320000000)</code>	"2024-01-15T12:00:00.000Z"

エンコード (6)

関数	説明	例
<code>\$encodeURIComponent(string)</code>	URL エンコード (、 /、 などの構造文字を保持?&)	<code>\$encodeURIComponent('https://example.com/path?q=hello world')</code>
<code>\$encodeURIComponentComponent(string)</code>	URL エンコード単一のコンポーネント (すべての特殊文字をエンコード)	<code>\$encodeURIComponentComponent('a&b=c')</code> → "a%26b%3Dc"
<code>\$decodeUrl(string)</code>	URL エンコードされた文字列をデコードする	<code>\$decodeUrl('hello%20world')</code> → "hello world"

関数	説明	例
<code>\$decodeURIComponent(string)</code>	URL エンコードされたコンポーネントのデコード	<code>\$decodeURIComponent('a%26b')</code> → "a&b"
<code>\$base64encode(string)</code>	Base64 へのエンコード	<code>\$base64encode('hello')</code> → "aGVsbG8="
<code>\$base64decode(string)</code>	Base64 からのデコード	<code>\$base64decode('aGVsbG8=')</code> → "hello"

i Tip

個々のクエリパラメータ値 `$encodeURIComponent()` に を使用します。構造を保持しながら完全な URL をエンコードする必要がある `$encodeUrl()` 場合にのみ使用します。

一般的なパターン

フォールバック値

値が存在しない可能性がある場合は、デフォルトを指定します。

```
{%$exists(player_params.region) ? player_params.region : 'us-east-1'%}
```

動的 URL 構造

複数の入力から広告決定サーバー URL を構築します。

```
{%'https://ads.example.com/v1/vast?ip=' & $encodeURIComponent(session.client_ip) &
'&ua=' & $encodeURIComponent(session.user_agent) & '&sid=' & session.id%}
```

HTTP_REQUEST 出力のステータスコードチェック

出力値を HTTP 障害から保護します。

```
{%response.statusCode != null and response.statusCode = 200 ? response.body.envelope :
'default-envelope'%}
```

プレイヤーパラメータからの数値変換

プレイヤーパラメータは文字列として到着します。算術比較または数値比較の前に変換します。

```
{%$number(player_params.max_duration) > 30 ? 'long' : 'short'%}
```

Important

が数値以外の文字列`$number()`を受け取ると、`undefined`が返されます。パラメータが欠落しているか無効である可能性がある`$exists()`場合は、と組み合わせてください。
(`$val := $number(player_params.max_duration); $exists($val) and $val > 30 ? 'long' : 'short'`)

ランダムトラフィック分割

を使用して、実験グループにビューワーを割り当てます`$random()`。

```
{%$random() > 0.5 ? 'https://ads.example.com/v1/vast-a' : 'https://ads.example.com/v1/vast-b'%}
```

デバイスタイプの分類

ユーザーエージェント文字列に基づいてデバイスを分類します。

```
{%$contains(session.user_agent, 'CTV') ? 'ctv' : $contains(session.user_agent, 'Mobile') ? 'mobile' : 'desktop'%}
```

関数の例

これらの例を実行する前に、以下を理解しておく必要があります。

- [ライフサイクルフック](#) — 各ライフサイクルフックで使える入力フィールドと出力名前空間。
- [関数のタイプと構成](#) — 各関数タイプの仕組みとその設定フィールド。
- [の作成と管理](#) — 関数を作成して再生設定にアタッチする方法。

このページでは、一般的なユースケースの完全で実用的な関数設定について説明します。各例には、シナリオ、完全な設定、関数マッピング、関数の実行時に何が起こるかの説明が含まれます。

例	シナリオの説明
the section called “例 1: データエンリッチメント”	セッションの開始時に LiveRamp ID エンベロップを取得し、プレイヤーパラメータに保存します。
the section called “例 2: A/B トラフィック分割”	A/B テストのために 2 つの広告決定サーバー URLs 間で広告リクエストトラフィックを均等に分割します。

例 1: データエンリッチメント

シナリオ

ストリーミングサービスは、セッションの開始時に外部サービス (この場合は LiveRamp) からオーディエンスアイデンティティデータを取得し、セッションの後半で広告ターゲティングのためにプレイヤーパラメータに保存したいと考えています。ID エンベロップは、アイデンティティ解決サービスによって提供される暗号化された識別子であり、広告システムが個人情報が公開されることなく、デバイスやプラットフォーム間でビューワーを認識できるようにします。

設定

LiveRamp エンベロップ (HTTP_REQUEST):

```
{
  "FunctionId": "fetchLiveRamp",
  "FunctionType": "HTTP_REQUEST",
  "HttpRequestConfiguration": {
    "Runtime": "JSONATA",
    "MethodType": "GET",
    "Url": "{%'https://api.example-identity.com/v2/envelope?pid=12345&iv=' &
player_params.hashId%}",
    "Headers": {
      "Authorization": "{%'Bearer ramp-token-789'%}"
    },
    "RequestTimeoutMilliseconds": 2000,
    "Output": {
      "player_params.envelope_id": "{%response.body.envelopes[0].value%}"
    }
  }
}
```

```
}
```

Note

を独自の API 認証情報 Bearer ramp-token-789 に置き換えます。機密性の高いトークンを本番環境の関数設定に直接保存しないでください。トークンローテーション戦略の使用を検討してください。

外部サービスの予想される応答時間 `RequestTimeoutMilliseconds` に基づいて を設定します。値を小さくするとレイテンシーは短縮されますが、タイムアウトエラーが発生する可能性が高くなります。

関数マッピング

```
{
  "FunctionMapping": {
    "PRE_SESSION_INITIALIZATION": "fetchLiveRamp"
  }
}
```

関数が実行されるとどうなるか

1. ビューワーが再生セッションを開始します。
2. MediaTailor は `PRE_SESSION_INITIALIZATION` ライフサイクルフックを実行し、 を実行します `fetchLiveRamp`。
3. この関数は、 を使用してリクエスト URL を構築し `player_params.hashedImage`、LiveRamp API を呼び出します。URL では、 `pid` はパートナー ID で、 `iv` はハッシュされたビューワー識別子です。
4. 出力式は、レスポンス配列 (`response.body.envelopes[0].value`) の最初のエンベロープから値を抽出し、 に書き込みます `player_params.envelope_id`。

Tip

エラーを明示的に処理するには、レスポンスデータにアクセスする `response.statusCode` 前に を確認してください。 `{%response.statusCode = 200 ? response.body.envelopes[0].value : ''%}`

詳細については、「[HTTP_REQUEST](#)」、「[PRE_SESSION_INITIALIZATION](#)」、「[トラブルシューティングとモニタリング](#)」、および「[制限](#)」を参照してください。

例 2: A/B トラフィック分割

シナリオ

ストリーミングサービスは、CUSTOM_OUTPUT 関数を使用して、広告リクエストトラフィックを A/B テスト用の 2 つの広告決定サーバー (ADS) URLs にランダムに分割したいと考えています。広告リクエストの半分は v1 エンドポイントに送信され、半分は v2 エンドポイントに送信されます。

設定

A/B トラフィック分割 (CUSTOM_OUTPUT):

```
{
  "FunctionId": "abTestAdsUrl",
  "FunctionType": "CUSTOM_OUTPUT",
  "CustomOutputConfiguration": {
    "Runtime": "JSONATA",
    "Output": {
      "adsRequest.url": "{%$random() < 0.5 ? 'https://ads.example.com/v1/decision?session=' & session.id : 'https://ads.example.com/v2/decision?session=' & session.id}"
    }
  }
}
```

この設定では、次のようになります。

- FunctionId — この関数の一意の名前。
- FunctionType — HTTP 呼び出しを行わずに式CUSTOM_OUTPUTを評価します。
- Output — 2 つの ADS URL のいずれかをランダムに選択する式adsRequest.urlにマッピングします。URLs

関数マッピング

```
{
  "FunctionMapping": {
```

```
    "PRE_ADS_REQUEST": "abTestAdsUrl"  
  }  
}
```

関数が実行されるとどうなるか

1. 再生中に広告時間枠が発生します。
2. MediaTailor はPRE_ADS_REQUESTライフサイクルフックを実行し、 を実行しますabTestAdsUrl。
3. `$random()` 関数は 0~1 の値を返します。値が 0.5 未満の場合、関数は ADS URL を v1 エンドポイントに設定します。それ以外の場合は、URL を v2 エンドポイントに設定します。

`$random()` 関数は広告時間枠ごとに評価されるため、分割はセッションごとではなく広告リクエストごとです。

Tip

トラフィックの分割率を調整するには、しきい値を変更します。たとえば、`$random() < 0.8` はトラフィックの 80% を最初の URL に送信し、20% を 2 番目の URL に送信します。

Note

に書き込むと、現在の広告時間枠の再生設定で設定されたデフォルトの ADS URL が`adsRequest.url`上書きされます。

詳細については、[CUSTOM_OUTPUT](#)、[PRE_ADS_REQUEST](#)、および [JSONata 式リファレンス](#) を参照してください。

Functions のトラブルシューティングとモニタリング

このページは、一般的な関数エラーを診断し、本番環境で関数のパフォーマンスをモニタリングするのに役立ちます。トラブルシューティングセクションは症状別に整理されています。観察した内容から始めて、原因と修正に従ってください。

モニタリング

CloudWatch メトリクス

MediaTailor は、関数実行のメトリクスを Amazon CloudWatch に発行します。オプトインは必要ありません。

フックレベルのメトリクス — ライフサイクルフック実行ごとに 1 つのデータポイント:

メトリクス	説明	ディメンション
PreSessionInitHook. .Invocations	フック実行の数	ConfigurationName
PreSessionInitHook. .Errors	フックエラーの数	ConfigurationName
PreSessionInitHook. .Latency	フック実行時間 (ms)	ConfigurationName
PreAdsRequestHook. Invocations	フック実行の数	ConfigurationName
PreAdsRequestHook. Errors	フックエラーの数	ConfigurationName
PreAdsRequestHook. Latency	フック実行時間 (ms)	ConfigurationName

関数レベルのメトリクス — 関数実行ごとに 1 つのデータポイント:

メトリクス	説明	ディメンション
Function.Invocations	関数実行の数	ConfigurationName , FunctionId , FunctionT ype , HookType

メトリクス	説明	ディメンション
Function.Errors	関数エラーの数	ConfigurationName , FunctionId , FunctionT ype , HookType
Function.Latency	関数の実行時間 (ミリ秒)	ConfigurationName , FunctionId , FunctionT ype , HookType

アラームの設定とこれらのメトリクスの操作の詳細については、「」を参照してください[Amazon CloudWatch メトリクス AWS Elemental MediaTailor によるモニタリング](#)。

ログイベント

MediaTailor は、関数実行のログイベントを発行します。エラーイベントはデフォルトで出力されます。完了したイベントと概要イベントはオプションされます。

イベントタイプ	デフォルト/オプション	ロググループ	説明
PRE_SESSION_INIT_HOOK_SUMMARY	オプション	マニフェストログ	フック実行の概要 (成功/エラー)
PRE_SESSION_INIT_HOOK_ERROR	デフォルト	マニフェストログ	errorType と cause によるフック障害
PRE_SESSION_INIT_FUNCTION_COMPLETED	オプション	マニフェストログ	入力/出力で完了した個々の関数
PRE_SESSION_INIT_FUNCTION_ERROR	デフォルト	マニフェストログ	個々の関数の失敗

イベントタイプ	デフォルト/オプション	ロググループ	説明
PRE_ADS_REQUEST_HOK_SUMMARY	オプション	ADS インタラクションログ	フック実行の概要 (成功/エラー)
PRE_ADS_REQUEST_HOK_ERROR	デフォルト	ADS インタラクションログ	errorType と cause によるフック障害
PRE_ADS_REQUEST_FUNCTION_COMPLETED	オプション	ADS インタラクションログ	入力/出力で完了した個々の関数
PRE_ADS_REQUEST_FUNCTION_ERROR	デフォルト	ADS インタラクションログ	個々の関数の失敗

オプションログイベントを有効にするには、「」を参照してください[Amazon CloudWatch メトリクス AWS Elemental MediaTailor によるモニタリング](#)。

eventId フィールドを使用して、同じ実行のフックレベルイベントと関数レベルのイベントを関連付けます。

次の Amazon CloudWatch Logs Insights クエリは、関数エラーイベントを でフィルタリングeventIdして、単一の実行をトレースします。

```
fields @timestamp, eventType, functionId, errorType, cause
| filter eventId = "5dc6f040-0f72-4e8c-a64e-25eeef62708c"
| sort @timestamp asc
```

トラブルシューティング

関数が失敗すると、MediaTailor はエラーイベントに errorType フィールドを記録します。このフィールドを使用して、障害のクラスを特定します。

エラータイプ	説明
SYNTAX_ERROR	式がコンパイルに失敗したか、ランタイムタイプエラーが発生しました
RESOURCE_LIMIT_ERROR	式が CPU 時間、メモリ、またはスタック深度の制限を超えました
RESTRICTION_ERROR	ブロックされた関数を使用した式、または入力ペイロードサイズの制限を超えた式
TIMEOUT_ERROR	関数の実行が時間制限を超えました
VALIDATION_ERROR	出力パスは、現在のフックのスコープで書き込み不可能なフィールドをターゲットにします。
INTERNAL_ERROR	関数に関連しないインフラストラクチャの障害

エントリは症状別に整理され、必要に応じてこれらのエラータイプを参照します。

式が予期せず null を返す

症状: 入力されるはずの出力値がプレイヤーパラメータであるか、null不足しています。

考えられる原因:

原因	の識別方法	Fix
入力フィールドはこのライフサイクルフックに存在しません。	PRE_SESSION_INITIALIZATION 関数adsRequest.url で参照した。ADS リクエストデータはセッション開始時には使用できません。	関数をPRE_ADS_REQUEST ライフサイクルフックに移動するか、別の入力フィールドを使用します。「 ライフサイクルフック 」を参照してください。
入力フィールドがセッションデータにありません。	を参照しましたがplayer_params.campaign_id 、プレイヤーはセッションの初期	を使用して\$exists() 、にアクセスする前に確認します{%\$exists(player_params.campaign_id) ?

原因	の識別方法	Fix
	化時にそのパラメータを渡していません。	<code>player_params.campaign_id : 'default' %}</code> 。
オブジェクトまたは配列をプレイヤーパラメータまたはADS リクエストに書き込みました。	これらの名前空間は、文字列、数値、およびブール値のみを受け入れます。オブジェクトと配列は除外されます。	複雑なデータを に保存temp.*し、後続のステップで文字列、数値、またはブール値を抽出します。

RESOURCE_LIMIT_ERROR: スタックオーバーフロー

症状: 関数は `errorType: "RESOURCE_LIMIT_ERROR"` と で失敗します `cause: "Stack overflow error"`。

原因: 式がスタックの最大深度である 100 レベルを超えました。これは通常、深くネストされた条件式 (if/then/else) または複雑な変数割り当てで発生します。

つまり、式には MediaTailor が処理するネストのレベルが多すぎます。

修正: 式を簡素化します。複雑なロジックを複数の出力エントリまたはシーケンシャルエグゼキューター内の複数のステップに分割します。

RESOURCE_LIMIT_ERROR: CPU タイムアウト

症状: 関数は `errorType: "RESOURCE_LIMIT_ERROR"` と で失敗します `cause: "Expression evaluation timeout: Check for infinite loop"`。

原因: 式が 100 ミリ秒の CPU 時間制限を超えました。これは、大規模なデータ構造に対して高価な計算を実行する式で発生する可能性があります。

修正: 式の複雑さを軽減します。大きな配列を処理する場合は、そのロジックを外部サービスに移動し、HTTP_REQUEST関数を使用して呼び出すことを検討してください。

RESTRICTION_ERROR: 関数は許可されていません

症状: 関数は `errorType: "RESTRICTION_ERROR"` と で失敗します `cause: "Function '<name>' is not allowed"`。

原因: 式は、38 個の関数の許可されたリストに含まれていない JSONata 関数を呼び出します。一般的な例には、`$filter`、`$reduce`、`$eval`、`$split`、および `$join`。

修正: ブロックされた関数名の `cause` フィールドを確認します。許可された代替手段に置き換えます。38 個の許可された関数の完全なリスト [JSONata 式リファレンス](#) については、「」を参照してください。

一般的に使用される許可された関数には `$string`、`$number`、`$substring`、`$contains`、および `$encodeURIComponent`。

RESTRICTION_ERROR: 式が長すぎます

症状: 関数はでの作成または更新に失敗します `cause: "Expression length <actual> exceeds limit <limit>"`。

原因: 単一の式が 1,000 文字を超えています。

修正: 式を小さな部分に分割します。複数の出力エントリを使用するか、シーケンシャルエグゼキューター内の複数のステップにロジックを分割します。変数バインディング (`:=`) を使用して、長い部分式が繰り返されないようにします。

HTTP エラー: ステータスコードが `null`

症状: `HTTP_REQUEST` 関数の出力では、`response.statusCode` は `null` です。

原因: 外部 API に到達できなかった、接続がタイムアウトした、またはネットワークエラーが発生しました。この場合、MediaTailor は `response.statusCode` を `null` に設定し、`response.body` を `null` に設定し、`response.text` を `"Internal Error"` に設定します。

修正: レスポンスデータにアクセスする `response.statusCode` 前に必ず確認してください。

```
{%response.statusCode = 200 ? response.body.value : 'default'%}
```

この式は、HTTP 呼び出しが 200 ステータスコードを返したかどうかを確認します。その場合は、レスポンス値を使用します。それ以外の場合は、デフォルト値に戻ります。

この問題が頻繁に発生する場合は、外部 API が正常かどうかを確認します。API が遅い `RequestTimeoutMilliseconds` 場合はを増やし、高速に失敗する場合は減らすことを検討してください。

HTTP エラー: レスポンス本文が `null`

症状: `response.statusCode` は 200 ですが、`response.body` は `null` です。

原因: レスポンス本文が無効な JSON であるか、20,000 文字を超えています。MediaTailor は、最大 20,000 文字の JSON レスポンスのみを に解析します `response.body`。

修正: フォールバック `response.text` として を使用します。 `response.text` フィールドには、20,000 文字に切り捨てられた未加工のレスポンス本文が含まれます。

```
{%response.statusCode = 200 ? ($exists(response.body.id) ? response.body.id :  
  $substring(response.text, 0, 100)) : 'error'%}
```

必要なデータが 20,000 文字の制限を超える場合は、外部 API に小さなレスポンスを返すように依頼することを検討してください (たとえば、特定のフィールドをリクエストするなど)。

HTTP エラー: URL 検証の失敗

症状: 関数は実行時に失敗し、URL の形式が正しくないか、無効なスキームを使用しているか、最大長を超えています。

考えられる原因:

原因	Fix
URL は <code>http</code> または <code>https</code> を使用しません。	URL 式が <code>http://</code> または <code>https://</code> で始まる URL を生成していることを確認します。
式の評価後、URL が 2,048 文字を超えています。	URL を短くします。POST メソッドを使用して、大きなパラメータ値をリクエスト本文に移動します。
URL の形式が正しくありません (有効な URI ではありません)。	式に欠落している文字や余分な文字がないか確認します。特殊文字を含む可能性のあるクエリパラメータ値 <code>\$encodeURIComponent()</code> には、 <code>encodeURIComponent()</code> を使用します。

VALIDATION_ERROR

症状: 関数は で失敗します `errorType: "VALIDATION_ERROR"`。このエラーは、作成時 (関数を作成または更新するとき) または実行時 (セッション中に関数を実行するとき) に発生する可能性があります。

考えられる原因:

原因	例	Fix
出力キーは、現在のフックで書き込みできない名前空間をターゲットにします。	にアタッチされた関数 <code>adsRequest.url</code> の書き込み <code>PRE_SESSION_INITIALIZATION</code> 。	ライフサイクルフックで許可される出力名前空間を確認します。は <code>PRE_SESSION_INITIALIZATION</code> のみを許可します <code>player_params.*</code> 。関数を正しいフックに移動するか、出力キーを変更します。「 ライフサイクルフック 」を参照してください。
出力キーは無効な文字を使用します。	のような出力キー <code>player_params.device type</code> (スペース付き)。文字、数字、アンダースコア、ハイフンのみを使用できます。	有効な文字のみを使用するように出力キーの名前を変更します。たとえば、 <code>player_params.device_type</code> 代わりに <code>device_type</code> を使用します。
出力キーが有効なプレフィックスで始まらない。	<code>player_params.myValue</code> または <code>custom.myValue</code> の代わりに のような出力キー <code>temp.myValue</code> 。	有効な出力プレフィックス <code>player_params.*</code> 、 <code>temp.*</code> 、または <code>adsRequest.*</code> を使用します。
JSONata 式には構文エラーがあります。	不足している終了引用符または不完全な条件: <code>{%session.id & %}</code> 。	式で、引用符の欠落、括弧の不一致、または <code>??</code> やなどのサポートされていない演算子を確認します <code>?:</code> 。
HTTP_REQUEST 関数に必須フィールドがありません。	URL フィールドが空であるか、メソッドが指定されていません。	URL フィールドとメソッドフィールドが設定されていることを確認します。メソッドは GET または POST である必要があります。
式によって構築された URL が無効です。	評価された URL は、 のようなサポートされて	URL 式が有効な URL <code>http://</code> または <code>https://</code>

原因	例	Fix
	いないスキームを使用するftp://か、2,048 文字を超えているか、形式が正しくありません。	URL を生成することを確認します。特殊文字を含む可能性のあるクエリパラメータ値 <code>\$encodeURIComponent()</code> に を使用します。
HTTP ヘッダーに無効な文字が含まれているか、制限された名前を使用しています。	ヘッダー値に改行が含まれているか、ヘッダー名が host または <code>transfer-encoding</code> 。	ヘッダー値から無効な文字を削除します。制限されたヘッダー名は避けてください。ヘッダーの制限 HTTP_REQUEST については、「」を参照してください。

エラーログイベントの `cause` フィールドを確認します。どのフィールドまたは式が検証に失敗したかを識別します。

INTERNAL_ERROR

症状: 関数は で失敗します `errorType: "INTERNAL_ERROR"`。

原因: 関数設定に関連しないインフラストラクチャの障害が発生しました。

修正: リクエストを再試行します。エラーが解決しない場合は、AWS サポートにお問い合わせください。

関数の制限

このページには、関数に適用される制限が一覧表示されます。これらの制限は、作成時 (関数を作成または更新するとき) と実行時 (関数の実行時) に適用されます。

フック実行の制限

フック	制限	値
PRE_SESSION_INITIALIZATION	合計タイムアウト	2,000 ミリ秒

フック	制限	値
PRE_SESSION_INITIALIZATION	合計player_params サイズ	1,000 文字
PRE_ADS_REQUEST	合計タイムアウト	2,000 ミリ秒

これらのタイムアウトは、すべての関数ステップや HTTP 呼び出しを含むフック実行全体を対象としています。フックがタイムアウトを超えると、MediaTailor はすべての出力を破棄し、関数がアタッチされていないかのように続行します。

個々の関数タイムアウト (HTTP_REQUEST RequestTimeoutMilliseconds 関数の場合など) は、フックの合計タイムアウト内に収まる必要があります。たとえば、フックタイムアウトが 2,000 ミリ秒で、HTTP_REQUEST 関数が 2,000 ミリ秒RequestTimeoutMillisecondsに設定されている場合、HTTP 呼び出しの前後に処理が発生した場合、関数は完了する前にタイムアウトすることがあります。

player_params 合計サイズには、すべての出力キーの名前と値の合計文字数が含まれます。たとえば、player_params.deviceType = "mobile"と mobileを出力すると、deviceType1,000 文字の合計us-east-1に対して region、、、 がplayer_params.region = "us-east-1"カウントされます。

式の制限

制限	値	で強制
式あたりの CPU 時間	100 ミリ秒	実行時間
スタックの深さ	100	実行時間
式の長さ	1,000 文字	オーサリング

実行時に式が制限を超えると、MediaTailor は式を停止し、エラーを記録します。特定のエラータイプの詳細については、「」を参照してください[トラブルシューティングとモニタリング](#)。

関数構成の制限

制限	値
シーケンシャルエグゼキューターあたりのステップ	1 ~ 10
最大ネスト深度	2
フックあたりの関数実行の合計数	20
関数あたりの出力エントリ	20
円形参照	許可されていません

これらの制限は、オーサリング時に適用されます。

HTTP リクエストの制限

制限	値
URL の長さ	2,048 文字
リクエスト本文のサイズ	64 KB
ヘッダー数	50
ヘッダー名の長さ	256 文字
ヘッダー値の長さ	8,192 文字
リクエストタイムアウト	100 ~ 2,000 ミリ秒 (を介してお客様が設定可能RequestTimeoutMilliseconds)
許可された URL スキーム	https, http
制限付きヘッダー	Host, Transfer-Encoding , Content-Length , Connection

URL の長さ、本文サイズ、ヘッダー数、ヘッダーサイズの制限は、オーサリング時に適用されます。リクエストのタイムアウトと制限付きヘッダーは、実行時に適用されます。

関数が制限付きヘッダーを設定すると、MediaTailor は作成時に関数設定を受け入れますが、実行中に HTTP リクエストが送信されるとヘッダーを削除します。

レスポンスの制限

制限	値
<code>response.body</code> 最大サイズ	20,000 文字
<code>response.text</code> 最大長	20,000 文字

これらの制限は実行時に適用されます。

出力制限

制限	値
出力値の最大長 (キーあたり)	1,000 文字

これらの制限は実行時に適用されます。

許可された JSONata 関数

許可される関数の完全なリストについては、「」を参照してください[JSONata 式リファレンス](#)。

これらの制限に関連するエラーの診断については、「」を参照してください[トラブルシューティングとモニタリング](#)。

ADS リクエストの MediaTailor 動的広告変数

AWS Elemental MediaTailor は動的広告変数を使用して、表示セッションから広告決定サーバー (ADS) に情報を渡します。この情報は、ADS がビューワーに最も関連性の高い広告を選択するのに役立ちます。

このセクションでは、動的広告変数の概要と、特定の実装ガイドへのリンクについて説明します。step-by-stepの設定手順については、以下の個々のトピックを参照してください。

動の変数タイプ

MediaTailor は、次の 4 種類の動の変数をサポートしています。

- セッション変数 – セッション ID や SCTE-35 データなど、自動的に生成された値。「[ADS リクエストの MediaTailor セッション変数](#)」を参照してください。
- プレイヤー変数 – ビデオプレイヤーによって送信されるカスタムパラメータ。「[ADS リクエストの MediaTailor プレイヤー変数](#)」を参照してください。
- 設定エイリアスを持つドメイン変数 – マルチオリジン設定の動的 URL ドメイン。
- 設定エイリアス – 動の変数置換の事前定義されたマッピング。「[設定エイリアス](#)」を参照してください。

一般的なユースケース

動的广告変数を使用して以下を行います。

- ビューワの属性と設定を ADS に渡す
- 地理的位置に基づいて異なるオリジンにリクエストをルーティングする
- MediaPackage 統合でタイムシフト表示を有効にする
- A/B テストとフェイルオーバーのシナリオを実装する

以下のセクションでは、MediaTailor での動的广告変数の使用について詳しく説明します。

トピック

- [ADS リクエストの MediaTailor セッション変数](#)
- [ADS リクエストの MediaTailor プレイヤー変数](#)
- [複数のコンテンツソースの MediaTailor ドメイン変数](#)
- [MediaTailor 設定エイリアスの概要](#)
- [MediaTailor が ADS にパラメータを渡す](#)
- [ADS およびオリジンの MediaTailor パラメータルーティング](#)
- [MediaTailor と MediaPackage のタイムシフト表示統合](#)
- [MediaTailor パラメータセッションの動作と永続性](#)
- [MediaTailor パラメータのリファレンスと制限事項](#)
- [MediaTailor パラメータのトラブルシューティングガイド](#)

- [MediaTailor 設定エイリアスのトラブルシューティングガイド](#)

パラメータのフォーマット要件とトラブルシューティングについては、[MediaTailor パラメータのリファレンスと制限事項](#)「」および「」を参照してください[MediaTailor パラメータのトラブルシューティングガイド](#)。

ADS リクエストの MediaTailor セッション変数

AWS Elemental MediaTailor テンプレート ADS URL のこのセクションに記載されている 1 つ以上の変数を指定する AWS Elemental MediaTailor 用に を設定すると、 はセッションデータを広告決定サーバー (ADS) に送信します。個々の変数を使用する、および複数の変数を連結して単一の値を作成することが可能です。MediaTailor は一部の値を生成し、マニフェスト、およびプレイヤーのセッション開始リクエストといったソースから残りの値を取得します。

次の表は、テンプレート ADS リクエスト URL 設定で使えるセッションデータ変数を示しています。表に記載されているセクション番号は、米国ケーブル電気通信技術者協会 (SCTE)-35 仕様の 2019 年版、[デジタルプログラム挿入キューイングメッセージに対応しています。広告プリフェッチの詳細については、「」](#)を参照してください[広告のプリフェッチ](#)。

名前	広告プリフェッチに使用可能	SCTE-35 仕様セッション	[Description] (説明)
[avail.index]	はい		インデックス内の広告表示の位置を表す数値。再生セッションの開始時に、MediaTailor はマニフェスト内のすべての ad avail のインデックスを作成し、セッションが継続する間そのインデックスを保存します。MediaTailor が avail を埋めるために ADS に対してリクエストを行うときは、そのリクエストに ad avail インデックス番号を含めます。このパラメータにより、ADS は、競合相手の除外や頻度の上限設定などの機能を使用して広告の選択を改良できます。
[avail.random]	はい		MediaTailor が ADS へのリクエストごとに生成する 0 ~ 10,000,000,000 の乱数。競合する会社から広告を切り離すなどの機能を有効にするために、このパラメータを使用する広告サーバーもあります。

名前	広告プリフェッチに使用可能	SCTE-35仕様セクション	[Description] (説明)
[scte.archive_allowed_flag]	はい	10.3.3.1	オプションのブール値。この値が 0 の場合、記録制限はセグメントでアサートされます。この値が 1 の場合、記録の制限はセグメントでアサートされません。
[scte.avail_num]	はい	9.7.2.1	長い数値としてavail_num、SCTE-35 フィールドから MediaTailor によって解析された値。MediaTailor はこの値を使用して、リニア ad avail 番号を指定できます。 値は整数である必要があります。
[scte.avails_expected]	はい	9.7.2.1	現在のイベント内の予想可用性数を指定するオプションのロング値。
[scte.delivery_not_restricted_flag]	はい	10.3.3.1	オプションのブール値。この値が 0 の場合、次の 5 ビットが予約されます。この値が 1 の場合、SCTE-35 仕様で説明されているように、次の 5 ビットが意味を持ちます。
[scte.device_restrictions]	はい	10.3.3.1	デバイスの 3 つの事前定義された独立した非階層グループにシグナルを送信するオプションの整数値。この変数の詳細については、SCTE-35 仕様の segments_expected の説明を参照してください。
[scte.event_id]	はい	9.1 および 9.7.2.1	長い数値としてsplice_event_id、SCTE-35 フィールドから MediaTailor によって解析された値。MediaTailor はこの値を使用して、リニア ad avail 番号の指定や、広告ポッドの位置などの広告サーバークエリ文字列の入力を行います。 値は整数である必要があります。

名前	広告プリフェッチに使用可能	SCTE-35仕様セクション	[Description] (説明)
[scte.no_regional_blackout_flag]	はい	10.3.3.1	オプションのブール値。この値が 0 の場合、リージョンのブラックアウト制限がセグメントに適用されます。この値が 1 の場合、リージョンのブラックアウト制限はセグメントには適用されません。
[scte.segment_num]	はい	10.3.3.1	セグメントのコレクション内のセグメントに番号を付けるオプションの整数値。この変数の詳細については、SCTE-35 仕様の segment_num の説明を参照してください。
[scte.segmentation_event_id]	はい	10.3.3.1	MediaTailor はこの変数を として公開します scte.event_id 。
[scte.segmentation_type_id]	はい	10.3.3.1	セグメンテーションタイプを指定するオプションの 8 ビット整数値。この変数の詳細については、SCTE-35 仕様の segmentation_type_id の説明を参照してください。

名前	広告プリ フェッチに 使用可能	SCTE-35 仕様セク ション	[Description] (説明)
[scte.segmentation_upid]	segmentation_upid_type : はい private_data : はい	segmentation_upid: 10.3.3.1 マネージド プライベート UPID: 10.3.3.3	SCTE-35 segmentation_upid 要素に対応します。segmentation_upid 要素には、segmentation_upid_type と segmentation_upid_length が含まれます。 MediaTailor は以下の segmentation_upid タイプをサポートします。 • ADS 情報 (0x0e) - 広告情報。詳細については、SCTE-35 仕様の segmentation_upid の説明を参照してください。 • Managed Private UPID (0x0C) - SCTE-35 仕様で定義されている Managed Private UPID (MPU) 構造。MediaTailor は、バイナリまたは DASH XML SCTE 表現をサポートします。 この構造は podbustler ワークフローで使用できます。使用するには、32 ビット (4 バイト) の format_identifier を指定し、private_data 属性に以下のパラメータを含めます。 <pre>ABCD{"assetId": " my_program ", "cueData": {"cueType": " theAdType ", "key": " pb", "value": " 123456"}}</pre> MediaTailor は、上記の JSON からの値を解析し、それらを scte.segmentation_upid.assetId、scte.segmentation_upid.cueData.key、および scte.segmentation_upid.cueData.value 動的変数に渡します。 • ユーザー定義 (0x01) - ユーザー定義の構造。詳細については、SCTE-35 仕様の segmentation_upid の説明を参照してください。

名前	広告プリフェッチに使用可能	SCTE-35仕様セクション	[Description] (説明)
[scte.segmentation_upid.assetId]	はい		ポッドバスターワークフローsegmentation_upid_type 用の Managed Private UPID (0xC) と組み合わせて使用します。MediaTailor は、この値を MPU の private_data JSON 構造内の assetId パラメータから取得します。詳細については、「 Managed Private UPID JSON structure for a podbuster workflow 」を参照してください。
[scte.segmentation_upid.cueData.key]	はい		ポッドバスターワークフローsegmentation_upid_type 用の Managed Private UPID (0xC) と組み合わせて使用します。MediaTailor は、この値を MPU の private_data JSON 構造内の cueData.key パラメータから取得します。詳細については、「 Managed Private UPID JSON structure for a podbuster workflow 」を参照してください。
[scte.segmentation_upid.cueData.value]	はい		ポッドバスターワークフローsegmentation_upid_type 用の Managed Private UPID (0xC) と組み合わせて使用します。MediaTailor は、この値を MPU の private_data JSON 構造内の cueData.key パラメータから取得します。詳細については、「 Managed Private UPID JSON structure for a podbuster workflow 」を参照してください。 値は文字列にすることができます。

名前	広告プリフェッチに使用可能	SCTE-35仕様セクション	[Description] (説明)
[scte.segmentation_upid.private_data.{index}]	はい		ターゲットを絞った広告ワークフローsegmentation_upid_type の Managed Private UPID (0xC) と組み合わせて使用します。MediaTailor は、コロン区切りのセグメンテーション UPID トークンを分割し、インデックス付きセッション変数を作成します。インデックスはコロン区切りリスト内の位置に対応し、最初のコロンの先頭の空白は無視されます。 たとえば、 の場合segmentation_upid = ":3213214:2313321/5:3943" 、次のようになります。 • [scte.segmentation_upid.private_data.0] = 3213214 • [scte.segmentation_upid.private_data.1] = 2313321/5 • [scte.segmentation_upid.private_data.2] = 3943 値は文字列にすることができます。
[scte.segments_expected]	はい	10.3.3.1	セグメントのコレクション内の個々のセグメントの予想数を指定するオプションの整数値。この変数の詳細については、SCTE-35 仕様の segments_expected の説明を参照してください。
[scte.sub_segment_num]	はい	10.3.3.1	サブセグメントのコレクション内の特定のサブセグメントを識別するオプションの整数値。この変数の詳細については、SCTE-35 仕様の sub_segment_num の説明を参照してください。

名前	広告プリ フェッチに 使用可能	SCTE-35 仕様セク ション	[Description] (説明)
[scte.sub _segments _expected]	はい	10.3.3.1	サブセグメントのコレクション内の個々のサブセグメントの予想数を指定するオプションの整数値。この変数の詳細については、SCTE-35 仕様の sub_segments_expected の説明を参照してください。
[scte.unique_program_id]	はい	9.7.2.1	SCTE-35 splice_insert フィールド から MediaTailor によって解析された整数値unique_program_id 。ADS は、一意のプログラムID (UPID) を使用して、ライブリニアストリームにプログラムレベルの広告ターゲティングを提供します。SCTE-35 コマンドがスプライス挿入ではない場合、MediaTailor はこれを空の値に設定します。 値は整数である必要があります。

名前	広告プリフェッチに使用可能	SCTE-35仕様セクション	[Description] (説明)
[session.avail_duration_ms]	はい		広告可用性スロットのミリ秒単位の期間。デフォルト値は 300,000 ms です。次のように入カマニフェストから期間値 AWS Elemental MediaTailor を取得します。 - HLS の場合: MediaTailor は、#EXT-X-DATERANGE タグの #EXT-X-CUE-OUT: DURATIONまたは値から期間を取得します。入カマニフェストでこれらのタグの ad avail の継続時間が null、無効、または 0 になっている場合、MediaTailor はデフォルトを使用します。 - DASH の場合: MediaTailor は、イベント期間が指定されている場合、イベント期間から期間値を取得します。指定されていない場合は、デフォルト値を使用します。 - VOD の場合: VOD ストリームがプレロール広告呼び出しをトリガーする場合、マニフェストに継続時間値を持つ SCTE メッセージングが含まれていない場合、MediaTailor はデフォルトの継続時間値を含む [session.avail_duration_ms] の継続時間を入力しません。
[session.avail_duration_secs]	はい		最も近い秒に丸められた、広告可用性スロットまたは広告表示の秒単位の期間。MediaTailor は、[session.avail_duration_ms] を判断するときと同じ方法でこの値を判断します。
[session.client_ip]	いいえ		MediaTailor リクエストの送信元のリモート IP アドレス。X-forwarded-for ヘッダーが設定されている場合、その値は MediaTailor が client_ip に使用する値です。

名前	広告プリフェッチに使用可能	SCTE-35仕様セクション	[Description] (説明)
[session.id]	いいえ		現在の再生セッションの一意の数値識別子。プレイヤーがセッションに対して行うリクエストにはすべて同じ ID が割り当てられるため、その ID は同一の視聴に対するリクエストを関連付けるための ADS フィールドに使用できます。
[session.referer]	いいえ		通常、動画プレーヤーをホストしているページの URL。MediaTailor はこの変数を、プレイヤーが MediaTailor に対するリクエストで使用した Referer ヘッダーの値に設定します。プレイヤーがこのヘッダーを提供しない場合、MediaTailor は [session.referer] を空のままにしておきます。マニフェストエンドポイントの前にコンテンツ配信ネットワーク (CDN) またはプロキシを使用し、この変数を表示する場合は、ここでプレイヤーから正しいヘッダーをプロキシします。
[session.user_agent]	いいえ		MediaTailor がプレイヤーのセッション初期化リクエストから受け取った User-Agent ヘッダー。マニフェストエンドポイントの前で CDN またはプロキシを使用している場合は、ここでプレイヤーからの正しいヘッダーをプロキシする必要があります。
[session.uuid]	いいえ		の代わりに を使用します [session.id] 。これは、以下のような現在の再生セッションの一意の識別子です。 e039fd39-09f0-46b2-aca9-9871cc116cde

名前	広告プリフェッチに使用可能	SCTE-35仕様セクション	[Description] (説明)
[avail.source_content_time_epoch_ms]	いいえ		HLS の場合、値は表示を開始したオリジンセグメントの PDT です。DASH の場合、値は <code><Period></code> を含む <code><SupplementalProperty></code> <code>urn:scte:dash:utc-time</code> のです <code><EventStream></code>。 • プリロール HLS または DASH 広告の場合、広告 <code>[avail.source_content_time_epoch_ms]</code> は最初の HLS セグメントの PDT と最初の <code><SupplementalProperty></code> <code>urn:scte:dash:utc-time</code> のです <code><Period></code>。短いマニフェストウィンドウを持つライブストリームの場合、この値は、ビューワーがストリームの視聴を開始するタイミングに応じて異なります。完全なイベント DVR ウィンドウを持つライブストリームの場合、値はビューワーごとに同じになります。 • 表示の抑制による部分的な表示の場合、<code>[avail.source_content_time_epoch_ms]</code> は表示を開始したソースコンテンツセグメントの PDT です。例えば、MediaTailor が表示の最初の 20 秒を削除しても、<code>[avail.source_content_time_epoch_ms]</code> は元の PDT に設定され、20 秒シフトされません。 • プリフェッチ ADS リクエストの場合、<code>[avail.source_content_time_epoch_ms]</code> は空ですが、お客様はプリフェッチスケジュールの取得動的変数を使用して設定できます。

Example

ADS で一意のセッション識別子を付けて渡される `deviceSession` というクエリパラメータが必要な場合、AWS Elemental MediaTailor のテンプレート ADS URL は以下のようになります。

```
https://my.ads.server.com/path?deviceSession=[session.id]
```

AWS Elemental MediaTailor は、ストリームごとに一意の識別子を自動的に生成し、 の代わりに識別子を入力しますsession.id。識別子が 1234567 である場合、MediaTailor が ADS に対して行う最終リクエストは、以下のようになります。

```
https://my.ads.server.com/path?deviceSession=1234567
```

ADS で複数のクエリパラメータを渡す必要がある場合、 のテンプレート ADS URL は次の AWS Elemental MediaTailor ようになります。

```
https://my.ads.server.com/sample?
e=[scte.avails_expected]&f=[scte.segment_num]&g=[scte.segments_expected]&h=[scte.sub_segment_num]
```

次の DASH マーカーの XML フラグメントの例は、 の使用方法を示していますscte35:SpliceInsert。

```
<Period start="PT444806.040S" id="123456" duration="PT15.000S">
  <EventStream timescale="90000" schemeIdUri="urn:scte:scte35:2013:xml">
    <Event duration="1350000">
      <scte35:SpliceInfoSection protocolVersion="0" ptsAdjustment="180832" tier="4095">
        <scte35:SpliceInsert spliceEventId="1234567890"
spliceEventCancelIndicator="false" outOfNetworkIndicator="true"
spliceImmediateFlag="false" uniqueProgramId="1" availNum="1" availsExpected="1">
          <scte35:Program><scte35:SpliceTime ptsTime="5672624400"/></scte35:Program>
          <scte35:BreakDuration autoReturn="true" duration="1350000"/>
        </scte35:SpliceInsert>
      </scte35:SpliceInfoSection>
    </Event>
  </EventStream>
</Period>
```

次の DASH マーカーの XML フラグメントの例は、 の使用方法を示していますscte35:TimeSignal。

```
<Period start="PT346530.250S" id="123456" duration="PT61.561S">
  <EventStream timescale="90000" schemeIdUri="urn:scte:scte35:2013:xml">
    <Event duration="5310000">
      <scte35:SpliceInfoSection protocolVersion="0" ptsAdjustment="183003" tier="4095">
        <scte35:TimeSignal>
          <scte35:SpliceTime ptsTime="3442857000"/>
        </scte35:TimeSignal>
      </scte35:SpliceInfoSection>
    </Event>
  </EventStream>
</Period>
```

```

    <scte35:SegmentationDescriptor segmentationEventId="1234567"
segmentationEventCancelIndicator="false" segmentationDuration="8100000"
segmentationTypeId="52" segmentNum="0" segmentsExpected="0">
    <scte35:DeliveryRestrictions webDeliveryAllowedFlag="false"
noRegionalBlackoutFlag="false" archiveAllowedFlag="false" deviceRestrictions="3"/>
    <scte35:SegmentationUpid segmentationUpidType="12"
segmentationUpidLength="2">0100</scte35:SegmentationUpid>
  </scte35:SegmentationDescriptor>
</scte35:SpliceInfoSection>
</Event>

```

次の DASH マーカーの XML フラグメントの例は、 の使用方法を示しています scte35:Binary。

```

<Period start="PT444806.040S" id="123456" duration="PT15.000S">
  <EventStream schemeIdUri="urn:scte:scte35:2014:xml+bin" timescale="1">
    <Event presentationTime="1541436240" duration="24" id="29">
      <scte35:Signal xmlns="http://www.scte.org/schemas/35/2016">
        <scte35:Binary>/DAhAAAAAAAAAAP/wEAUAAAHaf+9/fgAg9YDAAAAAAAAA25aoh</Binary>
      </scte35:Signal>
    </Event>
    <Event presentationTime="1541436360" duration="24" id="30">
      <scte35:Signal xmlns="http://www.scte.org/schemas/35/2016">
        <scte35:Binary>QW5vdGhlciB0ZXN0IHN0cmVuZyBmb3IgaW5jb2RpbmcgdG8gQmFzZTY0IGVuY29kZWQgYmluYXJ5Lg==
Binary>
      </scte35:Signal>
    </Event>
  </EventStream>
</Period>

```

次の HLS タグの例は、 の使用方法を示しています EXT-X-DATERANGE。

```

#EXT-X-DATERANGE:ID="splice-6FFFFFF0",START-DATE="2014-03-05T11:
15:00Z",PLANNED-DURATION=59.993,SCTE35-OUT=0xFC002F0000000000FF0
00014056FFFFFFF000E011622DCAFF00005263620000000000A0008029896F50
0000087000000000

```

次の HLS タグの例は、 の使用方法を示しています EXT-X-CUE-OUT。

```

#EXT-OATCLS-SCTE35:/DA0AAAAAAAAAAAAABQb+ADAQ6QAeAhxDVUVJQAAA03/PAAEUrEoICAAAAAaG
+2UBNAAANvrtoQ==
#EXT-X-ASSET:CAID=0x0000000020FB6501
#EXT-X-CUE-OUT:201.467

```

次の HLS タグの例は、 の使用方法を示していますEXT-X-SPLICEPOINT-SCTE35。

```
#EXT-X-SPLICEPOINT-SCTE35:/DA9AAAAAAAAAAP/wBQb+uYbZqwAnAiVDVUVJAAAKqX//  
AAEjW4AMEU1EU05CMDAxMTMyMjE5M190NAAAmXz5JA==
```

次の例は、scte35:Binaryデコードの使用方法を示しています。

```
{  
  "table_id": 252,  
  "section_syntax_indicator": false,  
  "private_indicator": false,  
  "section_length": 33,  
  "protocol_version": 0,  
  "encrypted_packet": false,  
  "encryption_algorithm": 0,  
  "pts_adjustment": 0,  
  "cw_index": 0,  
  "tier": "0xFFF",  
  "splice_command_length": 16,  
  "splice_command_type": 5,  
  "splice_command": {  
    "splice_event_id": 448,  
    "splice_event_cancel_indicator": false,  
    "out_of_network_indicator": true,  
    "program_splice_flag": true,  
    "duration_flag": true,  
    "splice_immediate_flag": false,  
    "utc_splice_time": {  
      "time_specified_flag": false,  
      "pts_time": null  
    },  
    "component_count": 0,  
    "components": null,  
    "break_duration": {  
      "auto_return": false,  
      "duration": {  
        "pts_time": 2160000,  
        "wall_clock_seconds": 24.0,  
        "wall_clock_time": "00:00:24:000000"  
      }  
    }  
  },  
  "unique_program_id": 49152,  
  "avail_num": 0,  
}
```



```
    "avails_expected": 0,
    "segment_num": 0,
    "segments_expected": 0,
    "sub_segment_num": 0,
    "sub_segments_expected": 0
  },
  "splice_descriptor_loop_length": 0,
  "splice_descriptors": null,
  "Scte35Exception": {
    "parse_status": "SCTE-35 cue parsing completed with 0 errors.",
    "error_messages": [],
    "table_id": 252,
    "splice_command_type": 5
  }
}
```

ADS リクエストの MediaTailor プレイヤー変数

AWS Elemental MediaTailor テンプレート ADS URL

で `player_params.<query_parameter_name>` 変数を指定する AWS Elemental MediaTailor ように を設定すると、 はプレイヤーから受信したデータを ADS に送信します。例えば、プレイヤーが MediaTailor に対するリクエストで `user_id` という名前のクエリパラメータを送信する場合、ADS リクエストでそのデータを渡すには、ADS URL 設定に `[player_params.user_id]` を含めます。

これにより、ADS リクエストに含まれるクエリパラメータを制御できます。通常、ADS が認識する特殊なクエリパラメータを ADS リクエスト URL に追加し、そのパラメータの値としてキーと値のペアを指定します。

この後の手順で使用されている例では、以下のキーと値のペアを使用しています。

- `param1` と値 `value1`:
- `param2` と値 `value2`:

クエリパラメータをキーバリューペアとして追加する

1. で AWS Elemental MediaTailor、パラメータを参照するように ADS リクエストテンプレート URL を設定します。以下の URL では、サンプルパラメータが含まれていることがわかります。

```
https://my.ads.com/path?param1=[player_params.param1]&param2=[player_params.param2]
```

2. (オプション) サーバー側の広告追跡レポートの場合は、プレイヤーのキーバリューペアを URL でエンコードします。MediaTailor がセッション開始リクエストを受信すると、値を URL で 1 回エンコードしてから、それらを ADS リクエスト URL に代入します。

 Note

ADS が URL でエンコードされた値を必要とする場合は、プレイヤーでこの値を 2 回 URL エンコードします。そうすることで、MediaTailor によって行われたデコードから、ADS のために 1 回エンコードされた値が得られます。

例えば、ADS に送信された値のデコードされた表現が `param1=value1:&param2=value2:` である場合、URL でエンコードされた表現は `param1=value1%3A&param2=value2%3A` です。

3. プレイヤーからのセッション開始呼び出しで、キーバリューペアを単一のクエリパラメータの値として MediaTailor に渡します。以下のサンプルの呼び出しでは、サーバー側とクライアント側の広告追跡レポートに使用されるキーと値のペアの例を示しています。

- サーバー側の広告追跡レポートのリクエスト例 - URL エンコードペアを使用する

HLS:

```
<master>.m3u8?ads.param1=value1%3A&ads.param2=value2%3A
```

DASH:

```
<manifest>.mpd?ads.param1=value1%3A&ads.param2=value2%3A
```

- クライアント側の広告追跡レポートのリクエスト例 - URL エンコードを使用しない

HLS:

```
POST <master>.m3u8
{
  "adsParams": {
    "param1": "value1:",
    "param2": "value2:"
  }
}
```

DASH:

```
POST <manifest>.mpd
{
  "adsParams": {
    "param1": "value1:",
    "param2": "value2:"
  }
}
```

サーバー側のレポートの場合、MediaTailor はプレイヤーリクエストを受信したときにパラメータをデコードします。クライアント側のレポートの場合、MediaTailor は JSON ペイロードで受信したパラメータを変更しません。MediaTailor は以下のリクエストを ADS に送信します。

```
https://my.ads.com/<path>?param1=value1:&param2=value2:
```

そうすることで、param1 と param2 のキーバリューペアが第 1 クラスのクエリパラメータとして ADS リクエストに含まれます。

複数のコンテンツソースの MediaTailor ドメイン変数

AWS Elemental MediaTailor 動的ドメイン変数を使用すると、URL my-ads-server.com の http://my-ads-server.com 部分など、設定内のプレイヤーパラメータで複数のドメインを使用できます。これにより、単一の設定内で複数のコンテンツソースまたは広告決定サーバー (ADS) を使用することが可能になります。

ドメイン変数は、URI が含まれる任意のパラメータで使用できます。

- AdDecisionServerUrl
- AdSegmentUrlPrefix
- ContentSegmentUrlPrefix
- LivePreroll.AdDecisionServerUrl
- VideoContentSourceUrl

ドメイン変数は、動的変数置換を実行するために、設定のエイリアスと共に使用されます。設定エイリアスは、動的ドメイン設定に使用されるプレイヤーパラメータに一連のエイリアスと値をマップします。セットアップ手順については、「」を参照してください[MediaTailor での設定エイリアスの作](#)

[成と使用](#)。詳細なリファレンス情報については、「」を参照してください[MediaTailor 設定エイリアスの概要](#)。

MediaTailor 設定エイリアスの概要

AWS Elemental MediaTailor 設定エイリアスは、URL ドメインやその他のサポートされているフィールドで動的変数置換を有効にします。この機能を使用して、複数のドメインを使用し、セッションの初期化中に URLs動的に設定します。

ユースケース

設定エイリアスは、以下のシナリオで高度なマルチ設定アーキテクチャを有効にします。

- 地理的ルーティング: リージョン固有のエイリアスを使用して、ビューワーの場所に基づいて異なるオリジンまたは広告サーバーにリクエストをルーティングします。実装ガイダンスについては、[CloudFront オリジンフェイルオーバー](#)」を参照してください。
- コンテンツベースのルーティング: さまざまなコンテンツタイプを特殊なオリジンまたは処理パイプラインに向けます。ルーティング動作の設定については、「」を参照してください[the section called “CDN ルーティング動作を設定する”](#)。
- フェイルオーバーシナリオ: エイリアス切り替えを使用して、自動フェイルオーバーでバックアップオリジンと広告サーバーを実装します。詳細な実装については、[MQAR を使用して MediaTailor のマルチリージョンレジリエンスを実装する](#)「」および「」を参照してください[the section called “CDN 統合を計画する”](#)。
- A/B テスト: プレイヤーパラメータに基づいてトラフィックをルーティングすることで、さまざまな広告サーバー、オリジン、または設定をテストします。負荷テストのガイダンスについては、「[実際のユーザーモニタリングを使用した Amazon CloudFront のパフォーマンステストの準備と実行](#)」を参照してください。
- デバイス固有の最適化: さまざまなデバイスタイプや機能に合わせてコンテンツ配信と広告配信を最適化します。包括的なガイダンスについては、「」を参照してください[MediaTailor、MediaPackage、CDN でマニフェストフィルタリングを設定する](#)。
- ロードバランシング: 動的ルーティングを使用して、複数のオリジンまたは広告サーバーに負荷を分散します。実装ガイダンスについては、[MQAR を使用して MediaTailor のマルチリージョンレジリエンスを実装する](#)「」および「」を参照してください[the section called “CDN 統合を計画する”](#)。

サポートされているフィールド

動的変数は、次の設定フィールドで使用できます。

- VideoContentSourceUrl
- AdDecisionServerUrl
- LivePreroll.AdDecisionServerUrl
- AdSegmentUrlPrefix
- ContentSegmentUrlPrefix
- TranscodeProfileName
- SlateAdUrl
- StartUrl
- EndUrl

以下のセクションでは、設定エイリアスを使用する方法について説明します。

トピック

- [MediaTailor での設定エイリアスの作成と使用](#)
- [MediaTailor を使用した設定エイリアスの使用例](#)

MediaTailor での設定エイリアスの作成と使用

ドメイン変数の使用を開始する前に、設定のための設定エイリアスを作成します。設定エイリアスは、セッション開始時のドメイン置換変数として使用します。

制限事項

設定エイリアスの使用時は、以下の制限に注意してください。

- ドメインで使用される動的変数は、すべて ConfigurationAliases 動的変数として定義される必要があります。
- プレイヤーパラメータ変数の前に player_params. を付ける必要があります。例えば、player_params.origin_domain などです。
- エイリアスされた値のリストは、重要な URLs のドメイン変数 (VideoContentSourceUrl、AdSegmentUrlPrefix、) に対して包括的である必要がありますContentSegmentUrlPrefix。

- 動変数を指定しないか、無効なエイリアスを使用する重要な URLs のドメイン変数に対してリクエストが行われた場合、リクエストは HTTP 400 ステータスコードで失敗します。重要でないフィールド (SlateAdUrl、バンパー URLs) は警告をログに記録しますが TranscodeProfileName、リクエストは失敗しません。

欠落しているエイリアスのフォールバック動作

設定エイリアスが見つからないか無効である場合、MediaTailor は次のフォールバック動作を実装します。

- ドメイン変数: ドメイン変数エイリアスがないか無効の場合、リクエストは HTTP 400 ステータスコードで失敗します。すべてのドメイン変数には、有効なエイリアスが定義されている必要があります。
- 非ドメイン変数: URLs の非ドメイン部分 (パス要素やクエリパラメータなど) で使用される変数の場合、エイリアスがないと空の文字列置換になります。
- 設定の検証: MediaTailor は、設定の作成および更新オペレーション中にすべての必要なエイリアスが存在することを確認します。

ステップ 1: 設定エイリアスを作成する

MediaTailor コンソールを使用してドメイン置換に使用する設定エイリアスを作成するには、以下の手順を実行します。

Console

コンソールを使用して設定エイリアスを作成する

1. MediaTailor コンソール (<https://console.aws.amazon.com/mediatailor/>) を開きます。
2. [Configurations] (設定) ページの [Configuration aliases] (設定エイリアス) ページで、[Add player parameter] (プレイヤーパラメータを追加) をクリックします。
3. Player パラメータには、動変数として使用するプレイヤーパラメータの名前を入力します。例えば、player_params.origin_domain。
4. エイリアスには、プレイヤーパラメータに使用するエイリアスとその値を入力します。
5. [OK] を選択してください。

AWS Elemental MediaTailor は、設定エイリアスセクションのテーブルに新しいパラメータを表示します。

6. 上記のステップを繰り返して、プレイヤーパラメータを追加します。
7. [保存] を選択します。

API

API を使用して設定エイリアスを作成するには

MediaTailor 設定を作成または更新するときは、次の JSON 構造で `ConfigurationAliases` パラメータを使用します。

```
{
  "ConfigurationAliases": {
    "player_params.origin_domain": {
      "pdx": "abc.mediapackage.us-west-2.amazonaws.com",
      "iad": "xyz.mediapackage.us-east-1.amazonaws.com"
    },
    "player_params.ad_type": {
      "customized": "abc12345",
      "default": "defaultAdType"
    }
  }
}
```

ステップ 2: セッション初期化で設定エイリアスを使用する

設定エイリアスをセットアップしたら、セッション開始リクエスト内のドメイン用の置換変数としてそれらを使用できます。これは、セッションのドメインを動的に設定することを可能にします。

Example 基本設定エイリアスの例

設定エイリアスと動的ドメイン変数を含む設定の基本的な例を次に示します。

```
PUT /playbackConfiguration
{
  "Name": "aliasedConfig",
  "AdDecisionServerUrl": "https://abc.execute-api.us-west-2.amazonaws.com/ads?sid=[session.id]&ad_type=[player_params.ad_type]",
  "VideoContentSourceUrl": "https://[player_params.origin_domain].mediapackage.[player_params.region].amazonaws.com/out/v1/[player_params.endpoint_id]",
  "ConfigurationAliases": {
    "player_params.origin_domain": {
```

```
        "pdx": "abc",
        "iad": "xyz"
    },
    "player_params.region": {
        "pdx": "us-west-2",
        "iad": "us-east-1"
    },
    "player_params.endpoint_id": {
        "pdx": "abcd",
        "iad": "wxyz"
    },
    "player_params.ad_type": {
        "customized": "abc12345",
        "default": "defaultAdType"
    }
}
```

Example エイリアスによるセッションの初期化

上記の設定を使用すると、プレイヤー変数とエイリアスを使用したセッション初期化リクエストは次のようになります。

```
POST index.m3u8
{
  "playerParams": {
    "origin_domain": "pdx",
    "region": "pdx",
    "endpoint_id": "pdx",
    "ad_type": "customized"
  }
}
```

MediaTailor は、エイリアス文字列を、設定エイリアスの設定内のマップされた値に置き換えます。

ADS へのリクエストは次のようになります。

```
https://abc.execute-api.us-west-2.amazonaws.com/ads?sid=[session.id]&ad_type=abc12345
```

マニフェストのオリジンへのリクエストは次のようになります。

```
https://abc.mediapackage.us-west-2.amazonaws.com/out/v1/abcd
```


MediaTailor を使用した設定エイリアスの使用例

次の例は、設定エイリアスを含む完全な MediaTailor 設定、エイリアスを含むセッション初期化リクエスト、およびエイリアスの処理フローを示しています。

Exampleエイリアスを使用して設定を完了する

次の例は、設定エイリアスと動的ドメイン変数を含む完全な設定を示しています。

```
PUT /playbackConfiguration
{
  "Name": "aliasedConfig",
  "AdDecisionServerUrl": "https://abc.execute-api.us-west-2.amazonaws.com/ads?sid=[session.id]&ad_type=[player_params.ad_type]",
  "VideoContentSourceUrl": "https://[player_params.origin_domain].mediapackage.[player_params.region].amazonaws.com/out/v1/[player_params.endpoint_id]",

  "AdSegmentUrlPrefix": "https://[player_params.ad_cdn_domain]/ads/",
  "ContentSegmentUrlPrefix": "https://[player_params.content_cdn_domain]/content/",
  "TranscodeProfileName": "[player_params.transcode_profile]",
  "SlateAdUrl": "https://[player_params.slate_domain]/slate/[player_params.slate_type].mp4",
  "StartUrl": "https://[player_params.tracking_domain]/start?session=[session.id]",
  "EndUrl": "https://[player_params.tracking_domain]/end?session=[session.id]",

  "ConfigurationAliases": {
    "player_params.origin_domain": {
      "pdx": "abc",
      "iad": "xyz"
    },
    "player_params.region": {
      "pdx": "us-west-2",
      "iad": "us-east-1"
    },
    "player_params.endpoint_id": {
      "pdx": "abcd",
      "iad": "wxyz"
    },
    "player_params.ad_type": {
      "customized": "abc12345",
      "default": "defaultAdType"
    },
    "player_params.ad_cdn_domain": {
```

```

        "pdx": "ads-west.cdn.example.com",
        "iad": "ads-east.cdn.example.com"
    },
    "player_params.content_cdn_domain": {
        "pdx": "content-west.cdn.example.com",
        "iad": "content-east.cdn.example.com"
    },
    "player_params.transcode_profile": {
        "mobile": "mobile_optimized",
        "desktop": "high_quality",
        "tv": "4k_profile"
    },
    "player_params.slate_domain": {
        "pdx": "slate-west.example.com",
        "iad": "slate-east.example.com"
    },
    "player_params.slate_type": {
        "standard": "default_slate",
        "branded": "brand_slate"
    },
    "player_params.tracking_domain": {
        "pdx": "tracking-west.example.com",
        "iad": "tracking-east.example.com"
    }
}
}

```

Example エイリアスによるセッションの初期化

次の例は、プレイヤー変数とエイリアスを指定するセッション初期化リクエストを示しています。

```

POST master.m3u8
{
    "playerParams": {
        "origin_domain": "pdx",
        "region": "pdx",
        "endpoint_id": "pdx",
        "ad_type": "customized",
        "ad_cdn_domain": "pdx",
        "content_cdn_domain": "pdx",
        "transcode_profile": "mobile",
        "slate_domain": "pdx",
        "slate_type": "branded",
        "tracking_domain": "pdx"
    }
}

```

```
}  
}
```

Exampleパラメータ処理フロー

次の例では、MediaTailor はエイリアス文字列を設定エイリアスのマッピングされた値に置き換えます。処理の結果、次のリクエストが発生します。

- ADS リクエスト:

```
https://abc.execute-api.us-west-2.amazonaws.com/ads?sid=[session.id]&ad_type=abc12345
```

- VideoContentSource リクエスト:

```
https://abc.mediapackage.us-west-2.amazonaws.com/out/v1/abcd
```

- AdSegmentUrlPrefix:

```
https://ads-west.cdn.example.com/ads/
```

- ContentSegmentUrlPrefix:

```
https://content-west.cdn.example.com/content/
```

- TranscodeProfileName:

```
mobile_optimized
```

- SlateAdUrl:

```
https://slate-west.example.com/slate/brand_slate.mp4
```

- StartUrl:

```
https://tracking-west.example.com/start?session=[session.id]
```

- EndUrl:

```
https://tracking-west.example.com/end?session=[session.id]
```

MediaTailor が ADS にパラメータを渡す

AWS Elemental MediaTailor は、次の手順を使用して ADS への MediaTailor リクエストで動的変数の設定をサポートします。

- クエリパラメータでサポートされているフォーマットについては、「」を参照してください [MediaTailor パラメータのリファレンスと制限事項](#)。
- 設定エイリアスとドメイン変数については、「」を参照してください [MediaTailor 設定エイリアスの概要](#)。
- ADS リクエストのその他のカスタマイズについては、「」を参照してください [高度な使用法](#)。

セッションの初期化方法

MediaTailor は、セッションの初期化とパラメータの受け渡しに複数の方法をサポートしています。

1. リクエスト本文を含む POST:

```
POST <master>.m3u8
{
  "adsParams": {"param1": "value1", "param2": "value2"},
  "playerParams": {"param3": "value3"}
}
```

2. URL のクエリパラメータ:

```
GET <master>.m3u8?ads.param1=value1&ads.param2=value2&playerParams.param3=value3
```

Important

初期化時に指定できるパラメータは 1 回のみです。設定エイリアスは、転送前に実際の値に解決されます。

セッションとプレイヤーの情報を ADS に渡す

- ADS と連携して、広告クエリに応答するために必要な情報を決定します AWS Elemental MediaTailor。

- MediaTailor で、ADS 要件を満たすテンプレート ADS リクエスト URL を使用する設定を作成します。URL に、静的パラメータを含め、動的パラメータのプレースホルダーを含めます。設定の [Ad decision server (広告決定サーバー)] フィールドにテンプレートの URL を入力します。

以下のテンプレート URL の例では、correlation からセッションデータが渡され、deviceType からプレイヤーデータが渡されます。

```
https://my.ads.server.com/path?  
correlation=[session.id]&deviceType=[player_params.deviceType]
```

- プレーヤーで、プレーヤーデータのパラメータを渡すように AWS Elemental MediaTailor のセッション開始リクエストを設定します。セッション開始リクエストにお客様のパラメータを含め、セッションに対する後続のリクエストからそれらのパラメータを省きます。

プレーヤーがセッションを開始するために行う呼び出しのタイプによって、プレーヤー (クライアント) または MediaTailor (サーバー) のどちらがセッションの広告追跡レポートを提供するかが決まります。これらの 2 つのオプションについては、「[広告追跡データの報告](#)」を参照してください。

サーバー側とクライアント側のどちらの広告追跡レポートが必要かどうかに応じて、以下のいずれかのタイプの呼び出しを行います。どちらの例の呼び出しでも、userID は ADS を対象とし、auth_token はオリジンを対象としています。

- (オプション) サーバー側の広告追跡レポートの呼び出し - MediaTailor が ADS に送信するパラメータの前に ads を付けます。MediaTailor がオリジンサーバーに送信するパラメータの前には何も付けません。

次の例は、への HLS および DASH の受信リクエストを示しています AWS Elemental MediaTailor。MediaTailor は、ADS に対するリクエストでは deviceType、オリジンサーバーに対するリクエストでは auth_token を使用します。

HLS の例:

```
GET master.m3u8?ads.deviceType=ipad&auth_token=kjhdsaf7gh
```

DASH の例:

```
GET manifest.mpd?ads.deviceType=ipad&auth_token=kjhdsaf7gh
```

- (オプション) クライアント側の広告追跡レポートの呼び出し - adsParams オブジェクト内の ADS のパラメータを提供します。

HLS の例:

```
POST master.m3u8
{
  "adsParams": {
    "deviceType": "ipad"
  }
}
```

DASH の例:

```
POST manifest.mpd
{
  "adsParams": {
    "deviceType": "ipad"
  }
}
```

プレイヤーがセッションを開始すると、はテンプレート ADS リクエスト URL の変数をセッションデータとプレイヤーのadsパラメータ AWS Elemental MediaTailor に置き換えます。プレイヤーからの残りのパラメータは、オリジンサーバーに渡されます。

Example広告変数を使用した MediaTailor リクエスト

以下の例では、前のプレーヤーのセッション開始呼び出しの例に対応する、AWS Elemental MediaTailor から ADS とオリジンサーバーへの呼び出しを示しています。

- MediaTailor は、セッションデータとプレイヤーのデバイスタイプを使用して ADS を呼び出します。

```
https://my.ads.server.com/path?correlation=896976764&deviceType=ipad
```

- MediaTailor は、プレイヤーの認可トークンを使用してオリジンサーバーを呼び出します。
- HLS の例:

```
https://my.origin.server.com/master.m3u8?auth_token=kjhdsaf7gh
```

- DASH の例:

```
https://my.origin.server.com/manifest.mpd?auth_token=kjhdsaf7gh
```

高度な使用法

プレイヤーとセッションのデータを使用して、さまざまな方法で ADS リクエストをカスタマイズできます。ADS ホスト名のみを含める必要があります。

以下の例では、リクエストをカスタマイズできるいくつかの方法を示しています。

- プレイヤーパラメータとセッションパラメータを連結して新しいパラメータを作成します。例:

```
https://my.ads.com?key1=[player_params.value1][session.id]
```

- パス要素の一部としてプレイヤーパラメータを使用します。例:

```
https://my.ads.com/[player_params.path]?key=value
```

- プレイヤーパラメータを使用して、値だけではなくパス要素とキー自体の両方を渡します。例:

```
https://my.ads.com/[player_params.path]?[player_params.key1]=[player_params.value1]
```

ADS およびオリジンの MediaTailor パラメータルーティング

AWS Elemental MediaTailor は、プレフィックスと目的に基づいてクエリパラメータをさまざまな宛先にルーティングします。MediaPackage でのタイムシフト表示などのオリジン固有の機能を実装するには、パラメータルーティングを理解することが重要です。

パラメータルーティングルール

MediaTailor は、クエリパラメータに次のルーティングルールを使用します。

- オリジンパラメータ (プレフィックスなし): 特定のプレフィックスのないパラメータは、オリジン固有の機能のためにオリジンサーバーに渡されます。
- ADS パラメータ (**ads.** プレフィックス): プレフィックスが付いたパラメータ **ads.** が広告決定サーバーに送信されます

- マニフェストパラメータ (**manifest.** プレフィックス): プレフィックス が付いたパラメータ **manifest.** が CDN ルーティングと認可に使用されます

Exampleパラメータルーティングの例

次のセッション初期化は、パラメータルーティングを示しています。

```
POST /v1/session/123456789/originId/index.m3u8
{
  "adsParams": {
    "param1": "value1",
    "param2": "value2"
  },
  "manifestParams": {
    "auth_token": "abc123"
  }
}
```

この例では、以下のようにになっています：

- param1 と param2 は ADS に送信されます。
- auth_token は CDN ルーティングと認可に使用されます
- プレフィックスのないパラメータはオリジンサーバーに渡されます

オリジンサーバーパラメータの動作

オリジンサーバーに渡されるパラメータは、タイムシフト表示、コンテンツフィルタリング、認証などのオリジン固有の機能を有効にします。

一般的なオリジンパラメータのユースケース

オリジンパラメータは一般的に以下に使用されます。

- タイムシフト表示: MediaPackage タイムシフトコンテンツの start および end パラメータ
- コンテンツ認証: オリジンサーバーに必要な認証トークン
- コンテンツフィルタリング: 返されるコンテンツバリエーションを制御するパラメータ
- オリジン固有の機能: オリジンサーバーがコンテンツ処理に使用するパラメータ

⚠ Important

パラメータはセッションの初期化時に処理され、セッション全体を通して維持されます。タイムシフトウィンドウなどのパラメータを変更するには、更新された値で新しいセッションを作成する必要があります。

MediaTailor と MediaPackage のタイムシフト表示統合

AWS Elemental MediaTailor は、タイムシフト表示パラメータを MediaPackage オリジンに渡し、スタートオーバーおよびキャッチアップ表示機能を有効にできます。この統合により、ビューワーは以前の時点からライブコンテンツの視聴を開始できます。

MediaPackage のタイムシフト表示パラメータ

MediaPackage は、MediaTailor を介して渡すことができる次のタイムシフト表示パラメータをサポートしています。

- **start**: タイムシフトされたマニフェストの先頭を定義するエポックまたは ISO 8601 タイムスタンプ
- **end**: タイムシフトマニフェストの終了を定義するエポックまたは ISO 8601 タイムスタンプ
- **time_delay**: 指定した秒数でコンテンツの可用性を遅らせる
- **manifest_window_seconds**: 設定されたウィンドウより短いマニフェストをリクエストする

Example MediaPackage のタイムシフトされたパラメータを使用した MediaTailor MediaTailor セッションの初期化

次の例は、タイムシフト表示パラメータを使用してセッションを初期化する方法を示しています。

```
GET /v1/master/123456789/originId/index.m3u8?  
start=2024-08-26T10:00:00Z&end=2024-08-26T11:00:00Z
```

または、明示的なセッション初期化を使用します。

```
POST /v1/session/123456789/originId/index.m3u8  
{  
  "adsParams": {  
    "param1": "value1"
```

```
}  
}
```

追加のクエリパラメータを使用する場合:

```
?start=2024-08-26T10:00:00Z&end=2024-08-26T11:00:00Z
```

セッション中のパラメータ動作

タイムシフト表示パラメータには特定の動作特性があります。

- セッションの初期化: セッションの作成時にパラメータが処理されます
- パラメータの永続性: パラメータは再生中もセッションに関連付けられ続けます
- 初期化後のイミュータブル: アクティブなセッション中にパラメータを変更することはできません
- 新しいセッションが必要: タイムシフトウィンドウを変更するには、更新されたパラメータ値を使用して新しいセッションを作成します。

MediaPackage のスタートオーバーウィンドウの要件

MediaPackage でタイムシフト表示を使用するには、以下を確認してください。

1. MediaPackage エンドポイントでスタートオーバーウィンドウを設定する (最大 24 時間)
2. CDN が必要なクエリパラメータを MediaPackage に転送することを確認する
3. CDN キャッシュを改善するために、プレイヤーセッション間で一貫した再生ウィンドウを使用する
4. 開始時刻と終了時刻が設定されたスタートオーバーウィンドウ内に収まることを確認する

Important

タイムシフト表示を使用する場合は、各ビューワーに一貫の開始時刻または終了時刻を生成するのではなく、プレイヤーセッション間で一貫した再生ウィンドウを使用します。これにより、CDN でのキャッシュが改善され、スロットリングの可能性を回避できます。

MediaPackage のタイムシフト表示設定とパラメータの詳細については、AWS Elemental MediaPackage 「ユーザーガイド」の「[でのタイムシフト表示 AWS Elemental MediaPackage](#)」を参照してください。

MediaTailor パラメータセッションの動作と永続性

AWS Elemental MediaTailor はセッションの初期化時にパラメータを処理し、セッションライフサイクルを通じてパラメータを維持します。動的パラメータシナリオを実装するには、セッションの動作を理解することが不可欠です。

セッションの初期化方法

MediaTailor は、パラメータを使用してセッションを初期化するための複数の方法をサポートしています。

1. 暗黙的なセッション初期化: 最初のマニフェストリクエストに含まれるパラメータ

```
GET /v1/master/123456789/originId/index.m3u8?
manifest.auth_token=abc123&start=2024-08-26T10:00:00Z
```

2. 明示的セッション初期化 (POST): リクエスト本文で提供されるパラメータ

```
POST /v1/session/123456789/originId/index.m3u8
{
  "adsParams": {"param1": "value1"},
  "manifestParams": {"auth_token": "abc123"}
}
```

3. 明示的セッション初期化 (GET): クエリパラメータとして提供されるパラメータ

```
GET /v1/session/123456789/originId/index.m3u8?
ads.param1=value1&manifestParams.auth_token=abc123
```

パラメータの永続性とイミュータビリティ

MediaTailor パラメータの動作は、次のルールに従います。

- 1 回限りの仕様: パラメータはセッションの初期化時に 1 回のみ指定できます
- セッション全体の永続性: パラメータはセッション全体で維持されます
- 初期化後にイミュータブル: セッションの作成後にパラメータを変更することはできません
- 設定エイリアス解決: エイリアスは送信先に転送する前に実際の値に解決されます

パラメータ変更シナリオ

再生中にパラメータを変更するには:

- 新しいセッションを作成する: 更新されたパラメータ値を使用して新しいセッションを初期化する
- プレイヤーの移行: プレイヤーを新しいセッションにシームレスに移行する
- パラメータ継承: 変更されていないパラメータを転送して整合性を維持する

Exampleタイムシフトパラメータの変更

を 1 時間のウィンドウから 2 時間のウィンドウに変更するには:

1. 現在のセッション: `start=2024-08-26T10:00:00Z&end=2024-08-26T11:00:00Z`
2. 新しいセッションを作成する:
`start=2024-08-26T10:00:00Z&end=2024-08-26T12:00:00Z`
3. プレイヤーを新しいセッション URL に移行する

Important

1 つのセッションに対する複数のマルチバリエーションプレイリストリクエストは、最初のリクエストの後にパラメータを更新しません。パラメータはセッション期間中変更できません。

MediaTailor パラメータのリファレンスと制限事項

動的広告変数を設定する前に、すべての MediaTailor 設定に適用されるパラメータフォーマットの要件と制限を理解してください。

AWS Elemental MediaTailor は、マニフェストクエリパラメータと ADS パラメータの両方でパラメータ文字の制限、長さの制限、サポートされている形式に関する包括的な情報を提供します。

マニフェストクエリパラメータの文字制限

マニフェストクエリパラメータは特定の文字をサポートし、長さの制限が定義されています。

サポートされている文字 (URL エンコードなし)

マニフェストクエリパラメータでは、次の文字を直接使用できます。

- 英数字 (A~Z、a~z、0~9)
- 期間 (.)

- ハイフン (-)
- アンダースコア (_)
- バックスラッシュ (\)

URL エンコードでサポートされている文字

URL エンコードでは、次の特殊文字がサポートされています。

- 期間 (.) = %2E
- ダッシュ (-) = %2D
- アンダースコア (_) = %5F
- パーセント (%) = %25
- チルダ (~) = %7E
- スラッシュ (/) = %2F
- アスタリスク (*) = %2A
- 等号 (=) = %3D
- 質問 (?) = %3F

URL エンコードのサポート

MediaTailor は、URL エンコード (hello%20world = hello world など) で使用すると、パーセント (%) 記号をサポートします。HTTP 仕様に従って有効な URL エンコードである限り、任意の URL エンコード文字を使用できます。

サポートされていない文字

URL エンコードなしでマニフェストクエリパラメータに次の文字を使用することはできません：
:, ?, &, =%, , / (スラッシュ)。

Important

MediaTailor は、%%% や == などの二重文字をサポートしていません。文字制限のため、マニフェストクエリパラメータ値として完全な URLs を使用することはできません。

長さの制限

すべてのマニフェストクエリパラメータ (キーと値の組み合わせ) の合計長は 2000 文字を超えることはできません。

ADS パラメータの長さの制限

ADS へのリクエストで使用されるパラメータには、次の長さ制限が適用されます。

- ADS パラメータ名: 最大 10,000 文字
- ADS パラメータ値: 最大 25,000 文字
- ADS URL: 最大 25,000 文字

MediaTailor パラメータのトラブルシューティングガイド

AWS Elemental MediaTailor は、文字制限、URL エンコードの問題、設定エイリアスエラーなど、MediaTailor の一般的なパラメータ関連の問題をトラブルシューティングするためのガイダンスを提供します。

文字制限エラー

サポートされていない文字を含むパラメータ値は、エラーや予期しない動作を引き起こす可能性があります。

一般的な症状

次の症状は、文字制限の問題を示している可能性があります。

- マニフェスト URLs に表示されないパラメータ
- セッションの初期化中の HTTP 400 エラー
- 切り捨てられたパラメータ値または破損したパラメータ値
- 不正な形式の URLs が原因で ADS リクエストが失敗する

解決の手順

文字制限エラーを解決するには:

1. サポートされていない文字のパラメータ値を確認する: :、?、&、=、%、/
2. 特殊文字に適切な URL エンコードを適用する (「」を参照 [MediaTailor パラメータのリファレンスと制限事項](#))

3. %% や などの二重文字は避けてください。 ==

4. 完全な URLs

Example URL エンコードの例

以下を使用する代わりに、以下を使用します。

```
manifest.redirect_url=https://example.com/path?param=value
```

URL エンコード形式を使用します。

```
manifest.redirect_url=https%3A%2F%2Fexample.com%2Fpath%3Fparam%3Dvalue
```

長さ制限エラー

長さ制限を超えるパラメータは切り捨てられたり、エラーが発生する可能性があります。

長さの制限

次の長さ制限が適用されます (詳細については[MediaTailor パラメータのリファレンスと制限事項](#)、「」を参照してください)。

- マニフェストクエリパラメータ (合計): 2000 文字
- ADS パラメータ名: 10,000 文字
- ADS パラメータ値: 25,000 文字
- ADS URLs: 25,000 文字

解決戦略

長さの制限を処理するには:

1. 可能な場合は、より短いパラメータ名と値を使用する
2. 大きなパラメータ値を複数の小さなパラメータに分割する
3. 設定エイリアスを使用して短いエイリアスを長い値にマッピングする (「」を参照[MediaTailor 設定エイリアスの概要](#))
4. パラメータリファレンスを含む大規模なデータには外部ストレージを使用することを検討する

設定エイリアスエラー

設定エイリアスの問題により、HTTP 400 エラーまたは予期しないパラメータ値が発生する可能性があります。

一般的な設定エイリアスエラー

設定エイリアスでは、一般的に次のエラーが発生します。

- HTTP 400 エラー: エイリアス値がないか、無効です
- ドメイン変数が正しく解決されない
- プレイヤーパラメータがエイリアス値に置き換えられていない

解決策チェックリスト

設定エイリアスエラーを解決するには:

1. すべてのドメイン変数が として定義されていることを確認します。 ConfigurationAliases
2. プレイヤーパラメータ変数が player_params プレフィックスを使用していることを確認する
3. 重要な URLs のドメイン変数 (VideoContentSourceUrl、AdSegmentUrlPrefix、ContentSegmentUrlPrefix) のエイリアス化された値のリストが網羅的であることを確認します。
4. セッション初期化リクエストで有効なエイリアス値が指定されていることを確認します。
5. ConfigurationAliases パラメータの JSON 構造を検証する

トラブルシューティングの詳細なガイダンスについては、「」を参照してください [MediaTailor 設定エイリアスのトラブルシューティングガイド](#)。

Example 設定エイリアスの検証

設定に必要なエイリアスがすべて含まれていることを確認します。

```
"ConfigurationAliases": {
  "player_params.origin_domain": {
    "pdx": "abc.mediapackage.us-west-2.amazonaws.com",
    "iad": "xyz.mediapackage.us-east-1.amazonaws.com"
    // Must include all possible values used in session initialization
  }
}
```



```
}
```

パラメータ処理フローの問題

パラメータ処理フローを理解すると、パラメータの転送と変換に関する問題のトラブルシューティングに役立ちます。

パラメータ処理順序

MediaTailor は、次の順序でパラメータを処理します。

1. セッション初期化パラメータの検証
2. 設定エイリアスの解決 (該当する場合)
3. パラメータフィルタリング (ADS とオリジンとマニフェスト)
4. URL エンコードとフォーマット
5. URLsへのパラメータアプリケーション

デバッグパラメータフロー

パラメータ処理の問題をデバッグするには:

1. セッションの初期化でパラメータが正しく指定されていることを確認する
2. 設定エイリアスが期待値に解決することを確認する
3. パラメータが正しい URLs (マニフェスト、ADS、オリジン) に表示されることを確認する
4. URL エンコーディングが正しく適用されていることを確認する

Exampleパラメータフローの例

セッションの初期化:

```
POST master.m3u8
{
  "playerParams": {"origin_domain": "pdx"},
  "manifestParams": {"test": "123"}
}
```

エイリアスの解決と処理後:

- オリジンリクエスト: `https://abc.mediapackage.us-west-2.amazonaws.com/out/v1/abcd`
- マニフェスト URL: `/v1/master/.../index.m3u8?aws.sessionId=session&test=123`

セキュリティに関する考慮事項とベストプラクティス

MediaTailor は、一般的なセキュリティ問題を防ぐために、パラメータ処理のためのセキュリティ対策を実装します。

セキュリティ対策

MediaTailor は、次のセキュリティ対策を実装します。

1. データベースの肥大化を防ぐための入力サイズの制限
2. ユーザー入力の適切なエンコードとサニタイズ
3. レスポンスの破損を防ぐための入力の URL エンコード

ベストプラクティス

安全なパラメータ処理については、次のベストプラクティスに従ってください。

- 送信前にクライアント側でパラメータ値を検証する
- 設定エイリアスを使用して、可能なパラメータ値を制限する
- パラメータに機密情報を含めないようにする
- 異常なパターンがないかパラメータの使用状況をモニタリングする
- パラメータ値を推奨長さ制限内に維持する

MediaTailor 設定エイリアスのトラブルシューティングガイド

AWS Elemental MediaTailor は、一般的な設定エイリアスの問題とエラーシナリオに関する体系的なトラブルシューティングガイダンスを提供します。

設定エイリアスの検証エラー

設定エイリアスがないか無効である場合、MediaTailor は問題の特定に役立つ特定のエラーレスポンスを返します。

一般的なエラーシナリオ

次の表に、一般的な設定エイリアスエラーとその解決手順を示します。

エラー	原因	解決策
HTTP 400: 無効なプレイヤーパラメータエイリアス	ConfigurationAliases にプレイヤーパラメータ値が見つかりません	対応する ConfigurationAliases マッピングにプレイヤーパラメータ値がキーとして存在することを確認します。
HTTP 400: 必要な設定エイリアスがありません	対応する ConfigurationAliases エントリなしで使用されるドメイン変数	必要なすべてのエイリアスマッピングを使用して、欠落しているプレイヤーパラメータを ConfigurationAliases に追加する
HTTP 400: 設定の検証に失敗しました	ConfigurationAliases 構造が正しくないか、不完全です	JSON 構造を検証し、すべてのドメイン変数に対応するエイリアスがあることを確認する
URLs の空の文字列置換	ドメイン以外の変数エイリアスが見つかりません	ConfigurationAliases で欠落しているエイリアスマッピングを追加するか、デフォルト値を指定する

検証チェックリスト

次のチェックリストを使用して、設定エイリアスの設定を検証します。

1. ドメイン変数カバレッジ: URLs のドメイン部分で使用されるすべての変数に対応する ConfigurationAliases エントリがあることを確認します。
2. エイリアスの完全性: 可能なすべてのプレイヤーパラメータ値がエイリアスマッピングのキーとして含まれていることを確認します。
3. JSON 構造: ConfigurationAliases JSON が適切にフォーマットされ、ネストされていることを確認する
4. パラメータの命名: すべてのプレイヤーパラメータで player_params. プレフィックスが使用されていることを確認します。
5. 値の整合性: エイリアス値が意図した用途 (URLs、プロファイル名など) に対して有効であることを確認します。

設定エイリアスの解決のデバッグ

この体系的なアプローチに従って、設定エイリアス解決の問題をデバッグします。

Step-by-stepのデバッグ方法

次の手順を使用して、設定エイリアスの問題を特定して解決します。

設定エイリアスのデバッグ手順

1. 設定構造の検証: 再生設定に正しくフォーマットされた ConfigurationAliases が含まれていることを確認します。

```
{
  "ConfigurationAliases": {
    "player_params.example_param": {
      "alias1": "value1",
      "alias2": "value2"
    }
  }
}
```

2. プレイヤーパラメータの形式を確認する: セッションの初期化に正しくフォーマットされたプレイヤーパラメータが含まれていることを確認します。

```
{
  "playerParams": {
    "example_param": "alias1"
  }
}
```

3. エイリアスマッピングの検証: プレイヤーパラメータ値 ("alias1") が ConfigurationAliases マッピングのキーとして存在することを確認します。
4. シンプルな設定でテストする: 最小限の設定から始めて問題を分離する
5. エラーレスポンスのモニタリング: 特定の検証メッセージの MediaTailor エラーレスポンスを確認する
6. 解決URLs の検証: 最終的な解決済み URLsが有効でアクセス可能であることを確認します。

設定エイリアスのベストプラクティス

次のベストプラクティスに従って、信頼性の高い設定エイリアスの実装を確保します。

セキュリティに関する考慮事項

設定エイリアスを使用する場合は、次のセキュリティ対策を実装します。

- 入力検証: エイリアス解決で使用する前に、すべてのプレイヤーパラメータ値を検証する
- エイリアス値のサニタイズ: エイリアス値に想定される文字と形式のみが含まれていることを確認します。
- ドメインの制限: ドメインエイリアスを信頼できる制御されたドメインに制限する
- アクセスコントロール: 承認された担当者だけに設定変更を制限する

パフォーマンスの最適化

次の推奨事項を使用して、設定エイリアスのパフォーマンスを最適化します。

- エイリアス数を最小限に抑える: 必要なエイリアスのみを使用して処理オーバーヘッドを削減する
- 効率的な命名: エイリアスとパラメータに明確で一貫した命名規則を使用する
- デフォルト値: 一般的なユースケースに適したデフォルトエイリアスを指定します。
- 設定キャッシュ: MediaTailor の設定キャッシュを活用してパフォーマンスを向上させる

メンテナンスとモニタリング

次のプラクティスを使用して、信頼性の高い設定エイリアスオペレーションを維持します。

- 定期的な検証: すべてのエイリアスマッピングが最新で機能していることを定期的に検証する
- エラーモニタリング: 欠落または無効なエイリアスに関連する HTTP 400 エラーをモニタリングする
- ドキュメント: すべてのエイリアスマッピングとその目的を明確に文書化する
- テスト手順: すべてのエイリアスの組み合わせに対して包括的なテストを実装する

MediaTailor マニフェストクエリパラメータ

AWS Elemental MediaTailor は、CDN ルーティングと認可用のマニフェストクエリパラメータ、オリジン固有の機能に使用できるその他のクエリパラメータなど、さまざまな目的でクエリパラメータを処理します。

AWS Elemental MediaTailor はセッション初期化からのクエリパラメータを保持し、パーソナライズされたマニフェスト URLs やその他のアセットに追加します。この機能は、MediaTailor とクライアントプレーヤーの間にコンテンツ配信ネットワーク (CDN) がある場合に使用します。

CDN が以下のクエリパラメータを必要とする場合は、マニフェストクエリパラメータを使用します。

- さまざまな MediaTailor エンドポイントへの動的ルーティング
- トークン認可

クライアント側と CDN の動作

MediaTailor はクライアント側のレポートエンドポイントにクエリパラメータを追加しますが、CDN セグメントには追加しません。更新された機能により、さまざまな MediaTailor アセットのクエリパラメータをより包括的にサポートできるため、CDN ルーティングと認可のユースケースの柔軟性が向上します。

MediaTailor はクライアント側のレポートエンドポイントにクエリパラメータを追加しますが、CloudFront (または他の CDN) セグメントにはクエリパラメータを追加しません。

パラメータの保存を使用するには、[AWS サポート](#)に連絡して、マニフェストクエリパラメータのパスルーを有効にするようにリクエストしてください。

動作は、HLS と DASH、および明示的なセッション初期化と暗黙的なセッション初期化によって異なります。以下のトピックでは、MediaTailor がマニフェストにパラメータを渡すようにセッション初期化リクエストを設定する方法について説明します。

オリジンの MediaTailor クエリパラメータ処理

AWS Elemental MediaTailor は、クエリパラメータの目的に応じて異なる方法でクエリパラメータを処理します。マニフェストクエリパラメータ (プレフィックスは manifest.) は CDN のルーティングと認可に使用されますが、オリジン固有の機能には他のクエリパラメータを使用できます。

MediaPackage によるタイムシフト表示

MediaPackage のタイムシフト表示機能では、リクエストに start および end パラメータを含めることができます。これらのパラメータは、スタートオーバーとキャッチアップ表示の特定のコンテンツウィンドウを定義します。

Exampleタイムシフト表示リクエスト

タイムシフト表示のマニフェストリクエストに開始パラメータと終了パラメータを含めます。

```
GET /v1/master/111122223333/originId/index.m3u8?  
start=2024-08-26T10:00:00Z&end=2024-08-26T11:00:00Z
```

セッション中のパラメータ動作

クエリパラメータはセッションの初期化時に処理されます。タイムシフト表示またはその他のオリジン固有の機能の場合:

- セッションの初期化: 最初のマニフェストリクエストに必要なパラメータを含める
- パラメータの永続性: パラメータはセッションに関連付けられ、再生中も維持されます。
- パラメータの変更: タイムシフトウィンドウやその他のパラメータを変更するには、更新された値を使用して新しいセッションを作成します。

Important

クエリパラメータの特定の処理は、オリジン設定とコンテンツオリジンがサポートするパラメータによって異なります。MediaPackage 統合の場合、の説明に従って必要なクエリパラメータを転送するように CDN が設定されていることを確認します [必須クエリパラメータを設定する](#)。

MediaTailor パラメータの文字制限と URL エンコード

AWS Elemental MediaTailor は、マニフェストクエリパラメータで特定の文字をサポートします。特殊文字には URL エンコードを使用できます。

URL エンコードでサポートされている文字

URL エンコードでは、次の特殊文字がサポートされています。

- 期間 (.) = %2E
- ダッシュ (-) = %2D
- アンダースコア (_) = %5F

- パーセント (%) = %25
- チルダ (~) = %7E
- スラッシュ (/) = %2F
- アスタリスク (*) = %2A
- 等しい (=) = %3D
- 質問 (?) = %3F

URL エンコードのサポート

MediaTailor は、URL エンコード (hello%20world = hello world など) で使用すると、パーセント (%) 記号をサポートします。HTTP 仕様に従って有効な URL エンコードである限り、任意の URL エンコード文字を使用できます。

Important

MediaTailor は、%%% や == などの二重文字をサポートしていません。

セキュリティに関する考慮事項

MediaTailor は、パラメータ処理に次のセキュリティ対策を実装します。

1. データベースの肥大化を防ぐための入力サイズの制限
2. ユーザー入力の適切なエンコードとサニタイズ
3. レスポンスの破損を防ぐための入力の URL エンコード

トピック

- [MediaTailor HLS 暗黙的なセッションの初期化](#)
- [MediaTailor DASH 暗黙的なセッションの初期化](#)
- [MediaTailor HLS と DASH の明示的なセッション初期化](#)
- [MediaTailor プロトコル固有のパラメータの動作](#)
- [MediaTailor CDN 統合とパラメータルーティング](#)

MediaTailor HLS 暗黙的なセッションの初期化

AWS Elemental MediaTailor リクエストにキー を持つクエリパラメータが含まれている場合、 は MediaTailor リソースへのリンクにクエリパラメータを含めますmanifest.*。次の例は、このリクエスト形式を示しています。

```
GET /v1/master/111122223333/originId/index.m3u8?manifest.test=123&other=456
```

リンクには manifest. プレフィックスは含まれません。

HLS のパラメータアプリケーション

HLS 暗黙的なセッションの場合、MediaTailor はマニフェスト階層の次の場所にパラメータを適用します。

- 多変量プレイリスト URLs
- メディアプレイリスト URLs
- コンテンツセグメント URLs
- 広告セグメント URLs
- HLS URLs

Exampleマルチバリエーションプレイリスト

次の例は、MediaTailor が多変量プレイリストの URL にクエリパラメータを含める方法を示しています。

```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-INDEPENDENT-SEGMENTS
#EXT-X-MEDIA:LANGUAGE="eng",AUTOSELECT=YES,FORCED=NO,TYPE=SUBTITLES,URI=".../.../
manifest/111122223333/originId/session/1.m3u8?test=123",GROUP-
ID="subtitles",DEFAULT=YES,NAME="caption_1"
#EXT-X-STREAM-INF:CODECS="avc1.640029,mp4a.40.2",AVERAGE-
BANDWIDTH=2525657,RESOLUTION=960x540,SUBTITLES="subtitles",FRAME-
RATE=29.97,BANDWIDTH=2665212
.../.../manifest/111122223333/originId/session/0.m3u8?test=123
```

Exampleメディアプレイリスト

次の例は、MediaTailor がコンテンツセグメントの URLs にクエリパラメータを含める方法を示しています。

```
#EXTM3U
#EXT-X-VERSION:6
#EXT-X-TARGETDURATION:7
#EXT-X-MEDIA-SEQUENCE:28716269
#EXT-X-DISCONTINUITY-SEQUENCE:0
#EXTINF:6.006,
https://origin.com/contentSegment_1.ts?originQueryParam=foo&test=123
#EXT-X-DISCONTINUITY
#EXTINF:6.006,
../../../../segment/111122223333/originId/session/0/2?test=123
```

MediaTailor DASH 暗黙的なセッションの初期化

AWS Elemental MediaTailor はクライアントのセッションを作成し、クライアントがセッションなしでマニフェストリクエストを行うときにクエリパラメータでリダイレクトします。次の例は、このリクエスト形式を示しています。

```
GET /v1/dash/111122223333/originId/index.mpd?manifest.test=123&other=456
```

MediaTailor はクライアントのセッションを作成し、クエリパラメータを使用してリダイレクトします。

```
/v1/dash/111122223333/originId/index.mpd?sessionId=session&test=123
```

DASH のパラメータアプリケーション

DASH マニフェストレスポンスには、コンテンツセグメント、広告セグメント、初期化 URLs など、さまざまな場所のクエリパラメータが含まれます。MediaTailor は、以下のパラメータを適用します。

- DASH マニフェストの場所要素
- SegmentTemplate 初期化属性
- SegmentTemplate メディア属性

- コンテンツセグメント URLs
- 広告セグメント URLs

クライアントがリクエストを行うと、MediaTailor は次の例のような DASH マニフェストで応答します。最初の期間はコンテンツ期間であるため、MediaTailor はそこにマニフェストクエリパラメータを挿入しません。広告期間である 2 番目の期間では、MediaTailor はマニフェストクエリパラメータを SegmentTemplate 要素の initialization 属性と media 属性に挿入します。Location 要素には、マニフェストクエリパラメータもあります。

```
<?xml version="1.0" encoding="UTF-8"?>
<MPD availabilityStartTime="2018-07-27T09:48:23.634000+00:00"
  id="201" minBufferTime="PT30S" minimumUpdatePeriod="PT15S"
  profiles="urn:mpeg:dash:profile:isoff-live:2011" publishTime="2023-02-14T23:37:43"
  suggestedPresentationDelay="PT25.000S" timeShiftBufferDepth="PT56.997S" type="dynamic"
  xmlns="urn:mpeg:dash:schema:mpd:2011" xmlns:scte35="urn:scte:scte35:2013:xml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 http://standards.iso.org/ittf/
PubliclyAvailableStandards/MPEG-DASH_schema_files/DASH-MPD.xsd">
  <BaseURL>https://origin.com/contentSegments/</BaseURL>
  <Location>https://mediatailor.com/v1/dash/111122223333/originId/index.mpd?
test=123&aws.sessionId=session</Location>
  <Period duration="PT29.963S" id="28737823" start="PT143732873.178S">
    <AdaptationSet bitstreamSwitching="true" mimeType="video/mp4"
segmentAlignment="true" startWithSAP="1" subsegmentAlignment="true"
subsegmentStartsWithSAP="1">
      <Representation bandwidth="2200000" codecs="avc1.640029"
frameRate="30000/1001" height="540" id="1" width="960">
        <SegmentTemplate initialization="index_video_7_0_init.mp4"
m=1611174111" media="index_video_7_0_${Number$.mp4?m=1611174111"
presentationTimeOffset="4311986195351" startNumber="28737828" timescale="30000">
          <SegmentTimeline>
            <S d="180180" t="4311986911066"/>
            <S d="3003" t="4311987091246"/>
          </SegmentTimeline>
        </SegmentTemplate>
      </Representation>
    </AdaptationSet>
  </Period>
  <Period id="28737829_1" start="PT39925H48M23.141S">
    <BaseURL>https://mediatailor.com/v1/
dashsegment/111122223333/originId/session/28737829/28737829_1/</BaseURL>
```

```

    <AdaptationSet bitstreamSwitching="false" frameRate="30000/1001"
    mimeType="video/mp4" segmentAlignment="true" startWithSAP="1"
    subsegmentAlignment="true" subsegmentStartsWithSAP="1">
      <SegmentTemplate startNumber="1" timescale="90000"/>
      <Representation bandwidth="2200000" codecs="avc1.64001f" height="540"
      id="1" width="960">
        <SegmentTemplate initialization="asset_540_2_0init.mp4?test=123"
        media="asset_540_2_0_${Number}%09d$.mp4?test=123" startNumber="1" timescale="90000">
          <SegmentTimeline>
            <S d="180180" r="6" t="0"/>
            <S d="87087" t="1261260"/>
          </SegmentTimeline>
        </SegmentTemplate>
      </Representation>
    </AdaptationSet>
  </Period>
</MPD>

```

MediaTailor HLS と DASH の明示的なセッション初期化

AWS Elemental MediaTailor クライアントが明示的なセッション初期化リクエストを行うと、はクエリパラメータmanifestParamsとして多変量プレイリストに、追跡 URLsに含めます。

セッションの初期化方法

明示的なセッション初期化では、リクエスト本文で POST を使用するか、クエリパラメータで GET を使用できます。

1. リクエスト本文を含む POST:

```

POST /v1/session/111122223333/originId/index.m3u8
{
  "adsParams": {"param1": "value1", "param2": "value2", "param3": "value3"},
  "manifestParams": {"test": "123"}
}

```

2. クエリパラメータを使用した GET:

```

GET /v1/session/111122223333/originId/index.m3u8?
ads.param1=value1&ads.param2=value2&manifestParams.test=123

```

Exampleセッション初期化リクエスト

```
POST /v1/session/111122223333/originId/index.m3u8
{
  "adsParams": {
    "param1": "value1",
    "param2": "value2",
    "param3": "value3"
  },
  "manifestParams": {
    "test": "123"
  },
  "reportingMode": "client"
}
```

Exampleマニフェストと追跡レスポンス

```
{
  "manifestUrl": "/v1/master/111122223333/originId/index.m3u8?
aws.sessionId=session&test=123",
  "trackingUrl": "/v1/tracking/111122223333/originId/session?test=123"
}
```

セッションのマニフェストレスポンスには、前述の暗黙的なセッション初期化ワークフローと同様のMediaTailor URLs manifestParamsに固有の **があります**。主な違いは、明示的なセッション初期化のマニフェストパラメータが **で始まらない**ことですmanifest..

マニフェストクエリパラメータはイミュータブルであり、セッションの初期化時にのみ設定できます。クライアントが1つのセッションに対して複数のマルチバリエーションプレイリストリクエストを行う場合、MediaTailorは最初のリクエスト後にマニフェストクエリパラメータを更新しません。

パラメータ処理フロー

初期化時に指定できるパラメータは1回のみです。設定エイリアスは、転送前に実際の値に解決されます。例: ConfigurationAliases 設定ad_type=abc12345に基づいて **を**にplayer_params.ad_type=customized解決します。

MediaTailor プロトコル固有のパラメータの動作

AWS Elemental MediaTailor は、HLS プロトコルと DASH プロトコルでマニフェストクエリパラメータを異なる方法で処理します。各プロトコルタイプには、特定のアプリケーションの場所と処理方法があります。

HLS と DASH のパラメータ処理の比較

次の表は、MediaTailor が HLS および DASH プロトコル間でマニフェストクエリパラメータを処理する方法を比較しています。

側面	HLS の動作	DASH の動作
パラメータアプリケーション	マニフェスト URLs とセグメント URLs に直接適用	Location 要素、SegmentTemplate 属性、セグメント URLs に適用
マニフェスト階層	多変量プレイリスト → メディアプレイリスト → セグメント	MPD → 期間 → AdaptationSets → 表現
初期化 URLs	存在する場合は HLS init URLs に適用されます	SegmentTemplate 初期化属性に適用
セッション処理	プレイリストの更新全体で保持されるパラメータ	セッション継続性の MPD Location 要素に含まれるパラメータ
広告セグメントの処理	メディアプレイリストの広告セグメント URLs に適用	広告期間 SegmentTemplate メディア属性に適用

パラメータアプリケーションの場所

MediaTailor は、マニフェストクエリパラメータを次の場所に適用します。

HLS パラメータアプリケーション

HLS ストリームの場合、MediaTailor はマニフェストクエリパラメータを以下に適用します。

- 多変量プレイリスト URLs: パラメータがメディアプレイリストリファレンスに追加されます
- メディアプレイリスト URLs: パラメータはメディアプレイリスト内のセグメント URLs に含まれます

- コンテンツセグメント URLs: すべてのコンテンツセグメントにマニフェストクエリパラメータが含まれます
- 広告セグメント URLs: 広告セグメントは CDN ルーティングと認可のパラメータを受け取ります
- HLS 初期化 URLs: Init セグメントには、ストリームに存在する場合にパラメータが含まれます
- スレートセグメント URLs: スレートコンテンツには、一貫した CDN 動作のためのパラメータが含まれています

Example HLS パラメータアプリケーションの例

セッションの初期化を前提とすると、次のようになります。

```
GET /v1/master/123456789/originId/index.m3u8?  
manifest.auth_token=abc123&manifest.region=us-west
```

多変量プレイリストには、メディアプレイリストリファレンスにパラメータが含まれます。

```
#EXTM3U  
#EXT-X-VERSION:3  
#EXT-X-STREAM-INF:BANDWIDTH=2665212,RESOLUTION=960x540  
../../../manifest/123456789/originId/session/0.m3u8?auth_token=abc123&region=us-west
```

メディアプレイリストには、セグメント URLs にパラメータが含まれます。

```
#EXTM3U  
#EXT-X-VERSION:6  
#EXT-X-TARGETDURATION:7  
#EXTINF:6.006,  
https://origin.com/segment1.ts?auth_token=abc123&region=us-west  
#EXTINF:6.006,  
../../../segment/123456789/originId/session/0/2?auth_token=abc123&region=us-west
```

DASH パラメータアプリケーション

DASH ストリームの場合、MediaTailor はマニフェストクエリパラメータを以下に適用します。

- MPD Location 要素: Location 要素には、マニフェスト更新リクエストのパラメータが含まれます
- SegmentTemplate 初期化属性: Init セグメント URLs パラメータが含まれます
- SegmentTemplate メディア属性: メディアセグメント URL テンプレートにはパラメータが含まれます

- コンテンツセグメント URLs: テンプレートから生成されたすべてのコンテンツセグメントにパラメータが含まれます
- 広告セグメント URLs: 広告期間セグメントには CDN 統合のパラメータが含まれます
- サーバー側のレポートリダイレクト: 広告セグメントへの 302 リダイレクトがパラメータを保持

Example DASH パラメータアプリケーションの例

セッションの初期化を前提とすると、次のようになります。

```
GET /v1/dash/123456789/originId/index.mpd?  
manifest.auth_token=abc123&manifest.region=us-west
```

DASH マニフェストには、複数の場所にパラメータが含まれています。

```
<MPD>  
  <Location>https://mediatailor.com/v1/dash/123456789/originId/index.mpd?  
auth_token=abc123&region=us-west&aws.sessionId=session</Location>  
  <Period>  
    <AdaptationSet>  
      <Representation>  
        <SegmentTemplate  
          initialization="init.mp4?auth_token=abc123&region=us-west"  
          media="segment_${Number$.mp4?auth_token=abc123&region=us-west"/>  
        </Representation>  
      </AdaptationSet>  
    </Period>  
</MPD>
```

MediaTailor CDN 統合とパラメータルーティング

AWS Elemental MediaTailor マニフェストクエリパラメータは、高度な CDN 統合シナリオを可能にします。動的ルーティング、認可、ロードバランシングに使用できます。

CDN ルーティングのユースケース

マニフェストクエリパラメータの恩恵を受ける一般的な CDN 統合シナリオは次のとおりです。

- 地理的ルーティング: ビューワーの場所に基づいてリージョン固有の MediaTailor エンドポイントにリクエストをルーティングする

- トークンベースの認可: CDN 経由で認可トークンを MediaTailor に渡して、コンテンツに安全にアクセスする
- ロードバランシング: CDN ルーティングロジックを使用して複数の MediaTailor エンドポイントにトラフィックを分散する
- A/B テスト: テストのために異なるユーザーセグメントを異なる MediaTailor 設定にルーティングする
- デバイス固有の最適化: デバイスタイプまたは機能に基づいてリクエストをルーティングする

CDN レイヤー間のパラメータの保存

MediaTailor は、マニフェストクエリパラメータが複数の CDN レイヤーとリクエストタイプにわたって保持されるようにします。

1. 初期リクエスト: パラメータはセッション初期化リクエストから抽出されます
2. マニフェスト生成: パラメータはマニフェスト内のすべての関連 URLs に適用されます
3. セグメントリクエスト: パラメータは、一貫した CDN 動作のためにすべてのセグメント URLs に含まれます
4. 広告挿入: パラメータは広告挿入時とセグメント置換時に保持されます

Example CDN 認可フロー

次の例は、マニフェストクエリパラメータを使用した完全な CDN 認可フローを示しています。

1. 認可トークンを含むクライアントリクエストマニフェスト:

```
GET https://cdn.example.com/mediatailor/v1/master/123456789/originId/index.m3u8?manifest.auth_token=jwt_token_here&manifest.user_id=12345
```

2. CDN は、パラメータを使用して MediaTailor にリクエストを転送します。

```
GET https://mediatailor.amazonaws.com/v1/master/123456789/originId/index.m3u8?manifest.auth_token=jwt_token_here&manifest.user_id=12345
```

3. MediaTailor は、すべての URLs に適用されるパラメータを使用してマニフェストを生成します。

```
#EXTM3U
#EXT-X-STREAM-INF:BANDWIDTH=2665212
```

```
../../../../manifest/123456789/originId/session/0.m3u8?  
auth_token=jwt_token_here&user_id=12345
```

4. 後続のセグメントリクエストには、CDN 認可のパラメータが含まれます。

```
GET https://cdn.example.com/mediatailor/segment/123456789/originId/session/0/1?  
auth_token=jwt_token_here&user_id=12345
```

広告追跡データの報告

MediaTailor には、視聴者が視聴した広告の量を追跡およびレポートするための 2 つのオプションがあります。サーバー側の広告レポートアプローチでは、MediaTailor は広告を追跡し、ビーコン (追跡シグナル) を広告サーバーに直接送信します。または、クライアント側の追跡アプローチでは、クライアントプレーヤー (ユーザーのデバイス) が広告を追跡し、ビーコンを広告サーバーに送信します。再生セッションで使用される広告レポートのタイプは、プレーヤーが MediaTailor でセッションを開始するために行う特定のリクエストによって異なります。

動変数を使用してセッションデータとプレーヤーデータを広告サーバーに渡す方法については、「」を参照してください[ADS リクエストの MediaTailor 動変数](#)。セッション初期化パラメータの詳細については、「」を参照してください[MediaTailor マニフェストクエリパラメータ](#)。

トピック

- [MediaTailor サーバー側の広告の追跡とレポート](#)
- [クライアント側の広告追跡](#)

MediaTailor サーバー側の広告の追跡とレポート

AWS Elemental MediaTailor は、包括的な広告の追跡と測定のためのサーバー側のレポートにデフォルト設定されています。サーバー側のレポートでは、プレーヤーがマニフェストからの広告 URL をリクエストするときに、サービスが広告消費を広告追跡 URL に直接報告します。プレーヤーが MediaTailor との再生セッションを開始したら、サーバー側のレポートを実行するためにユーザーまたはプレーヤーからの追加入力が必要になることはありません。MediaTailor は、各広告が再生されるたびに広告サーバーにビーコンを送信して、広告がどのくらい閲覧されたのかを報告します。MediaTailor は、広告の開始時点と、広告の進行における四分位点 (第 1 分位点、中間点、第 3 分位点、および広告の終了時点) に関するビーコンを送信します。

サーバー側の広告レポートを実行する

- プレイヤーから、プロトコルに従った以下のいずれかの形式を使用して、新しい MediaTailor 再生セッションを開始します。
- 例: HLS 形式

```
GET <mediatailorURL>/v1/master/<hashed-account-id>/<origin-id>/<asset-id>?  
ads.<key-value-pairs-for-ads>&<key-value-pairs-for-origin-server>
```

- 例: DASH 形式

```
GET <mediatailorURL>/v1/dash/<hashed-account-id>/<origin-id>/<asset-id>?ads.<key-  
value-pairs-for-ads>&<key-value-pairs-for-origin-server>
```

キーバリューペアは、広告追跡用の動的なターゲティングパラメータです。リクエストへのパラメータの追加については、「[the section called “動的広告変数”](#)」を参照してください。

AWS Elemental MediaTailor はマニフェスト URL を使用してリクエストに応答します。マニフェストには、メディアマニフェストの URL が含まれています。メディアマニフェストには、広告セグメントリクエストに対する埋め込みリンクが含まれています。

Note

MediaTailor が追跡 URL で二重スラッシュ (//) を検出すると、スラッシュは 1 (/) に折りたたまれます。

プレイヤーが広告セグメント URL (/v1/segment パス) に再生をリクエストすると、AWS Elemental MediaTailor は該当するビーコンを広告追跡 URL を介して広告サーバーに送信します。それと同時に、サービスが実際の *.ts 広告セグメントへのリダイレクトを発行します。広告セグメントは、MediaTailor がトランスコードされた広告を保存する Amazon CloudFront ディストリビューション、または広告をキャッシュしたコンテンツ配信ネットワーク (CDN) のいずれかにあります。

以下のセクションでは、MediaTailor からのサーバー側の広告追跡の操作について詳しく説明します。

トピック

- [サーバーガイド広告挿入によるサーバー側の追跡 \(SGAI\)](#)
- [サーバー側の追跡ビーコン用語集](#)
- [サーバー側の追跡タイミングとキャッシュ動作](#)
- [MediaTailor サーバー側の追跡機能](#)

サーバーガイド広告挿入によるサーバー側の追跡 (SGAI)

サーバーガイド広告挿入 (SGAI) を使用する場合、サーバー側の追跡では、上記のステッチモードアプローチとは異なるセッションレスビーコンメカニズムが使用されます。MediaTailor が広告セグメントをコンテンツマニフェスト (/v1/segment リクエストを追跡する場所) にステッチする代わりに、SGAI は広告参照を、広告 URIs。

セッションレスサーバー側のビーコンの仕組み

以下の手順では、SGAI セッションでサーバー側のビーコンがどのように機能するかについて説明します。

1. セッションの初期化: プレイヤーは を使用して HLS マルチバリエーションプレイリストをリクエストします `aws.insertionMode=GUIDED`。サーバー側のレポートはデフォルトです (`aws.reportingMode` パラメータは必要ありません)。ステッチモードとは異なり、セッション初期化レスポンスには は含まれません `trackingUrl`。
2. キャッシュ可能なマニフェスト: MediaTailor は、MediaTailor インターステイシャルアセットリストエンドポイントを指す `CLASS="com.apple.hls.interstitial"` および `X-ASSET-LIST` 属性を持つ `EXT-X-DATERANGE` タグを含むキャッシュ可能なマニフェストを返します。
3. ビーコンメタデータを含むアセットリスト: プレイヤーが広告時間枠に遭遇すると、アセットリストが取得されます。MediaTailor は、各広告 URI に暗号化されたビーコンメタデータが含まれる JSON レスポンスを返します。

```
{
  "ASSETS": [
    {
      "DURATION": 30.0,
      "URI": "https://cdn.example.com/ad/master.m3u8?
awsBeaconData=<encrypted>&awsBeaconDomain=<MediaTailor-
endpoint>&awsConfigurationName=<config-name>"
    }
  ]
}
```

サーバー側のレポートがアクティブな場合、レスポンスには TRACKING セクションは含まれません。広告 URIs 格納されます。

4. HLS 変数置換: プレイヤーはアドマルチバリエーションプレイリストを取得します。広告マニフェストは、`#EXT-X-DEFINE:QUERYPARAM` デイレクティブを使用して、URI クエリ文字列から HLS 変数置換を介してセグメント URLs にビーコンパラメータを渡します。

```
#EXTM3U
#EXT-X-DEFINE:QUERYPARAM="awsBeaconData"
#EXT-X-DEFINE:QUERYPARAM="awsBeaconDomain"
#EXT-X-DEFINE:QUERYPARAM="awsConfigurationName"
#EXTINF:5.0,
${awsBeaconDomain}/segment/hash/${awsConfigurationName}/${awsBeaconData}/0/0?
aws.segmentRelativePath=asset_00001.ts
```

プレイヤーは `${awsBeaconData}`、広告マニフェスト URI クエリ文字列の値を使用して `${awsBeaconDomain}`、および `${awsConfigurationName}` 変数を解決し、MediaTailor を介して各広告セグメントをリクエストします。

5. セグメントリクエストでビーコンが発射する: プレイヤーが各広告セグメントをリクエストすると、リクエストは MediaTailor を経由します。サービスはビーコンデータを復号し、広告内のセグメントの位置 (インプレッション、第 1 四分位、中間、第 3 四分位、または完了) を決定し、適切な VAST 追跡ビーコンを広告サーバーに発射します。MediaTailor はプレイヤーを実際の広告コンテンツセグメントにリダイレクトします。

SGAI サーバー側のビーコンのプレイヤー要件

SGAI でサーバー側のビーコンを使用するには、プレイヤーが次の要件を満たしている必要があります。

- HLS バージョン 11 以降
- HLS Interstitials の CLASS 属性 `EXT-X-DATERANGE` を使用した のサポート
- `#EXT-X-DEFINE:QUERYPARAM` 可変置換 (RFC 8216bis) のサポート。プレイヤーは、クエリパラメータ値をセグメント URLs に置き換える前に、パーセントデコードする必要があります。

Note

SGAI サーバー側のビーコンは現在、HLS でのみサポートされています。DASH は、SGAI サーバー側のビーコンではまだサポートされていません。

ステッチモードのサーバー側追跡との比較

次の表は、ステッチ広告挿入とサーバーガイド広告挿入でサーバー側の追跡がどのように異なるかをまとめたものです。

側面	ステッチ (SSAI)	サーバーガイド (SGAI)
マニフェストのキャッシュ可能性	セッションごと、キャッシュ不可	キャッシュ可能、ビューワー間で共有
広告セグメントのルーティング	セッション ID /v1/segment/ を使用する	暗号化されたビーコンデータ BLOB /v1/segment/ を使用する
ビーコンのセッション状態	MediaTailor にセッションごとに保存	セッションレス — すべての状態は暗号化されたawsBeaconData パラメータで伝送されます
セッション開始時の追跡 URL	セッション初期化レスポンスで返されます	指定なし — ビーコンデータは各アセットリストレスポンスの広告URLsに埋め込まれます
DASH サポート	サポート	まだサポートされていません。

Note

ライブ SGAI セッションでは、を使用してマニフェストベースの広告プリフェッチを有効にできますaws.guidedPrefetchMode=MANIFEST。これは、ステッチド (SSAI) セッションで使用されるスケジュールベースのプリフェッチ API とは異なります。詳細については、「[マニフェストハートビートを使用したガイド付きプリフェッチ](#)」を参照してください。

サーバー側の追跡ビーコン用語集

MediaTailor サーバー側の追跡では、標準化されたビーコンのセットを使用して、広告表示の進行状況を広告サーバーと検証サービスに報告します。これらのビーコンは、動画広告測定インタラクティブ広告局 (IAB) 標準に準拠しており、広告インプレッションと完了率の正確なレポートを提供します。

サーバー側の追跡ビーコンタイプ

ビーコンタイプ	発砲時	目的	タイミングの詳細
インプレッション	プレイヤーが最初の広告セグメントをリクエストしたとき	広告コンテンツのロードが開始され、ビューワーに表示しようとしていることを示します。	広告の最初の/v1/segment リクエストで発砲されました。インプレッションをカウントする前に広告コンテンツのロードを開始することを義務付ける IAB ガイドラインに準拠しています。完全なシーケンス サーバー側の追跡ワークフロー については、「」を参照してください。
開始	プレイヤーが広告コンテンツのレンダリングを開始するとき	広告再生が実際に開始されたことを確認します	通常、最初のセグメントリクエストでインプレッションビーコンと同時に起動されますが、広告レンダリングの実際の開始を表します。この区別は、インプレッションイベントと開始イベントの両方を個別に追跡する検証サービスにとって重要です。
第 1 四分位数	プレイヤーが広告期間の 25% に達した場合	広告の第 1 四半期を通じて広告表示を継続した測定値	プレイヤーが広告期間の 25% ポイントを含むセグメントをリクエストしたときに発生します。たとえば、2 秒のセグメントを持つ 20 秒の広告では、通常、これは 3 番目のセ

ビーコンタイプ	発砲時	目的	タイミングの詳細
			グメントのリクエストで発生します (広告から約 4~6 秒後)。
中間点	プレイヤーが広告期間の 50% に達した場合	広告の半分を通じて継続的な広告表示を測定する	プレイヤーが広告期間の 50% ポイントを含むセグメントをリクエストしたときに発生します。たとえば、2 秒のセグメントを持つ 20 秒の広告では、通常、これは 5 番目のセグメント (広告から約 8~10 秒) のリクエストで発生します。
第 3 四分位	プレイヤーが広告期間の 75% に達した場合	広告の 4 分の 3 を通じて継続的な広告表示を測定する	プレイヤーが広告期間の 75% ポイントを含むセグメントをリクエストしたときに発生します。たとえば、2 秒のセグメントを持つ 20 秒の広告では、通常、これは 8 番目のセグメント (広告から約 14~16 秒) のリクエストで発生します。

ビーコンタイプ	発砲時	目的	タイミングの詳細
完了	プレイヤーが広告の最後に到達したとき	広告全体がビューワーに配信されたことを確認します	プレイヤーが広告の最終セグメントをリクエストしたときに発生します。これは、ビューワーが広告コンテンツ全体を見ている可能性があることを示します。たとえば、2 秒のセグメントを持つ 20 秒の広告では、通常、これは 10 番目のセグメントのリクエストで発生します (広告から約 18 ~ 20 秒後)。

Note

ビーコン射撃の正確なタイミングは、セグメントの期間と広告の長さによって異なります。MediaTailor は、特定の広告期間とセグメント構造に基づいて、各四分位位置に対応する適切なセグメントリクエストを計算します。

サーバー側の追跡タイミングとキャッシュ動作

サーバー側のレポートでは、MediaTailor は、マニフェスト解析やプリロードアクティビティではなく、プレイヤーからの実際のセグメントリクエストに基づいて追跡イベントを実行します。このアプローチにより、動画広告測定の業界標準に沿った正確なインプレッションカウントが保証されます。

主要なタイミングの原則

MediaTailor サーバー側の追跡は、以下の基本的なタイミング原則に従います。

- 実際のセグメントリクエストでイベントを追跡する - ビーコンは、マニフェストの解析やキャッシュ中ではなく、プレイヤーが `/v1/segment URLs` に HTTP リクエストを行った場合にのみ送信されます。
- プレイヤーのキャッシュとマニフェストの事前ロードはイベントをトリガーしません - プレイヤーは追跡イベントを生成せずにマニフェスト情報を解析、キャッシュ、または事前ロードできます。

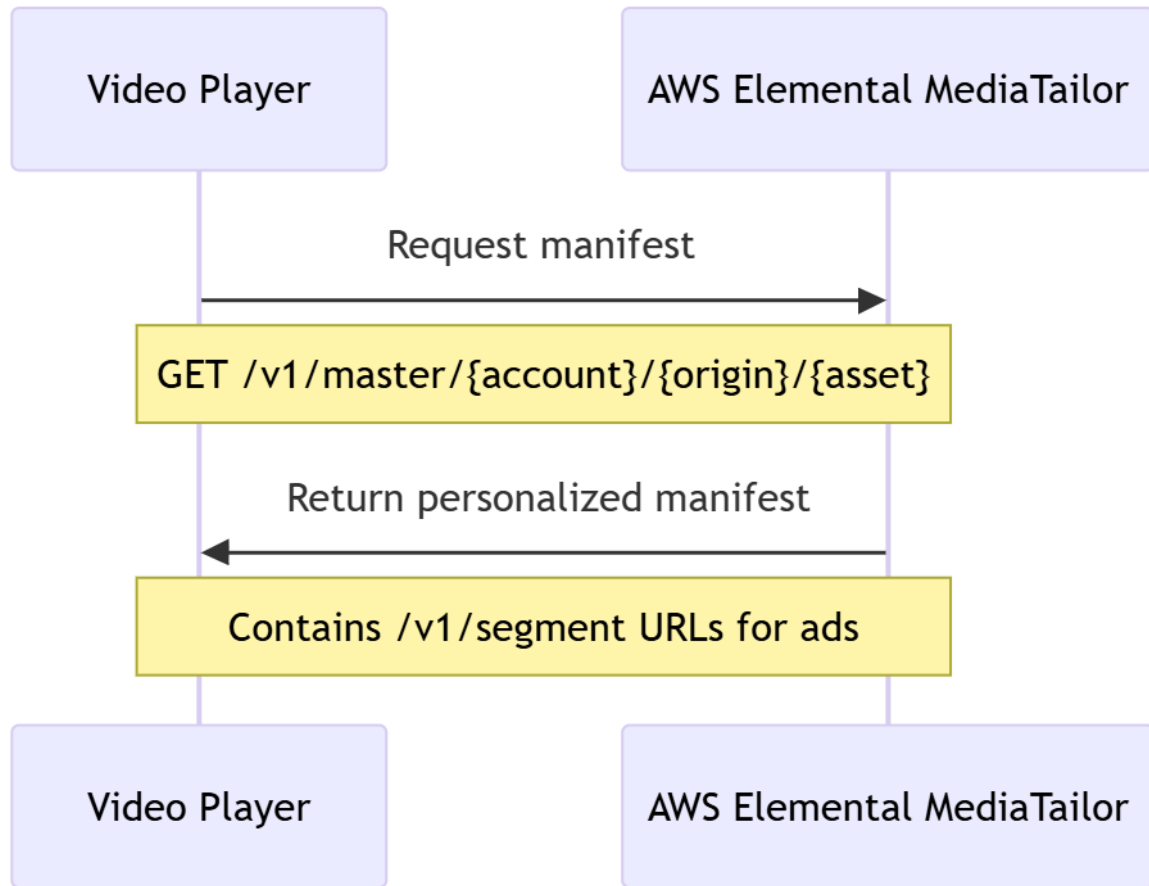
- セグメントプリフェッチはイベントをトリガーします - プレイヤーが再生前に実際の広告セグメントをプリフェッチする場合、これはセグメントリクエストが有効なインプレッションを構成する業界標準の動作に従います。
- 各 /v1/segment request triggers appropriate beacon - 特定の追跡イベント (インプレッション、四分位数、完了) は、リクエストされる広告の位置とセグメントによって決まります。
- タイミングは IAB 標準に準拠 - このアプローチは、動画広告の測定とインプレッションカウントに関するインタラクティブ広告局のガイドラインに従います。

サーバー側の追跡ワークフロー

次の図は、サーバー側の完全な追跡ワークフローを示しています。このワークフローは、プレイヤーリクエストに関連して追跡イベントが発生するタイミングを示しています。

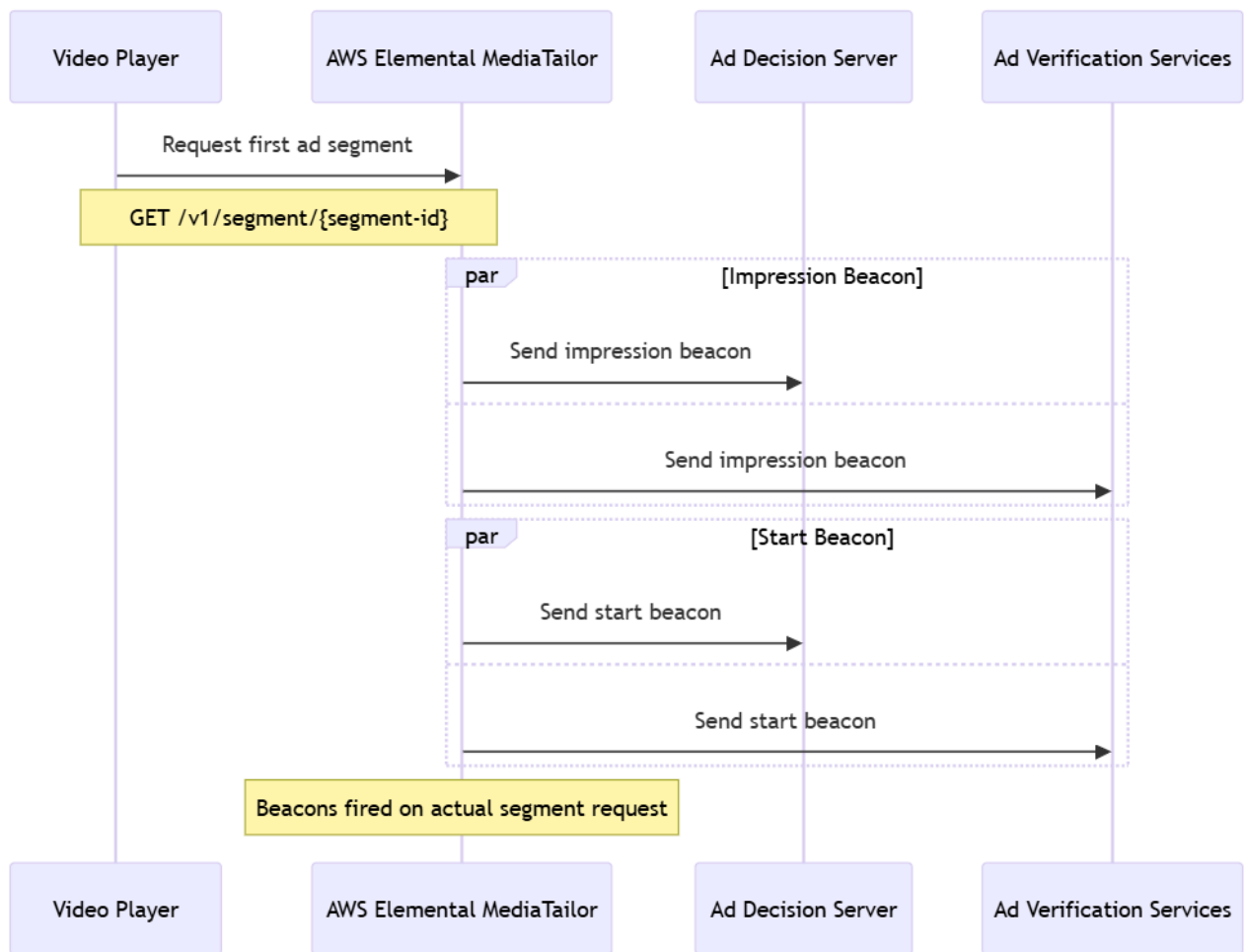
フェーズ 1: セッションの初期化

プレイヤーは MediaTailor にマニフェストをリクエストし、広告セグメント URLs を含むパーソナライズされたマニフェストを返します。



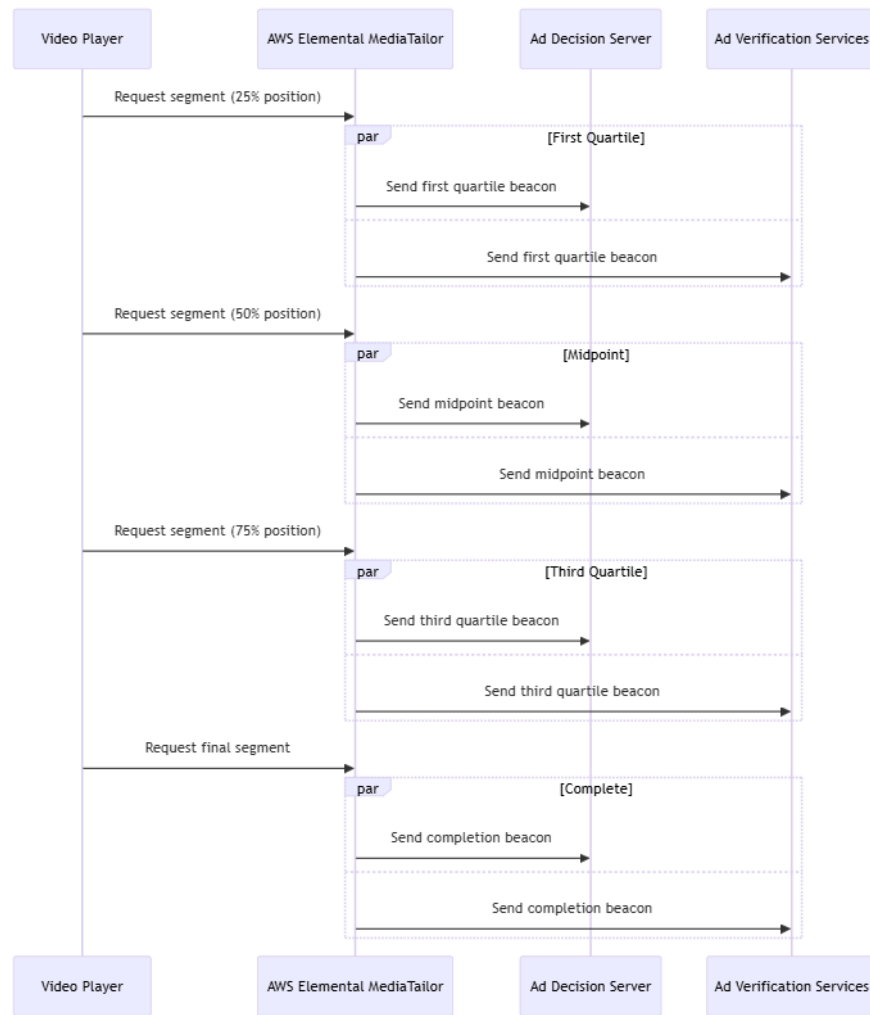
フェーズ 2: 広告リクエストとインプレッションの追跡

プレイヤーが最初の広告セグメントをリクエストすると、MediaTailor はインプレッションを起動し、広告決定サーバーと広告検証サービスの両方にビーコンを開始します。



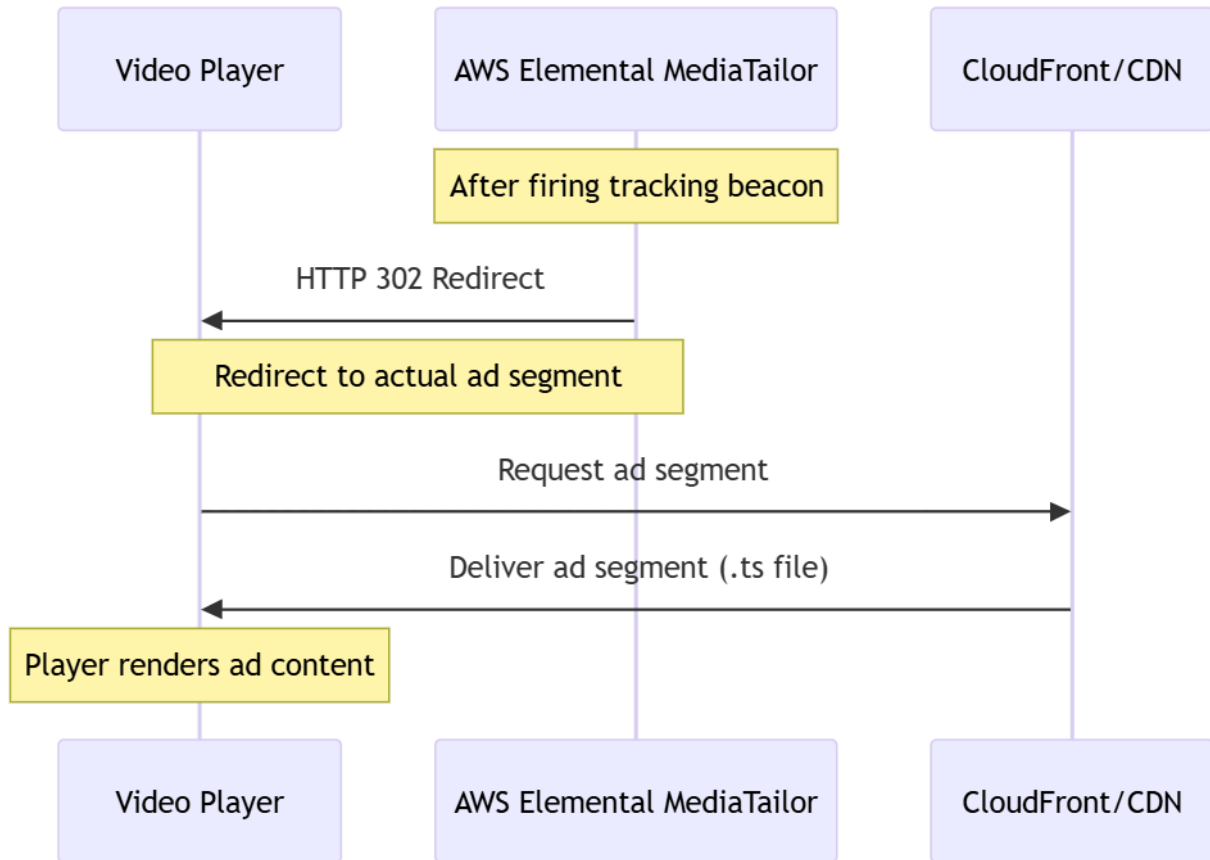
フェーズ 3: 四分位数の追跡

MediaTailor は、後続のセグメントリクエストに基づいて四分位ビーコン (第 1 四分位、中間点、第 3 四分位、完了) を起動します。



フェーズ 4: セグメント配信

追跡ビーコンを発射すると、MediaTailor は Amazon CloudFront または CDN から実際の広告セグメントにリダイレクトします。



サーバー側の追跡ワークフローには、次の主要なタイミング動作が含まれます。

1. セッションの初期化 - プレイヤーは MediaTailor にマニフェストをリクエストします。MediaTailor は、 /v1/segmentパスを持つ広告セグメント URLsを含むパーソナライズされたマニフェストを返します。
2. マニフェスト解析とキャッシュ - プレイヤーはマニフェストを解析し、セグメント情報をプリロードまたはキャッシュすることがあります。このフェーズでは、プレイヤーのキャッシュ動作に関係なく、追跡イベントは発生しません。
3. 広告セグメントのリクエストとインプレッションの追跡 - プレイヤーが実際に最初の広告セグメントをリクエストすると (通常は再生用)、MediaTailor はインプレッションビーコンを起動し、広告決定サーバーと広告検証サービスの両方にイベントの追跡を開始します。これは、マニフェストが解析されたときではなく、 /v1/segmentURL への実際の HTTP リクエストで発生します。

4. セグメントリクエストに基づく四分位追跡 - MediaTailor は、広告期間中の計算された四分位位置に対応する後続のセグメントリクエストに基づいて、広告決定サーバーと広告検証サービスの両方に四分位ビーコン (第 1 四分位、中間点、第 3 四分位、完了) を起動します。
5. セグメント配信 - 適切な追跡ビーコンを発射した後、MediaTailor は実際の広告セグメント (Amazon CloudFront または CDN) への HTTP リダイレクトを発行します。

プレイヤーのキャッシュとプリロードに関する考慮事項

MediaTailor サーバー側の追跡は、正確なインプレッション測定を維持しながら、さまざまなプレイヤーキャッシュおよびプリロード戦略と互換性があるように設計されています。

- マニフェストプリロード - マニフェスト情報をプリロードまたはキャッシュするプレイヤーは、追跡イベントをトリガーしません。追跡イベントは、実際のセグメントリクエストが行われた場合にのみ発生します。
- セグメントのプリフェッチ - プレイヤーが再生前に広告セグメントをプリフェッチすると、それらのセグメントがリクエストされたときに、実際の再生時間よりも前にイベントの追跡が開始されます。この動作は、セグメントリクエストを有効なインプレッションと見なす業界標準と一致しています。
- プレイヤーバッファリング - 標準プレイヤーバッファリング動作 (再生の少し前にセグメントをリクエスト) は、セグメントリクエストパターンに基づいて、適切なタイミングで追跡イベントをトリガーします。

追跡の不一致のトラブルシューティング

MediaTailor のサーバー側の追跡とサードパーティーのメトリクス間に不一致がある場合は、次の要素を考慮してください。

- プレイヤーの動作の違い - プレイヤーによってプリフェッチ戦略とバッファリング戦略が異なり、セグメントリクエストが行われたときに影響する場合があります。
- ネットワーク条件 - ネットワーク条件が悪いと、プレイヤーが複数回、または予想とは異なる間隔でセグメントをリクエストする可能性があります。
- CDN 設定 - `/v1/segment` リクエストの CDN キャッシュが正しくないと、追跡イベントが見逃されたり、重複したりする可能性があります。
- セッション管理 - イベント競合の追跡を防ぐために、各再生セッションで一意的なセッション識別子が使用されていることを確認します。

トラブルシューティングの詳細なガイダンスについては、「」を参照してください [一般的な問題のトラブルシューティング](#)。

MediaTailor サーバー側の追跡機能

AWS Elemental MediaTailor は、これらの統合されたサーバー側の追跡機能を自動的に適用して、広告測定精度と信頼性を最適化します。このシステムは、ビーコンの重複を防ぎ、大量の期間中のトラフィックを管理し、適切なイベントシーケンスを維持し、ユーザーによる設定を必要とせずに包括的なパフォーマンスモニタリングを提供します。広告決定サーバー (ADS) が VAST レスポンスで追跡ビーコンを提供することを確認するだけで済みます。

Note

これらの機能は、2025 年 9 月 30 日以降、新規のお客様にご利用いただけます。既存のお客様は、継続的なサービス改善の一環として、2025 年を通じて にアクセスできます。これらの機能にすぐにアクセスしたい場合は、[AWS サポート](#)にお問い合わせください。

Note

これらの機能は、ステイッチド (SSAI) 広告挿入方法とサーバーガイド (SGAI) 広告挿入方法の両方に適用されます。ビーコンのタイプとタイミングは、両方のモードで同じです。MediaTailor がビーコンをトリガーする方法は異なります。SGAI サーバー側のビーコンの詳細については、[サーバーガイド広告挿入によるサーバー側の追跡 \(SGAI\)](#)「」を参照してください。

ビーコン重複排除

MediaTailor は、同一の広告イベントに対するビーコンの重複射撃を防止します。サーバー側の追跡システムは、各インプレッション、四分位数、完了ビーコンを広告表示セッションごとに 1 回のみ送信します。ネットワーク条件、ビットレートの変更、バッファリング戦略により、ビデオプレイヤーが同じ広告セグメントを複数回リクエストすると、MediaTailor は発射されたビーコンを追跡し、冗長な送信をブロックします。

重複排除は、ビーコン数の増加を引き起こす一般的なシナリオを自動的に解決します。

- アダプティブビットレートストリーミング - プレイヤーが同じ広告セグメントの異なる品質バリエーションをダウンロードする場合

- ネットワーク再試行シナリオ - ネットワークの問題またはタイムアウトが原因でプレイヤーがセグメントを再リクエストする場合
- プレイヤーバッファリング戦略 - プレイヤーがバッファリング目的でセグメントをプリフェッチまたは再フェッチする場合

このシステムは、プレイヤーが異なる品質レベルを切り替える場合でも、インプレッションビーコンを 1 回だけ発射するように設計されています。

アダプティブスロットリングとビーコンの再試行

MediaTailor は、サーバーレスポンスインジケータに基づいてビーコントラフィックレートを自動的に管理します。システムは HTTP レスponsパターン、接続タイムアウト、エラーコードをモニタリングして輻輳を検出し、それに応じてトラフィックレートを調整します。システムがサーバーのストレスインジケータを識別すると、影響を受けるドメインのトラフィックレートが低下し、サーバーが容量の改善を示すと自動的にレートが増加します。

システムは、次のインジケータを使用してサーバーの状態をモニタリングします。

- HTTP 接続タイムアウト - 測定プラットフォームが想定期間内に応答しない場合
- エラーレスポンスコード - サーバーの過負荷を示す 503、504、507 レスpons。完全な互換性を確保するために、広告サーバーはこれらのエラーコードもサポートする必要があります。
- レスponsパターン - 容量の問題を示す測定プラットフォームのパフォーマンスの変化

再試行動作は、最大 1 時間、試行間の最小 30 秒の遅延で自動的に配信を試行します。この再試行動作は設定できません。

1 秒あたりのビーコントラフィックの管理

TPS 制限を設定して、ビーコン配信レートを制御できます。これは、サーバー側の追跡機能に対して設定可能な唯一の設定です。アカウントレベルの制限は、すべての測定パートナーに送信される広告追跡リクエストの合計数を制限します。MediaTailor は、エンタープライズ規模のオペレーションに十分な容量を提供するために、最小 TPS 制限 10,000 を適用します。

AWS サポートチケットを送信して、以下の情報を含む TPS 制限を確立します。

- AWS アカウント ID - 特定のアカウント識別子
- ターゲットリージョン - TPS 制限を適用する AWS リージョン

- 希望する TPS しきい値 - 1 秒あたりの必要なトランザクション数の制限 (最小 10,000)

デフォルトでは、TPS 制限はありません。広告決定サーバー (ADS) で必要な場合は TPS 制限をリクエストできますが、制限は 10,000 TPS を超える必要があります。MediaTailor は指定された制限を超えることはありませんが、その制限まで一貫したスループットを保証するものではありません。広告決定サーバーは、サポートできる TPS 制限を知らせます。

インオーダービーコン

MediaTailor は、広告追跡イベントのシーケンシャル配信を自動的に維持します。システムは、ネットワークの問題、再試行、トラフィック管理が発生した場合でもビーコンの順序を保持します。これにより、測定パートナーは正確な分析のために正しい順序でイベントを受信できます。

システムは、標準の業界ビーコンシーケンスに従います。

1. 開始イベント - 広告の再生が開始されると起動します
2. 第 1 四分位イベント - 広告完了率 25% で発生
3. 中間イベント - 広告完了率 50% で発生
4. 第 3 四分位イベント - 広告完了率 75% で発生
5. 完了イベント - 広告が終了すると起動します

これらの機能は、自動的に連携します。

- 適切な順序を維持するために、スロットリング中にビーコンが保持されます
- 各測定パートナードメインには、レート調整中の中断を防ぐために個別のイベントキューがあります。
- 重複排除は、時系列を維持しながらイベントタイプとタイムラインの位置を追跡します

クライアント側の広告追跡

AWS Elemental MediaTailor クライアント側の追跡 API を使用すると、広告時間枠中にプレーヤーコントロールをストリーミングワークフローに組み込むことができます。クライアント側の追跡では、プレーヤーまたはクライアントはインプレッションや四分位広告ビーコンなどの追跡イベントを広告決定サーバー (ADS) やその他の広告検証エンティティに発行します。これらのイベントは、全体的な広告時間枠のステータスと、各時間枠内の個々の広告表示の両方を追跡します。インプレッションと四分位数 (ADS) およびその他の広告検証エンティティの詳細については、「」を参照してください

い。インプレッションと四分位広告ビーコンの詳細については、「」を参照してください[クライアント側のビーコン](#)。ADS およびその他の広告検証エンティティの詳細については、「」を参照してください[クライアント側の広告追跡統合](#)。

クライアント側の追跡のためにプレイヤーパラメータとセッションデータを ADS に渡す方法については、[ADS リクエストの MediaTailor プレイヤー変数](#)「」および「」を参照してください[ADS リクエストの MediaTailor セッション変数](#)。

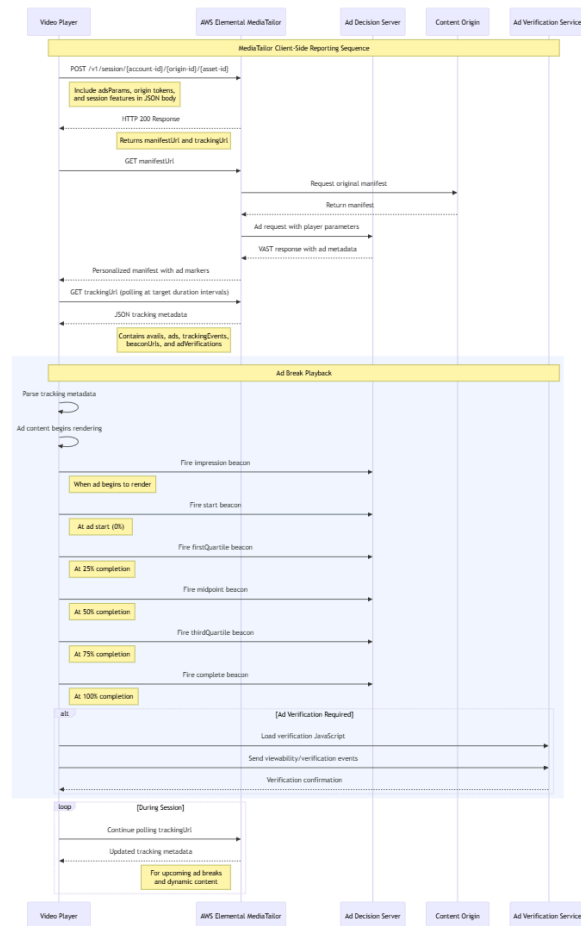
クライアント側の追跡では、次のような機能が有効になります。

- 広告ブレークカウントダウンタイマー - 詳細については、「」を参照してください[広告カウントダウンタイマー](#)。
- 広告クリックスルー - 詳細については、「」を参照してください[広告クリックスルー](#)。
- コンパニオン広告の表示 - 詳細については、「」を参照してください[コンパニオン広告](#)。
- スキップ可能な広告 - 詳細については、「」を参照してください[スキップ可能な広告](#)。
- プライバシーコンプライアンス用の VAST アイコンの表示 - 詳細については、「」を参照してください[Google Why This Ad \(WTA\) のアイコン](#)。
- 広告中のプレイヤースクラブの制御 - 詳細については、「」を参照してください[スクラブ](#)。

MediaTailor クライアント側の追跡 API を使用すると、クライアント側の追跡に加えて機能を有効にするメタデータを再生デバイスに送信できます。

クライアント側のレポートワークフロー

次の図は、セッションの初期化から広告の再生とビーコンまでのクライアント側のレポートワークフロー全体を示しています。



クライアント側のレポートワークフローには、次のステップが含まれます。

1. セッションの初期化 - ビデオプレーヤーは、オリジントークン `adsParams`、セッション機能などの JSON メタデータを使用して MediaTailor セッションエンドポイントに POST リクエストを送信します。MediaTailor はセッション `trackingUrl` に対して `manifestUrl` とで応答します。
2. マニフェストリクエストと広告決定 - プレイヤーは MediaTailor にパーソナライズされたマニフェストをリクエストします。MediaTailor は、オリジンから元のコンテンツマニフェストをリクエストし、プレイヤーパラメータを使用して広告決定サーバー (ADS) に広告リクエストを行い、広告メタデータを含む VAST レスポンスを受信し、広告マーカを含むパーソナライズされたマニフェストをプレイヤーに配信します。
3. 追跡データの取得 - プレイヤーは追跡 URL を定期的にポーリングします (HLS のターゲット期間または DASH の最小更新期間に一致)。MediaTailor は、表示、広告、追跡イベント、ビーコン URLs、広告検証データを含む JSON 追跡メタデータを返します。
4. 広告の再生とビーコン - 広告ブレイク中に、プレイヤーは追跡メタデータを解析し、広告のレンダリング開始時にインプレッションビーコンを発射し、四分位ビーコン (開始、firstQuartile、中

間、thirdQuartile、完了) を適切なタイミングで発射し、必要に応じて広告検証 JavaScript をロードして実行し、表示可能性/検証イベントをサードパーティー検証サービスに送信します。

5. 継続的なポーリング - プレイヤーはセッション全体で追跡 URL のポーリングを継続し、今後の広告ブレイクと動的コンテンツの更新されたメタデータを受け取ります。

このワークフローにより、広告カウントダウンタイマー、クリックスルー機能、コンパニオン広告、スキップ可能な広告、プライバシーコンプライアンスのための VAST アイコン表示などの高度な機能が可能になります。

トピック

- [クライアント側の追跡の有効化](#)
- [広告サーバーパラメータ](#)
- [オリジンインタラクションクエリパラメータ](#)
- [セッション設定機能](#)
- [クライアント側追跡のベストプラクティス](#)
- [クライアント側の広告追跡スキーマとプロパティ](#)
- [広告追跡アクティビティのタイミング](#)
- [クライアント側の広告追跡のプレイヤーコントロールと機能](#)
- [クライアント側のビーコン](#)
- [サーバー側の広告ビーコンを使用したハイブリッドモード](#)
- [クライアント側の広告追跡統合](#)
- [GetTracking を使用した広告ビーコンのページング](#)

クライアント側の追跡の有効化

セッションごとにクライアント側の追跡を有効にします。プレイヤーは MediaTailor 設定のセッション初期化プレフィックスエンドポイントPOSTに HTTP を作成します。オプションで、プレイヤーは MediaTailor が広告通話を行ったり、マニフェストのオリジンを呼び出したり、セッションレベルで MediaTailor 機能呼び出し無効にしたりするときに使用する追加のメタデータを送信できます。

次の例は、JSON メタデータの構造を示しています。

```
{
```

```
"adsParams": {                                # 'adsParams' is case sensitive
  "param1": "value1",                        # key is not case sensitive
  "param2": "value2",                        # Values can contain spaces. For example, 'value 2'
  is an allowed value.
},
"origin_access_token":"abc123", # this is an example of a query parameter designated
for the origin
"overlayAvails":"on"                    # 'overlayAvails' is case sensitive. This is an
example of a feature that is enabled at the session level.
}
```

MediaTailor コンソールまたは API を使用して、これらのパラメータを参照するように ADS リクエストテンプレート URL を設定します。次の例では、`player_params.param1`は のプレイヤーパラメータで`param1`、`player_params.param2`は のプレイヤーパラメータです`param2`。

```
https://my.ads.com/path?param1=[player_params.param1]&param2=[player_params.param2]
```

広告サーバーパラメータ

JSON 構造の最上位レベルには `adsParamsJSON` オブジェクトがあります。このオブジェクト内には、MediaTailor がすべてのセッションリクエストで広告サーバーを読み取って送信できるキーと値のペアがあります。MediaTailor は、次の広告サーバーをサポートしています。

- Google 広告マネージャー
- SpringServe
- FreeWheel
- Publica

オリジンインタラクションクエリパラメータ

`adsParams`、`origin_access_token` など、JSON 構造の最上位レベル内の予約されたキーと値のペアは`overlayAvails`、クエリパラメータの形式でオリジンリクエスト URL `availSuppression`に追加されません。MediaTailor がオリジンに対して行うすべてのセッションマニフェストリクエストには、これらのクエリパラメータが含まれます。オリジンは無関係なクエリパラメータを無視します。例えば、MediaTailor はキーと値のペアを使用して、オリジンにアクセストークンを送信できます。

セッション設定機能

session-initialization JSON 構造を使用して、overlayAvails、などの MediaTailor 機能を有効、無効にavailSuppression、または上書きしますadSignaling。セッションの初期化中に渡された機能設定は、MediaTailor 設定レベルでの設定よりも優先されます。

Note

セッションの初期化時に MediaTailor に送信されるメタデータはイミュータブルであり、セッション中にメタデータを追加することはできません。SCTE-35 マーカーを使用して、セッション中に変化するデータを転送します。詳細については、「[ADS リクエストの MediaTailor セッション変数](#)」を参照してください。

Example: HLS のクライアント側の広告追跡の実行

```
POST mediatailorURL/v1/session/hashed-account-id/origin-id/asset-id.m3u8

{
  "adsParams": {
    "deviceType": "ipad"    # This value does not change during the session.
    "uid": "abdgfdyei-2283004-ueu"
  }
}
```

Example: DASH のクライアント側の広告追跡を実行する

```
POST mediatailorURL/v1/session/hashed-account-id/origin-id/asset-id.mpd

{
  "adsParams": {
    "deviceType": "androidmobile",
    "uid": "xjhddli-9189901-uic"
  }
}
```

レポートモードパラメータ

リクエスト本文に reportingMode パラメータを含めることで、セッションを初期化するときにレポートモードを指定できます。このパラメータは、MediaTailor がセッションのクライアント側またはサーバー側の広告追跡を実行するかどうかを制御します。

- **client** - プレイヤーは広告追跡を実行し、ビーコンを広告サーバーに送信します。これは、 が指定されていない場合のデフォルトモード `reportingMode` です。
- **server** - MediaTailor はサーバー側の広告追跡を実行し、ビーコンを広告サーバーに直接送信します。

Exampleサーバー側のレポートモードによるセッションの初期化

POST *mediatailorURL*/v1/session/*hashed-account-id/origin-id/asset-id*.m3u8

```
{
  "adsParams": {
    "deviceType": "ipad",
    "uid": "abdgfdyei-2283004-ueu"
  },
  "reportingMode": "server"
}
```

Exampleクライアント側のレポートモードによるセッションの初期化 (明示的)

POST *mediatailorURL*/v1/session/*hashed-account-id/origin-id/asset-id*.mpd

```
{
  "adsParams": {
    "deviceType": "androidmobile",
    "uid": "xjhddli-9189901-uic"
  },
  "reportingMode": "client"
}
```

Note

`reportingMode` パラメータはセッションの初期化時に設定され、セッション中に変更することはできません。 `reportingMode` を指定しない場合、MediaTailor は下位互換性を維持するためにデフォルトでクライアント側のレポートになります。

正常なレスポンスは、レスポンス本文200を持つ HTTP です。本文には、 `manifestUrl` と `trackingUrl` キーを含む JSON オブジェクトが含まれています。値は、プレイヤーが再生と広告イベントの追跡の両方の目的で利用できる相対 URLs です。


```
{
  "manifestUrl": "/v1/dashmaster/hashed-account-id/origin-id/asset-id.m3u8?
aws.sessionId=session-id",
  "trackingUrl": "/v1/tracking/hashed-account-id/origin-id/session-id"
}
```

クライアント側の追跡スキーマの詳細については、「」を参照してください[クライアント側の広告追跡スキーマとプロパティ](#)。

クライアント側追跡のベストプラクティス

このセクションでは、ライブワークフローと VOD ワークフローの両方で MediaTailor でクライアント側を追跡するためのベストプラクティスの概要を説明します。

ライブワークフロー

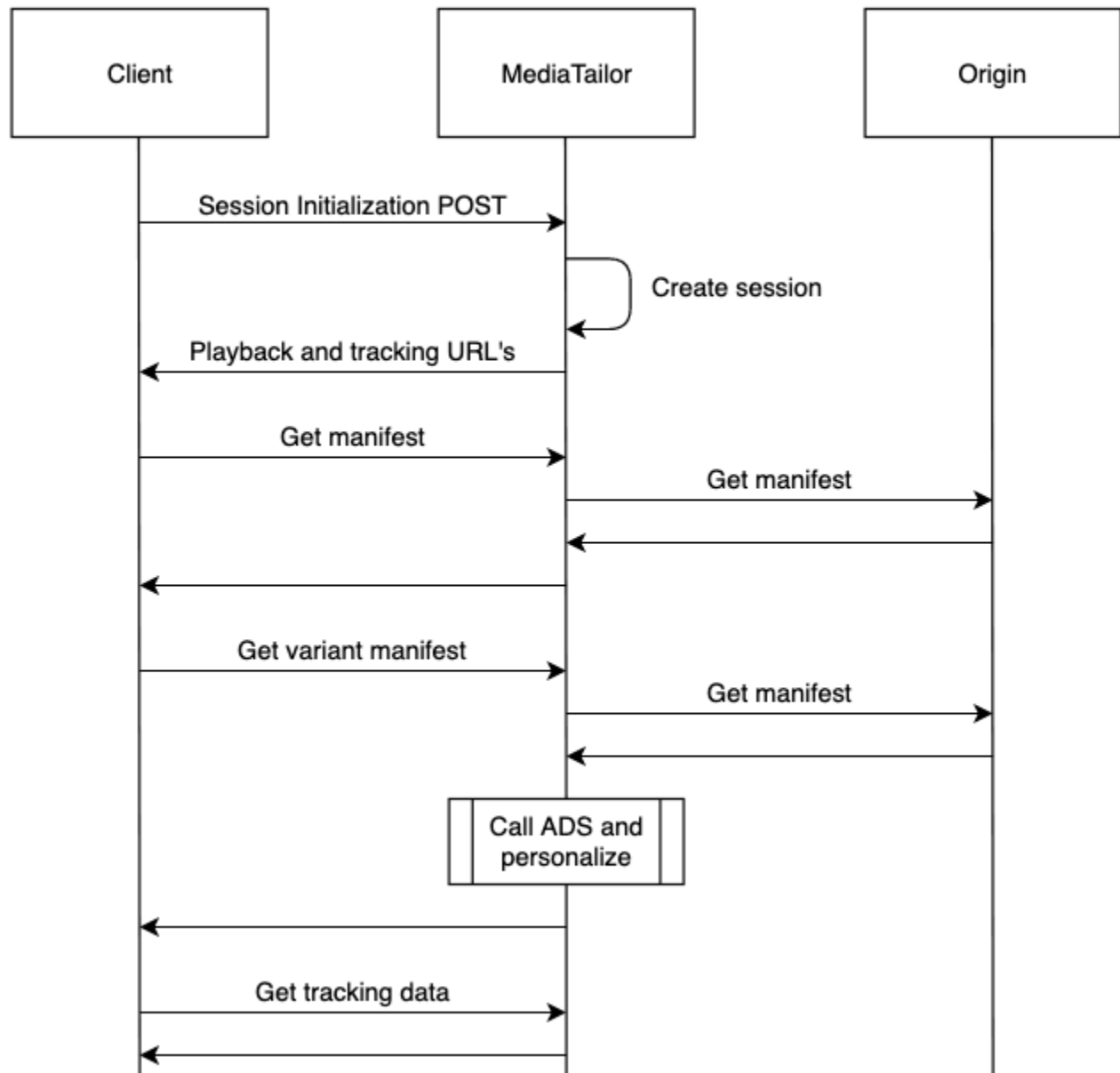
最新の広告追跡メタデータを常に取り得るために、HLS の各ターゲット期間、または DASH の最小更新期間に一致する間隔で追跡エンドポイントをポーリングします。この間隔を一致させることは、クリエイティブにインタラクティブコンポーネントまたはオーバーレイコンポーネントがあるワークフローで特に重要です。

Note

一部のプレイヤーは、ポーリングの代わりに使用できるイベントリスナーをサポートしています。例えば、MediaTailor 広告 ID デコレーション機能はセッションごとに有効にする必要があります。詳細については、「[広告 ID デコレーション](#)」を参照してください。この機能を使用すると、表示の各広告に日付範囲 (HLS) またはイベント要素 (DASH) 識別子が配置されます。プレイヤーは、これらのマニフェストタグを、セッションの MediaTailor 追跡エンドポイントを呼び出すプロンプトとして使用できます。

VOD ワークフロー

セッションの初期化が成功し、MediaTailor がメディアを含む最初のマニフェストを受け取ったら、追跡エンドポイントを一度呼び出すだけで済みます。



サーバーガイド広告挿入

サーバーガイド広告挿入 (SGAI) セッションは GetTracking API を使用しません。代わりに、`aws.reportingMode=CLIENT` を使用すると、MediaTailor は、プレイヤーが広告コンテンツをリクエストしたときに、各アセットリストレスポンスの TRACKING セクションに追跡情報を提供します。セッション初期化レスポンスには `trackingUrl` は含まれません。

クライアント側で追跡される SGAI セッションのアセットリストレスポンスの構造は次のとおりです。

```
{
  "ASSETS": [
    { "DURATION": 20.0, "URI": "https://cdn.example.com/ad1/master.m3u8" },
    { "DURATION": 10.0, "URI": "https://cdn.example.com/ad2/master.m3u8" }
  ],
  "TRACKING": {
    ...VAST tracking events and beacon URLs for each ad...
  }
}
```

SGAI メソッドのクライアント側追跡を実装する場合:

- を呼び出すのではなく、アセットリストレスポンスから TRACKING セクションを解析する `GetTracking`
- 広告イベントレポート用にアセットリストに記載されている追跡 URLs を使用する
- プレイヤーの実際の広告再生イベントに基づく射撃追跡ビーコン
- アセットリストが取得されるたびに、各広告時間枠の追跡を個別に処理する

Important

TRACKING セクションは、が設定されている場合にのみアセットリストに含まれ `aws.reportingMode=CLIENT` ます。サーバー側のレポートを使用する場合 (SGAI のデフォルト)、MediaTailor は TRACKING セクションを省略し、代わりに広告 URLs。詳細については、「[サーバーガイド広告挿入によるサーバー側の追跡 \(SGAI\)](#)」を参照してください。

クライアント側の広告追跡スキーマとプロパティ

MediaTailor クライアント側の広告追跡機能を使用すると、詳細なクライアント側の広告追跡データをプレイヤー環境に統合できます。以下のセクションでは、広告追跡スキーマ全体と、スキーマを構成する特定のプロパティと値について説明します。

Schema

次の JSON 構造は、MediaTailor クライアント側の広告追跡スキーマを示しています。この表現は、異なるパート間の関係を理解するのに役立つスキーマのネスト構造を示しています。

各プロパティの詳細については、「」を参照してください [the section called “プロパティ”](#)。

```
{
  "avails": [
    {
      "ads": [
        {
          "adID": "string",
          "adParameters": "string",
          "adSystem": "string",
          "adTitle": "string",
          "adVerifications": [
            {
              "executableResource": [
                {
                  "apiFramework": "string",
                  "type": "string",
                  "uri": "string",
                  "language": "string"
                }
              ],
              "javaScriptResource": [
                {
                  "apiFramework": "string",
                  "browserOptional": "string",
                  "uri": "string"
                }
              ],
              "trackingEvents": [
                {
                  "event": "string",
                  "uri": "string"
                }
              ],
              "vendor": "string",
              "verificationParameters": "string"
            }
          ],
          "companionAds": [
            {
              "adParameters": "string",
              "altText": "string",
              "attributes": {
                "adSlotId": "string",
                "apiFramework": "string",
```

```
    "assetHeight": "string",
    "assetWidth": "string",
    "expandedHeight": "string",
    "expandedWidth": "string",
    "height": "string",
    "id": "string",
    "pxratio": "string",
    "renderingMode": "string",
    "width": "string"
  },
  "companionClickThrough": "string",
  "companionClickTracking": "string",
  "htmlResource": "string",
  "iFrameResource": "string",
  "sequence": "string",
  "staticResource": "string",
  "trackingEvents": [
    {
      "event": "string",
      "uri": "string"
    }
  ]
},
],
"creativeId": "string",
"creativeSequence": "string",
"duration": "string",
"durationInSeconds": number,
"extensions": [
  {
    "type": "string",
    "content": "string"
  }
],
"icons": [
  {
    "attributes": {
      "apiFramework": "string",
      "duration": "string",
      "height": "string",
      "offset": "string",
      "program": "string",
      "pxratio": "string",
      "width": "string",
```

```
    "xPosition": "string",
    "yPosition": "string"
  },
  "htmlResource": "string",
  "iconClicks": {
    "iconClickThrough": "string",
    "iconClickTracking": {
      "id": "string"
    },
  },
  "iconClickFallbackImages": [
    {
      "altText": "string",
      "height": "string",
      "width": "string",
      "staticResource": {
        "creativeType": "string",
        "uri": "string"
      }
    }
  ],
  "iconViewTracking": "string",
  "iFrameResource": "string",
  "staticResource": {
    "creativeType": "string",
    "uri": "string"
  }
},
"mediaFiles": {
  "adParameters": "string",
  "duration": "string",
  "durationInSeconds": number,
  "mediaFilesList": [
    {
      "apiFramework": "string",
      "delivery": "string",
      "height": "string",
      "maintainAspectRatio": "string",
      "mediaFileUri": "string",
      "mediaType": "string",
      "scalable": "string",
      "width": "string",
      "bitrate": "string"
    }
  ]
}
```

```

    }
  ],
  "mezzanine": "string",
  "startTime": "string",
  "startTimeInSeconds": number,
  "trackingEvents": [
    {
      "beaconUrls": ["string"],
      "duration": "string",
      "durationInSeconds": number,
      "dateTime": "string",
      "eventId": "string",
      "eventType": "string",
      "startTime": "string",
      "startTimeInSeconds": number
    }
  ]
},
"startTime": "string",
"startTimeInSeconds": number,
"dateTime": "string",
"adBreakTrackingEvents": [...],
"vastAdId": "string"
}
],
"adType": "string",
"availID": "string",
"duration": "string",
"durationInSeconds": number,
"startTime": "string",
"startTimeInSeconds": number,
"dateTime": "string",
"adMarkerDuration": "string",
"adProgramDateTime": "string",
"dashAvailabilityStartTime": "string",
"hlsAnchorMediaSequenceNumber": "string"
}
],
"nonLinearAvails": [
  {
    "nonLinearAds": [...],
    "nonLinearAdsList": [...]
  }
]
],

```

```
"nextToken": "string",
"meta": {}
}
```

プロパティ

次の表に、クライアント側の追跡 API のプロパティ、その定義、値タイプ、および例を示します。

プロパティ	定義	値のタイプ	例
adID	- HLS - 広告の先頭に関連付けられたシーケンス番号 - DASH - 広告の期間 ID パス: /avails/ads/adId VAST マッピング: なし	String	10
adBreakTrackingEvents	VAST レスポンスから VMAP 追跡イベントを伝送する配列。詳細については、VMAP 1.0 仕様のセクション 2.3.3 を参照してください。 パス: /avails/ads/adBreakTrackingEvents	配列	[]
adMarkerDuration	マニフェストの広告マーカーから観測された表示期間。	String	30
adParameters	MediaTailor がプレイヤーに渡す VAST VPAID からの広告パラメータの文字列。 パス: /avails/ads/adParameters VAST マッピング: VAST/Ad/Inline/Creatives/Cr	String	

プロパティ	定義	値のタイプ	例
	ective/Linear/AdParameters		
adProgram DateTime	• HLS - 広告の最初のメディアシーケンスを表す ISO/IEC 8601:2004 形式の日付。 • DASH - [指定なし]	String	
ads	表示を構成する広告オブジェクトを含む配列。広告はマニフェストに表示される順序で一覧表示されます。 パス: /avails/ads	配列	[]
adSystem	広告を提供するシステムの名前。 ⚠ Important 必ず値を指定してください。 値を指定しない場合、問題が発生する可能性があります。	String	myADS
adTitle	広告のタイトル。	String	ad1

プロパティ	定義	値のタイプ	例
adVerifications	クリエイティブ再生を検証するためにサードパーティーの測定コードを実行するために必要なリソースとメタデータが含まれています。このプロパティの詳細については、VAST 4.2 仕様のセクション 3.16 を参照してください。 MediaTailor は VAST 3 拡張ノード adVerifications として をサポートします。 パス: /avails/ads/adVerifications VAST マッピング: VAST/Ad/InLine/AdVerifications	配列	[]
altText	コンパニオン広告のイメージの代替テキスト。このテキストにより、視覚障害者がイメージの説明を読み返すための記述的オーディオサポートを持つプレイヤーが可能になります。 パス: /avails/ads/companionAds/altText	String	video sequence advertising sneakers
apiFramework	この広告が VPAID 広告であることをプレイヤーに伝える VPAID には、 に設定します。 スキーマ内の複数の場所に表示できます。	String	VPAID

プロパティ	定義	値のタイプ	例
availID	- HLS - 広告表示の開始に関連付けられたシーケンス番号。 - DASH - 広告表示の期間 ID。通常、広告に置き換えられるコンテンツの期間 ID です。 パス: /avails/availID	String	34 - PT34S_1
avails	アクティブなマニフェストウィンドウに表示される広告ブレイクオブジェクトまたは表示を含む配列。表示は、マニフェストに表示される順序で一覧表示されます。 パス: /avails	配列	[]
adType	広告のタイプ。 パス: /avails/adType および /avails/ads/adType	String	
dateTime	広告表示または広告の開始を ISO 8601 秒形式でプログラムする日時。 パス: /avails/dateTime および /avails/ads/dateTime	String	
duration	長さ、ISO 8601 秒形式。レスポンスには、広告表示全体と各広告とビーコンの期間が含まれますが、ビーコンの期間は常にゼロです。 パス: /avails/duration および /avails/ads/duration	String	15.015

プロパティ	定義	値のタイプ	例
durationInSeconds	長さ、秒形式。 パス: /avails/durationInSeconds および /avails/ads/durationInSeconds	Number	
extensions	広告サーバーが使用する VAST のカスタム拡張機能。拡張機能の詳細については、 VAST 4.2 仕様 のセクション 3.18 を参照してください。 パス: /avails/ads/extensions VAST マッピング: VAST/Ad/Inline/Extensions	配列	[]
icons	広告のアイコン要素。 パス: /avails/ads/icons VAST マッピング: VAST/Ad/Inline/Creatives/Creative/Linear/Icons	配列	
mediaFiles	プレイヤーが広告表示に必要とする動画やその他のアセット。 パス: /avails/ads/mediaFiles	オブジェクト	
nonLinearAvails	非線形広告表示オブジェクトの配列。 パス: /nonLinearAvails	配列	

プロパティ	定義	値のタイプ	例
<code>executableResource</code>	検証用の実行可能リソース。 パス: <code>/avails/ads/adVerifications/executableResource</code> VAST マッピング: <code>VAST/Ad/InLine/AdVerifications/Verification/ExecutableResource</code>	配列	
<code>javascriptResource</code>	検証用の JavaScript リソース。 パス: <code>/avails/ads/adVerifications/javascriptResource</code> VAST マッピング: <code>VAST/Ad/InLine/AdVerifications/Verification/JavaScriptResource</code>	配列	
<code>trackingEvents</code>	検証広告またはコンパニオン広告のイベントを追跡します。 パス: <code>/avails/ads/adVerifications/trackingEvents</code> または <code>/avails/ads/companionAds/trackingEvents</code>	配列	

プロパティ	定義	値のタイプ	例
vendor	検証ベンダー。 パス: /avails/ads/adVerifications/vendor VAST マッピング: VAST/Ad/InLine/AdVerifications/Verification/@vendor	String	
uri	実行可能アセット、動画アセット、または追跡エンドポイントを指す URI。 パス: スキーマ内のさまざまな場所 VAST マッピング: VAST のさまざまな CDATA 要素	String	https://tracking.example.com/impression
verificationParameters	検証パラメータ。 パス: /avails/ads/adVerifications/verificationParameters VAST マッピング: VAST/Ad/InLine/AdVerifications/Verification/VerificationParameters	String	
attributes	ディメンションやレンダリングモードなどのコンパニオン広告属性。 パス: /avails/ads/companionAds/attributes	オブジェクト	

プロパティ	定義	値のタイプ	例
companionClickThrough	視聴者がコンパニオン広告をクリックしたときにメディアプレーヤーが開く、広告主のページへの URL。 パス: /avails/ads/companionAds/companionClickThrough VAST マッピング: VAST/Ad/Inline/Creatives/Creative/CompanionAds/Companion/CompanionClickThrough	String	<code>https://aws.amazon.com/</code>
companionClickTracking	companionClickThrough プロパティの追跡 URL。 パス: /avails/ads/companionAds/companionClickTracking VAST マッピング: VAST/Ad/Inline/Creatives/Creative/CompanionAds/Companion/CompanionClickTracking	String	<code>https://myads.com/beaconing/event=clicktracking</code>

プロパティ	定義	値のタイプ	例
htmlResource	ストリーミングプロバイダーの HTML ページに直接挿入される CDATA エンコードされた HTML。 パス: /avails/ads/companionAds/htmlResource VAST マッピング: VAST/Ad/Inline/Creatives/Creative/CompanionAds/Companion/HTMLResource	String	<pre><![CDATA[<!doctype html><html><head>< meta name= \"viewport \" content= \"width=1, initial-s cale=1.0, minimum-s cale=1.0, ...]]></pre>
iFrameResource	ストリーミングプロバイダーが iframe にロードする HTML リソース ファイルの URL。 パス: /avails/ads/companionAds/iFrameResource VAST マッピング: VAST/Ad/Inline/Creatives/Creative/CompanionAds/Companion/iFrameResource	String	
sequence	VAST レスポンスでクリエイティブに指定されたシーケンス値。 パス: /avails/ads/companionAds/sequence	String	1

プロパティ	定義	値のタイプ	例
startTime	ISO 8601 秒形式の時間位置。HLS の場合、これは再生セッションの開始を基準としています。DASH の場合、これはマニフェストの AST (可用性の開始時刻) に関連しています。レスポンスには、ad avail 全体、および各広告とビーコンの開始時間が含まれます。 パス: /avails/startTime および /avails/ads/startTime	String	PT18.581355S
startTimeInSeconds	秒形式の時間位置。HLS の場合、これは再生セッションの開始を基準としています。DASH の場合、これはマニフェストの AST (可用性の開始時刻) に関連しています。レスポンスには、ad avail 全体、および各広告とビーコンの開始時間が含まれます。 パス: /avails/startTimeInSeconds および /avails/ads/startTimeInSeconds	Number	18.581
eventId	- HLS - ビーコンに関連付けられたシーケンス番号。 - DASH - 広告ptsTimeの開始の。	String	

プロパティ	定義	値のタイプ	例
event	追跡イベントの名前。 パス: /avails/ads/adVerifications/trackingEvents/event または /avails/ads/companionAds/trackingEvents/event	String	impression , start, firstQuartile , midpoint, thirdQuartile , complete
beaconUrls	MediaTailor が広告ビーコンを送信する URL。 パス: /avails/ads/mediaFiles/trackingEvents/beaconUrls	配列	
bitrate	ビデオアセットのビットレート。このプロパティは通常、実行可能アセットには含まれません。	String	2048
companionAds	1 つ以上のコンパニオン広告コンテンツ仕様。それぞれが使用するリソースファイルを指定します。コンパニオン広告は広告表示に付随し、広告のフレームやバナーなどのコンテンツを動画の近くに表示します。 パス: /avails/ads/companionAds	配列	[]
creativeId	広告のCreativeタグのId属性値。	String	creative-1
creativeSequence	VAST レスポンスAd@idの値に従って、広告が再生されるシーケンス。	String	1

プロパティ	定義	値のタイプ	例
dashAvailabilityStartTime	ライブ/動的 DASH の場合、オリジンマニフェストMPD@dashAvailabilityStartTime の。	String	2022-10-05T19:38:39.263Z
delivery	progressive または streaming プロトコルが使用されているかどうかを示します。	String	progressive
eventType	ビーコンのタイプ。 パス: /avails/ads/mediaFiles/trackingEvents/eventType	String	impression
height	ビデオアセットの高さをピクセル単位で示します。	String	360
hlsAnchorMediaSequenceNumber	HLS オリジンマニフェストに表示される最初/最も古いメディアシーケンスのメディアシーケンス番号。	String	77
maintainAspectRatio	スケーリング中に動画のアスペクト比を維持するかどうかを示します。	ブール値	true
mediaFilesList	プレイヤーが広告表示に必要とする動画やその他のアセットを指定します。 パス: /avails/ads/mediaFiles/mediaFilesList	配列	[]
mediaFileUri	実行可能アセットまたはビデオアセットを指す URI。	String	https://myad.com/ad/ad134/vpaid.js

プロパティ	定義	値のタイプ	例
mediaType	クリエイティブアセットまたはコンパニオンアセットの MIME タイプ。	String	video/mp4
meta	広告の追加メタデータ。	オブジェクト	
mezzanine	VPAID 広告にメザニン MP4 アセットが含まれている場合に指定される URL。 パス: /avails/ads/mediaFiles/mezzanine	String	https://gcdn.2mdn.net/video/playback/id/itag/c/k2/file/file.mp4
nextToken	このような値が存在する場合、結果の次のページを指すトークンの値。	String	UFQz0S44NzNTXzIwMjMtMDctMzFUMTY6NTA6MDYuMzUwNjI2ODQ1Wl8x
nonLinearAds	動画コンテンツと一緒に表示される非線形広告。	配列	[]
nonLinearAdsList	非線形広告のリスト。	配列	[]
scalable	動画を他のディメンションにスケールするかどうかを示します。	ブール値	true
skipOffset	ユーザーがスキップコントロールを使用できるようにするタイミングを識別する時間値。	String	00:00:05

プロパティ	定義	値のタイプ	例
staticResource	広告コンポーネントに使用される静的クリエイティブファイルへのURL。 パス: /avails/ads/companionAds/staticResource	String	https://very-interactive-ads.com/campaign1/file.json?c=1019113602
vastAdId	Ad タグのId属性値。	String	ad1
width	ビデオアセットの幅、ピクセル単位。	String	640
xPosition	ビデオプレーヤー内のアイコンの水平位置。特定のピクセル値、または「左」や「右」などの位置を指定できます。 パス: /avails/ads/icons/attributes/xPosition	String	left、または 10
yPosition	ビデオプレーヤー内のアイコンの垂直位置。特定のピクセル値、または「上」や「下」などの位置を指定できます。 パス: /avails/ads/icons/attributes/yPosition	String	top、または 10
iconClicks	アイコンのクリックスルーと追跡情報が含まれます。 パス: /avails/ads/icons/iconClicks	オブジェクト	

プロパティ	定義	値のタイプ	例
iconClickThrough	ビューワーがアイコンをクリックしたときにメディアプレーヤーが開く、広告主のページへの URL。 パス: /avails/ads/icons/iconClicks/iconClickThrough	String	https://advertiser.com/landing-page
iconClickTracking	iconClickThrough プロパティの追跡 URL。 パス: /avails/ads/icons/iconClicks/iconClickTracking	オブジェクト	
iconClickFallbackImages	アイコンを表示できない場合に表示するフォールバックイメージの配列。 パス: /avails/ads/icons/iconClicks/iconClickFallbackImages	配列	
iconViewTracking	アイコンが表示されたときに追跡するための URL。 パス: /avails/ads/icons/iconViewTracking	String	https://tracking.example.com/icon-view
offset	広告の再生中にアイコンが表示されときのタイムオフセット。 パス: /avails/ads/icons/attributes/offset	String	00:00:05

プロパティ	定義	値のタイプ	例
program	アイコンに関連付けられた「AdChoices」などのプログラムまたはイニシアチブ。 パス: /avails/ads/icons/attributes/program	String	AdChoices
pxratio	アイコンまたはコンパニオン広告のピクセル比。高 DPI ディスプレイに使用されます。 パス: /avails/ads/icons/attributes/pxratio または /avails/ads/companionAds/attributes/pxratio	String	1、または 2
type	リソースまたは拡張機能のタイプ。 パス: /avails/ads/extensions/type または /avails/ads/adVerifications/executableResource/type	String	text/javascript
content	拡張機能の内容。 パス: /avails/ads/extensions/content	String	
language	実行可能リソースのプログラミング言語。 パス: /avails/ads/adVerifications/executableResource/language	String	javascript

プロパティ	定義	値のタイプ	例
browserOptional	JavaScript リソースにブラウザのサポートが必要かどうかを示します。 パス: /avails/ads/adVerifications/javaScriptResource/browserOptional	String	true、または false
id	スキーマ内のさまざまな要素の識別子。 パス: /avails/ads/companionAds/attributes/id または /avails/ads/icons/iconClicks/iconClickTracking/id	String	companion-1
assetHeight	コンパニオン広告アセットの高さ。 パス: /avails/ads/companionAds/attributes/assetHeight	String	250
assetWidth	コンパニオン広告アセットの幅。 パス: /avails/ads/companionAds/attributes/assetWidth	String	300
expandedHeight	展開時のコンパニオン広告の高さ。 パス: /avails/ads/companionAds/attributes/expandedHeight	String	600

プロパティ	定義	値のタイプ	例
expandedWidth	展開時のコンパニオン広告の幅。 パス: /avails/ads/companionAds/attributes/expandedWidth	String	600
renderingMode	コンパニオン広告のレンダリングモード。 パス: /avails/ads/companionAds/attributes/renderingMode	String	default、または transparent
adSlotId	コンパニオン広告を表示する広告スロットの ID。 パス: /avails/ads/companionAds/attributes/adSlotId	String	banner-1
creativeType	クリエイティブアセットの MIME タイプ。 パス: /avails/ads/icons/staticResource/creativeType	String	image/png

広告追跡アクティビティのタイミング

クライアント側のレポートでは、プレイヤーは精度の高い追跡イベント (ビーコン) を出力する必要があります。MediaTailor クライアント側の追跡スキーマを使用すると、すべての表示、広告、コンパニオン、オーバーレイ、追跡イベント、タイミングと期間情報、およびさまざまな形式が存在することを確認できます。

プレイヤーは、次の MediaTailor キーと値のペアを使用して、イベントの追跡などの広告イベントアクティビティを再生位置と正確に照合します。

- [startTime](#)
- [startTimeInSeconds](#)
- [adProgramDateTime](#)
- [adId/eventId](#)

HLS と DASH は、この値を実装 `startTime` `startTimeInSeconds` します。

- HLS - `startTime` 値は再生セッションの開始からの相対値です。再生セッションの開始は、ゼロ時間として定義されます。広告の `startTime` は、表示までのすべての EXT-INF セグメント期間の累積値の合計です。広告または追跡イベントが属するセグメントのメディアシーケンス番号は、クライアント側の追跡レスポンス `eventId` の `adId` または `PeriodId` にも対応します。
- DASH:
 - ライブ/動的マニフェスト - `startTime` 値は DASH マニフェスト `MPD@availabilityStartTime` のを基準としています。 `MPD@availabilityStartTime` は、ストリームを消費するすべての MediaTailor セッションのタイミングアンカーです。
 - VOD/静的マニフェスト - `startTime` 値は再生セッションの開始からの相対値です。再生セッションの開始は、ゼロ時間として定義されます。 `avail` 内の各広告は、独自の `Period` 要素内に含まれています。 `Period` 要素には、クライアント側の追跡ペイロードの値と同じ `startTime` 値を持つ `@start` 属性があります。は、クライアント側の追跡レスポンス `eventId` の `adId` または `PeriodId` にも対応します。

Example HLS:

次の例では、MediaTailor セッションが開始され、次のマニフェストがクライアントに提供される最初のマニフェストです。

```
#EXTM3U
#EXT-X-VERSION:6
#EXT-X-TARGETDURATION:6
#EXT-X-MEDIA-SEQUENCE:4603263
#EXT-X-DISCONTINUITY-SEQUENCE:0
#EXT-X-PROGRAM-DATE-TIME:2023-05-03T21:24:23.295678Z
#EXTINF:4.010667,
https://123.cloudfront.net/out/v1/index_1_34.ts
#EXT-X-PROGRAM-DATE-TIME:2023-05-03T21:24:27.306345Z
#EXTINF:4.010667,
https://123.cloudfront.net/out/v1/index_1_35.ts
```

```
#EXT-X-PROGRAM-DATE-TIME:2023-05-03T21:24:31.317012Z
#EXTINF:4.010667,
https://123.cloudfront.net/out/v1/index_1_36.ts
#EXT-X-PROGRAM-DATE-TIME:2023-05-03T21:24:35.327679Z
#EXTINF:4.010667,
https://123.cloudfront.net/out/v1/index_1_37.ts
#EXT-X-PROGRAM-DATE-TIME:2023-05-03T21:24:39.338346Z
#EXTINF:2.538667,
https://123.cloudfront.net/out/v1/index_1_38.ts
#EXT-X-DISCONTINUITY
#EXT-X-KEY:METHOD=NONE
#EXT-X-PROGRAM-DATE-TIME:2023-05-03T21:24:41.453Z
#EXTINF:2.0,
https://123.cloudfront.net/tm/asset_1080_4_8_00001.ts
#EXT-X-PROGRAM-DATE-TIME:2023-05-03T21:24:43.453Z
#EXTINF:2.0,
https://123.cloudfront.net/tm/asset_1080_4_8_00002.ts
#EXT-X-PROGRAM-DATE-TIME:2023-05-03T21:24:45.453Z
#EXTINF:2.0,
https://123.cloudfront.net/tm/asset_1080_4_8_00003.ts
```

クライアント側の追跡 JSON ペイロードでは、次の値が適用されます。

- `startTime`: "PT18.581355S"
- `startTimeInSeconds`: 18.581
- `availProgramDateTime`: "2023-05-03T21:24:41.453Z"
- `adId`: 4603269

Example DASH:

次の例では、MediaTailor セッションはマニフェストでミッドロールを取得します。広告期間である 2 番目の期間の `@start` 属性値には、値に対する相対 `MPD@availabilityStartTime` 値があることに注意してください。この値は、MediaTailor がすべてのセッションについてクライアント側の追跡レスポンス `startTime` フィールドに書き込む値です。

```
<?xml version="1.0" encoding="UTF-8"?>
<MPD availabilityStartTime="2022-10-05T19:38:39.263Z" minBufferTime="PT10S"
  minimumUpdatePeriod="PT2S" profiles="urn:mpeg:dash:profile:isoff-live:2011"
  publishTime="2023-05-03T22:06:48.411Z" suggestedPresentationDelay="PT10S"
  timeShiftBufferDepth="PT1M30S" type="dynamic" xmlns="urn:mpeg:dash:schema:mpd:2011"
  xmlns:scte35="urn:scte:scte35:2013:xml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
```

```
instance" xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 http://standards.iso.org/
itff/PubliclyAvailableStandards/MPEG-DASH_schema_files/DASH-MPD.xsd">
  <BaseURL>https://123.channel-assembly.mediatailor.us-west-2.amazonaws.com/v1/
channel/my-channel/</BaseURL>
  <Location>https://123.cloudfront.net/v1/
dash/94063eadf7d8c56e9e2edd84fdf897826a70d0df/MediaTailor-Live-HLS-DASH/channel/
channel1/dash.mpd?aws.sessionId=794a15e0-2a7f-4941-a537-9d71627984e5</Location>
  <Period id="1683151479166_1" start="PT5042H25M59.903S"
xmlns="urn:mpeg:dash:schema:mpd:2011">
    <BaseURL>https://123.cloudfront.net/out/v1/f1a946be8efa45b0931ea35c9055fb74/
ddb73bf548a44551a0059c346226445a/ea5485198bf497284559efb8172425e/</BaseURL>
    <AdaptationSet ...>
      ...
    </AdaptationSet>
  </Period>
  <Period id="1683151599194_1_1" start="PT5042H27M59.931S">
    <BaseURL>https://123.cloudfront.net/
tm/94063eadf7d8c56e9e2edd84fdf897826a70d0df/fpc5omz5wzd2rdepgieibp23ybyqyrme/</BaseURL>
    <AdaptationSet ...>
      ...
    </AdaptationSet>
  </Period>
</MPD>
```

クライアント側の追跡 JSON ペイロードでは、次の値が適用されます。

- startTime: "PT5042H27M59.931S"
- startTimeInSeconds: 18152879.931
- availProgramDateTime: *null*
- adId: 1683151599194_1_1

クライアント側の広告追跡のプレイヤーコントロールと機能

MediaTailor クライアント側の追跡メタデータは、さまざまなプレイヤーコントロールと機能をサポートしています。次のリストでは、一般的なプレイヤーコントロールについて説明します。

トピック

- [スクラブ](#)
- [広告カウントダウンタイマー](#)
- [スキップ可能な広告](#)

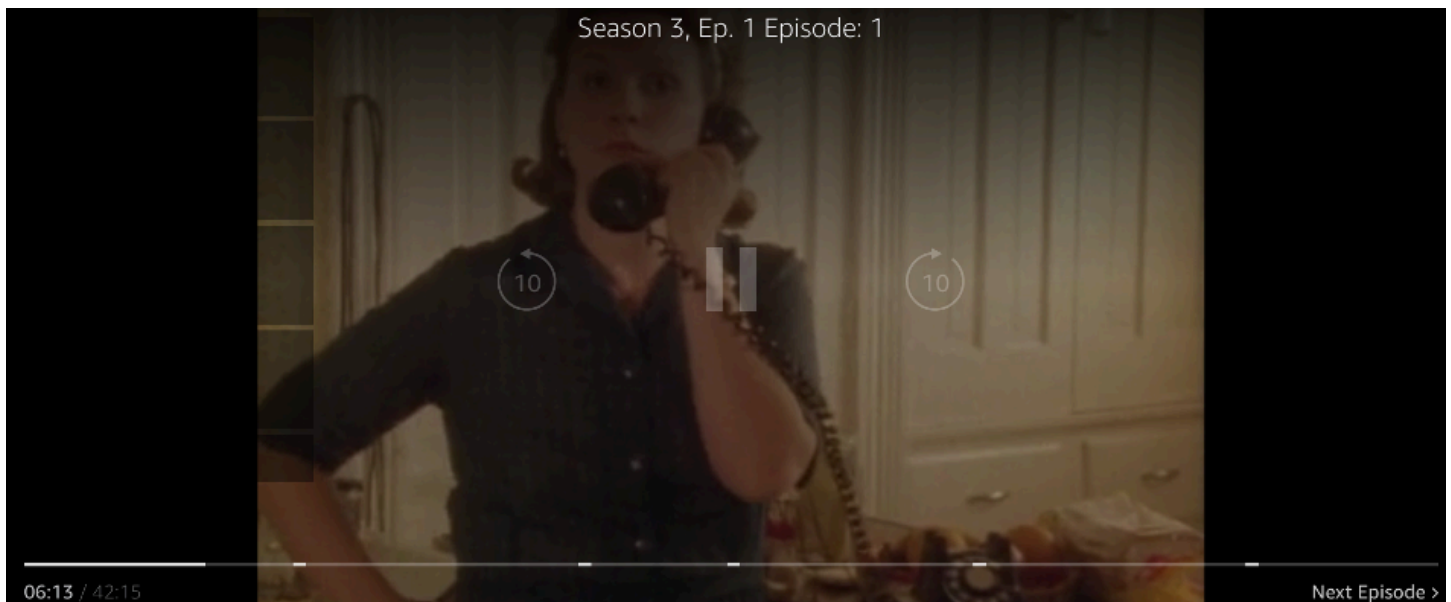
- [広告クリックスルー](#)
- [コンパニオン広告](#)
- [インタラクティブ広告 \(SIMID\)](#)
- [インタラクティブ広告 \(VPAID\)](#)
- [Google Why This Ad \(WTA\) のアイコン](#)

スクラブ

再生エクスペリエンスを向上させるために、プレイヤーは再生タイムラインに広告位置を表示できます。MediaTailor は、これらの広告位置をクライアント側の追跡レスポンス `startTimeInSeconds` の値の形式で利用できるようにします。

Note

一部のストリーミングプロバイダーは、広告位置を超えたスクラブを防止します。



次のクライアント側の追跡ペイロード JSON レスポンスは、`avails` 配列のルート JSON オブジェクト内の `avail` (ad break) 開始時間を示しています。プレイヤーはこのデータを使用して、プレイヤータイムライン上の広告時間枠の場所を 28 秒で表示します。

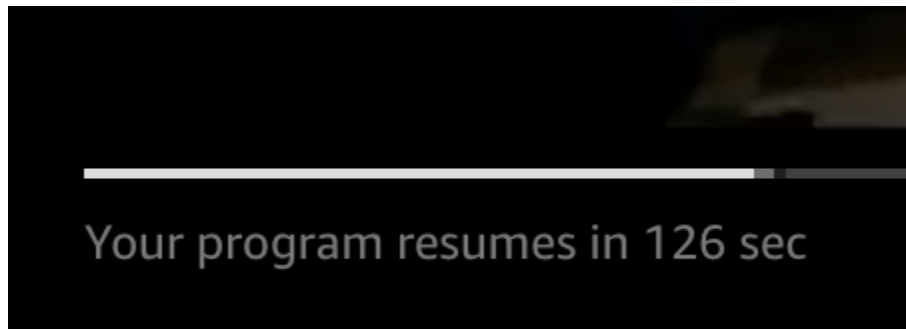
```
{  
  "avails": [  

```

```
{
  "adBreakTrackingEvents": [],
  "adMarkerDuration": null,
  "ads": [...],
  "availId": "7",
  "availProgramDateTime": null,
  "duration": "PT30S",
  "durationInSeconds": 30,
  "meta": null,
  "nonLinearAdsList": [],
  "startTime": "PT28S",
  "startTimeInSeconds": 28
},
{
  "dashAvailabilityStartTime": null,
  "hlsAnchorMediaSequenceNumber": null,
  "nextToken": "UFQxMk0zNC44NjhTXzIwMjMtMDctMjFUMjA6MjM6MDcuNzc1NzE2MzAyWl8x",
  "nonLinearAvails": []
}
```

広告カウントダウンタイマー

MediaTailor では、広告カウントダウンタイマーを使用して、広告ブレークの表示中にオーディエンスのエンゲージメントを維持できます。対象者はタイマーを使用して、広告時間枠が終了し、プログラムが再開されるタイミングを把握できます。



広告カウントダウンタイマーで役割を果たすクライアント側の追跡メタデータの要素は `startTime`、`startTimeInSeconds`、`duration`、および `durationInSeconds` です。プレイヤーはこのメタデータを個別に追跡するセッションの経過時間とともに使用して、タイマーを表示するタイミングとカウントダウンする値を決定します。

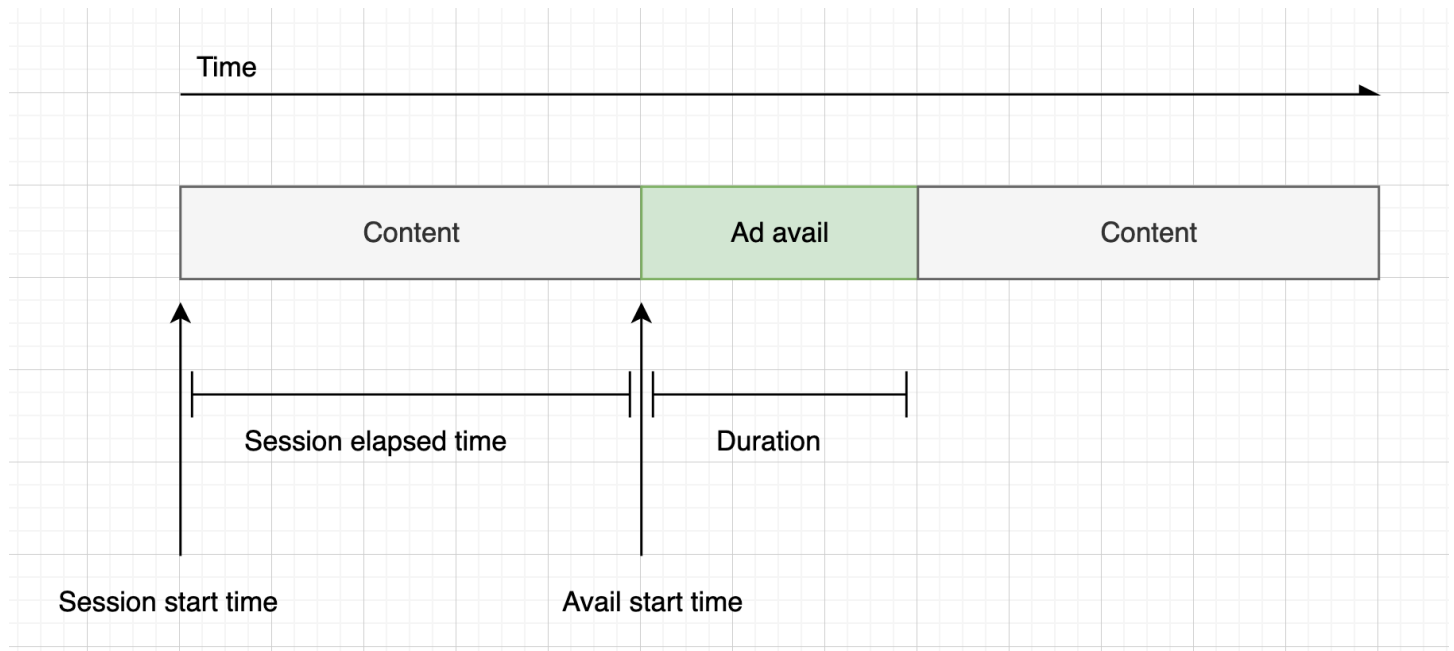
次のクライアント側の追跡ペイロード JSON レスポンスは、広告カウントダウンタイマーを表示するために必要な情報を示しています。

```
{
  "avails": [
    {
      "adBreakTrackingEvents": [],
      "adMarkerDuration": null,
      "ads": [...],
      "availId": "7",
      "availProgramDateTime": null,
      "duration": "PT30S",
      "durationInSeconds": 30,
      "meta": null,
      "nonLinearAdsList": [],
      "startTime": "PT28S",
      "startTimeInSeconds": 28
    }
  ],
  "dashAvailabilityStartTime": null,
  "hlsAnchorMediaSequenceNumber": null,
  "nextToken": "UFQxMk0zNC44NjhTXzIwMjMtMDctMjFUMjA6MjM6MDcuNzc1NzE2MzAyWl8x",
  "nonLinearAvails": []
}
```

セッションの経過時間が avail の開始時間に達すると、プレイヤーは avail の期間に一致する値を持つカウントダウンタイマーを表示します。カウントダウンタイマーの値は、経過時間が表示の開始時間を超えると減少します。

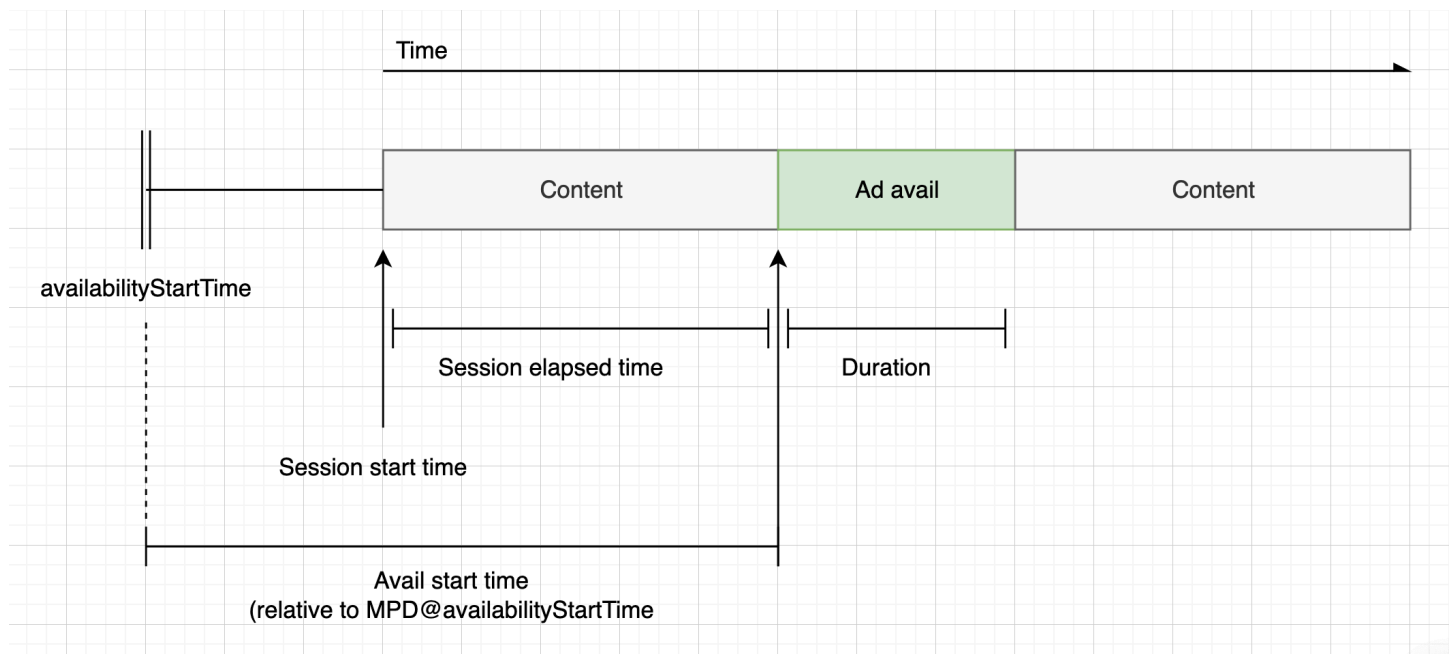
Example式: HLS (ライブと VOD) と DASH (VOD) のカウントダウンタイマー

- $\text{session\_start\_time} = \text{すべてのEXT-INF継続時間値の合計} - 3 \text{ つの最新のEXT-INFメディアシーケンスの継続時間値}$
- $\text{タイマー値} = \text{duration} - (\text{session\_elapsed\_time} - \text{startTime})$



Example 計算式: DASH のカウントダウンタイマー (ライブ)

- $\text{session\_start\_time} = (\text{最新のセグメントの } \text{startTime} + \text{duration}) / \text{timescale} - \text{MPD@suggestedPresentationDelay}$
- $\text{タイマー値} = \text{duration} - (\text{session\_elapsed\_time} - \text{startTime})$



スキップ可能な広告

スキップ可能な広告は、ビューワーが一部の広告をスキップしてプログラムの表示を再開できるようにする広告スポットです。VAST では、`Linear@skipOffset` 属性はスキップ可能な広告を識別します。

次の VAST レスポンスは、スキップ可能な広告の使用方法を示しています。

```
<?xml version="1.0" encoding="UTF-8"?>
<VAST xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="vast.xsd" version="3.0">
  <Ad>
    <InLine>
      ...
      <Creatives>
        <Creative id="1" sequence="1">
          <Linear skipoffset="00:00:05">
            <Duration>00:00:15</Duration>
            <MediaFiles>
              <MediaFile id="EMT" delivery="progressive" width="640" height="360"
                type="video/mp4" bitrate="143" scalable="true" maintainAspectRatio="true"><![
CDATA[https://ads.com/file.mp4]]></MediaFile>
              </MediaFiles>
            </Linear>
          </Creative>
        </Creatives>
        ...
      </InLine>
    </Ad>
  </VAST>
```

次のクライアント側の追跡ペイロード JSON レスポンスは、ads 配列内の広告メタデータを示しています。配列には、MediaTailor が VAST レスポンスから取得した `skipOffset` 値が含まれています。

```
{
  "avails": [
    {
      "adBreakTrackingEvents": [],
      "adMarkerDuration": null,
      "ads": [
        {
          "adId": "1",
```

```
    "adParameters": "",
    "adProgramDateTime": "2023-07-31T16:11:40.693Z",
    "adSystem": "2.0",
    "adTitle": "AD-skiing-15",
    "adVerifications": [],
    "companionAds": [...],
    "creativeId": "1",
    "creativeSequence": "1",
    "duration": "PT15.015S",
    "durationInSeconds": 15.015,
    "extensions": [],
    "mediaFiles": {
      "mediaFilesList": [],
      "mezzanine": ""
    },
    "skipOffset": "00:00:05",
    "startTime": "PT9.943S",
    "startTimeInSeconds": 9.943,
    "trackingEvents": [
      {
        "beaconUrls": [
          "https://adserverbeaconing.com/v1/impression"
        ],
        "duration": "PT15.015S",
        "durationInSeconds": 15.015,
        "eventId": "2697726",
        "eventProgramDateTime": null,
        "eventType": "impression",
        "startTime": "PT9.943S",
        "startTimeInSeconds": 9.943
      }
    ],
    "vastAdId": ""
  }
],
"availId": "2697726",
"availProgramDateTime": "2023-07-31T16:11:40.693Z",
"duration": "PT15.015S",
"durationInSeconds": 15.015,
"meta": null,
"nonLinearAdsList": [],
"startTime": "PT9.943S",
"startTimeInSeconds": 9.943
}
```

```
],  
"dashAvailabilityStartTime": null,  
"hlsAnchorMediaSequenceNumber": null,  
"nextToken": "",  
"nonLinearAvails": []  
}
```

広告クリックスルー

クリックスルー URIs を使用すると、広告が視聴者の注意を引くのにどの程度成功したかを広告主が測定できます。ビューワーが進行中の広告のアクティブな動画フレームをクリックすると、ウェブブラウザは広告主のホームページまたはキャンペーンランディングページの URI を開きます。プレイヤー開発者は、広告ビデオにボタンやラベルを重ねるなどのクリック動作を決定し、クリックして詳細を確認するメッセージを表示します。プレイヤー開発者は、視聴者がアクティブな動画フレームをクリックした後、広告の動画を一時停止することがよくあります。



Click here for deals on Amazon.com

MediaTailor は、VAST レスポンスで返されたリニアビデオクリックスルーイベント URLs を解析して使用できます。次の VAST レスポンスは、広告クリックスルーの例を示しています。

```
<?xml version="1.0" encoding="UTF-8"?>
<VAST xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="vast.xsd" version="3.0">
  <Ad>
    <Inline>
      ...
      <Creatives>
        <Creative id="1" sequence="1">
          <Linear>
            <Duration>00:00:15</Duration>
            <MediaFiles>
              <MediaFile id="EMT" delivery="progressive" width="1280" height="720"
                type="video/mp4" bitrate="143" scalable="true" maintainAspectRatio="true"><![
[CDATA[https://ads.com/file.mp4]]></MediaFile>
            </MediaFiles>
            <VideoClicks>
              <ClickThrough id="EMT"><![CDATA[https://aws.amazon.com]]></ClickThrough>
              <ClickTracking id="EMT"><![CDATA[https://myads.com/beaconing/
event=clicktracking]]></ClickTracking>
            </VideoClicks>
          </Linear>
        </Creative>
      </Creatives>
      ...
    </Inline>
  </Ad>
</VAST>
```

次のクライアント側の追跡ペイロード JSON レスポンスは、MediaTailor が trackingEvents 配列内にクリックスルー URL とクリック追跡 URLs を表示する方法を示しています。clickThrough イベントタイプはクリックスルー広告を表し、clickTracking イベントタイプはクリック追跡 URL を表します。

```
{
  "avails": [
    {
      "adBreakTrackingEvents": [],
      "adMarkerDuration": null,
      "ads": [
```

```
{
  "adId": "1",
  "adParameters": "",
  "adProgramDateTime": "2023-07-31T16:53:40.577Z",
  "adSystem": "2.0",
  "adTitle": "1",
  "adVerifications": [],
  "companionAds": [],
  "creativeId": "00006",
  "creativeSequence": "1",
  "duration": "PT14.982S",
  "durationInSeconds": 14.982,
  "extensions": [],
  "mediaFiles": {
    "mediaFilesList": [],
    "mezzanine": ""
  },
  "skipOffset": null,
  "startTime": "PT39.339S",
  "startTimeInSeconds": 39.339,
  "trackingEvents": [
    {
      "beaconUrls": [
        "https://myads.com/beaconing/event=impression"
      ],
      "duration": "PT14.982S",
      "durationInSeconds": 14.982,
      "eventId": "2698188",
      "eventProgramDateTime": null,
      "eventType": "impression",
      "startTime": "PT39.339S",
      "startTimeInSeconds": 39.339
    },
    {
      "beaconUrls": [
        "https://aws.amazon.com"
      ],
      "duration": "PT14.982S",
      "durationInSeconds": 14.982,
      "eventId": "2698188",
      "eventProgramDateTime": null,
      "eventType": "clickThrough",
      "startTime": "PT39.339S",
      "startTimeInSeconds": 39.339
    }
  ]
}
```

```

    },
    {
      "beaconUrls": [
        "https://myads.com/beaconing/event=clicktracking"
      ],
      "duration": "PT14.982S",
      "durationInSeconds": 14.982,
      "eventId": "2698795",
      "eventProgramDateTime": null,
      "eventType": "clickTracking",
      "startTime": "PT39.339S",
      "startTimeInSeconds": 39.339
    }
  ],
  "vastAdId": ""
}
],
"availId": "2698188",
"availProgramDateTime": "2023-07-31T16:53:40.577Z",
"duration": "PT14.982S",
"durationInSeconds": 14.982,
"meta": null,
"nonLinearAdsList": [],
"startTime": "PT39.339S",
"startTimeInSeconds": 39.339
}
],
"dashAvailabilityStartTime": null,
"hlsAnchorMediaSequenceNumber": null,
"nextToken": "UFQz0S4zMz1TXzIwMjMtMDctMzFUMTY6NTQ6MDQu0DA1Mzk2NTI5Wl8x",
"nonLinearAvails": []
}

```

コンパニオン広告

コンパニオン広告は、線形クリエイティブと一緒に表示されます。コンパニオン広告を使用して、製品、ロゴ、ブランド情報を表示することで、広告スポットの有効性を高めます。ディスプレイ広告には、クイックレスポンス (QR) コードとクリック可能なエリアが用意されており、オーディエンスのエンゲージメントを高めることができます。

MediaTailor は VAST レスポンスでコンパニオン広告をサポートしています。

StaticResource、iFrameResourceおよび HTMLResourceノードからそれぞれメタデータを渡すことができます。

次の VAST レスポンスは、線形広告とコンパニオン広告の場所と形式の例を示しています。

```
<?xml version="1.0" encoding="UTF-8"?>
<VAST xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="vast.xsd" version="3.0">
  <Ad>
    <Inline>
      ...
    <Creatives>
      <Creative id="1" sequence="1">
        <Linear>
          <Duration>00:00:10</Duration>
          <MediaFiles>
            <MediaFile id="EMT" delivery="progressive" width="640" height="360"
              type="video/mp4" bitrate="143" scalable="true" maintainAspectRatio="true"><![
[CDATA[https://ads.com/file.mp4]]></MediaFile>
            </MediaFiles>
          </Linear>
        </Creative>
        <Creative id="2" sequence="1">
          <CompanionAds>
            <Companion id="2" width="300" height="250">
              <StaticResource creativeType="image/png"><![CDATA[https://emt.com/companion/9973499273]]></StaticResource>
              <TrackingEvents>
                <Tracking event="creativeView"><![CDATA[https://beacon.com/1]]></
Tracking>
              </TrackingEvents>
              <CompanionClickThrough><![CDATA[https://beacon.com/2]]></
CompanionClickThrough>
            </Companion>
            <Companion id="3" width="728" height="90">
              <StaticResource creativeType="image/png"><![CDATA[https://emt.com/companion/1238901823]]></StaticResource>
              <TrackingEvents>
                <Tracking event="creativeView"><![CDATA[https://beacon.com/3]]></
Tracking>
              </TrackingEvents>
              <CompanionClickThrough><![CDATA[https://beacon.com/4]]></
CompanionClickThrough>
            </Companion>
          </CompanionAds>
        </Creative>
      </Creatives>
```

```
...
</Inline>
</Ad>
</VAST>
```

データは、/avail/x/ads/y/companionAds リストのクライアント側の追跡レスポンスに表示されます。各リニアクリエイティブには、最大 6 つのコンパニオン広告を含めることができます。以下の例に示すように、コンパニオン広告はリストに表示されます。

Note

ベストプラクティスとして、アプリケーションデベロッパーは、クリエイティブの最後にコンパニオン広告を明示的に削除またはアンロードするロジックを実装する必要があります。

```
{
  "avails": [
    {
      "adBreakTrackingEvents": [],
      "adMarkerDuration": null,
      "ads": [
        {
          "adId": "0",
          "adParameters": "",
          "adProgramDateTime": null,
          "adSystem": "EMT",
          "adTitle": "sample",
          "adVerifications": [],
          "companionAds": [
            {
              "adParameters": null,
              "altText": null,
              "attributes": {
                "adSlotId": null,
                "apiFramework": null,
                "assetHeight": null,
                "assetWidth": null,
                "expandedHeight": null,
                "expandedWidth": null,
                "height": "250",
                "id": "2",
                "pxratio": null,

```



```
        "renderingMode": null,
        "width": "300"
    },
    "companionClickThrough": "https://beacon.com/2",
    "companionClickTracking": null,
    "htmlResource": null,
    "iFrameResource": null,
    "sequence": "1",
    "staticResource": "https://emt.com/companion/9973499273",
    "trackingEvents": [
        {
            "beaconUrls": [
                "https://beacon.com/1"
            ],
            "eventType": "creativeView"
        }
    ]
},
{
    "adParameters": null,
    "altText": null,
    "attributes": {
        "adSlotId": null,
        "apiFramework": null,
        "assetHeight": null,
        "assetWidth": null,
        "expandedHeight": null,
        "expandedWidth": null,
        "height": "90",
        "id": "3",
        "pxratio": null,
        "renderingMode": null,
        "width": "728"
    },
    "companionClickThrough": "https://beacon.com/4",
    "companionClickTracking": null,
    "htmlResource": null,
    "iFrameResource": null,
    "sequence": "1",
    "staticResource": "https://emt.com/companion/1238901823",
    "trackingEvents": [
        {
            "beaconUrls": [
                "https://beacon.com/3"
```

```
        ],
        "eventType": "creativeView"
      }
    ]
  }
],
"creativeId": "1",
"creativeSequence": "1",
"duration": "PT10S",
"durationInSeconds": 10,
"extensions": [],
"mediaFiles": {
  "mediaFilesList": [],
  "mezzanine": ""
},
"skipOffset": null,
"startTime": "PT0S",
"startTimeInSeconds": 0,
"trackingEvents": [
  {
    "beaconUrls": [
      "https://beacon.com/impression/1"
    ],
    "duration": "PT10S",
    "durationInSeconds": 10,
    "eventId": "0",
    "eventProgramDateTime": null,
    "eventType": "impression",
    "startTime": "PT0S",
    "startTimeInSeconds": 0
  }
],
"vastAdId": ""
}
],
"availId": "0",
"availProgramDateTime": null,
"duration": "PT10S",
"durationInSeconds": 10,
"meta": null,
"nonLinearAdsList": [],
"startTime": "PT0S",
"startTimeInSeconds": 0
}
```

```

],
"dashAvailabilityStartTime": null,
"hlsAnchorMediaSequenceNumber": null,
"nextToken": "UFQxMFNFmJyMy0wNy0wNlQyMToxMDowOC42NzQ4NDA1NjJaXzE%3D",
"nonLinearAvails": []
}

```

インタラクティブ広告 (SIMID)

SecureInteractive Media Interface Definition (SIMID) は、インタラクティブ広告局 (IAB) の VAST 4.x 標準で導入されたインタラクティブ広告の標準です。SIMID は、インタラクティブ要素のロードをプレイヤーのプライマリリニアクリエイティブから切り離し、VAST レスポンスで両方を参照します。MediaTailor は、再生エクスペリエンスを維持するためにプライマリクリエイティブをステッチし、インタラクティブコンポーネントのメタデータをクライアント側の追跡レスポンスに配置します。

次の VAST 4 レスポンスの例では、SIMID ペイロードは InteractiveCreativeFile ノード内にあります。

```

<?xml version="1.0"?>
<VAST xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="vast.xsd" version="3.0">
  <Ad id="1234567">
    <Inline>
      <AdSystem>SampleAdSystem</AdSystem>
      <AdTitle>Linear SIMID Example</AdTitle>
      <Description>SIMID example</Description>
      <Error>https://www.beacons.com/error</Error>
      <Impression>https://www.beacons.com/impression</Impression>
      <Creatives>
        <Creative sequence="1">
          <Linear>
            <Duration>00:00:15</Duration>
            <TrackingEvents>
              ...
            </TrackingEvents>
            <VideoClicks>
              <ClickThrough id="123">https://aws.amazon.com</ClickThrough>
              <ClickTracking id="123">https://www.beacons.com/click</ClickTracking>
            </VideoClicks>
            <MediaFiles>
              <MediaFile delivery="progressive" type="video/mp4">

```

```

https://interactive-ads.com/interactive-media-ad-sample/media/file.mp4
    </MediaFile>
    <InteractiveCreativeFile type="text/html" apiFramework="SIMID"
variableDuration="true">
https://interactive-ads.com/interactive-media-ad-sample/sample\_simid.html
    </InteractiveCreativeFile>
  </MediaFiles>
</Linear>
</Creative>
</Creatives>
</InLine>
</Ad>
</VAST>

```

次の VAST 3 レスポンスでは、SIMID ペイロードは Extensions ノード内にあります。

```

<?xml version="1.0"?>
<VAST xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="vast.xsd" version="3.0">
  <Ad id="1234567">
    <InLine>
      <AdSystem>SampleAdSystem</AdSystem>
      <AdTitle>Linear SIMID Example</AdTitle>
      <Description>SIMID example</Description>
      <Impression>https://www.beacons.com/impression</Impression>
      <Creatives>
        <Creative id="1" sequence="1">
          <Linear>
            <Duration>00:00:15</Duration>
            <TrackingEvents>
              ...
            </TrackingEvents>
            <VideoClicks>
              <ClickThrough id="123">https://aws.amazon.com</ClickThrough>
              <ClickTracking id="123">https://myads.com/beaconing/event=clicktracking</
ClickTracking>
            </VideoClicks>
            <MediaFiles>
              <MediaFile delivery="progressive" type="video/mp4">
https://interactive-ads.com/interactive-media-ad-sample/media/file.mp4

```

```

        </MediaFile>
    </MediaFiles>
</Linear>
</Creative>
</Creatives>
<Extensions>
    <Extension type="InteractiveCreativeFile">
        <InteractiveCreativeFile type="text/html" apiFramework="SIMID"
variableDuration="true">
            https://interactive-ads.com/interactive-media-ad-sample/sample\_simid.html
        </InteractiveCreativeFile>
    </Extension>
</Extensions>
</InLine>
</Ad>
</VAST>

```

次のクライアント側の追跡レスポンスでは、SIMID データが /avails/x/ads/y/extensions リストに表示されます。

```

{
  "avails": [
    {
      "adBreakTrackingEvents": [],
      "adMarkerDuration": null,
      "ads": [
        {
          "adId": "1",
          "adParameters": "",
          "adProgramDateTime": "2023-07-31T16:53:40.577Z",
          "adSystem": "2.0",
          "adTitle": "Linear SIMID Example",
          "adVerifications": [],
          "companionAds": [],
          "creativeId": "1",
          "creativeSequence": "1",
          "duration": "PT14.982S",
          "durationInSeconds": 14.982,
          "extensions": [
            {
              "content": "<InteractiveCreativeFile type=\"text/html\" apiFramework=
\"SIMID\" variableDuration=\"true\">\nhttps://interactive-ads.com/interactive-media-ad-sample/sample\_simid.html</InteractiveCreativeFile>",

```

```
    "type": "InteractiveCreativeFile"
  }
],
"mediaFiles": {
  "mediaFilesList": [],
  "mezzanine": ""
},
"skipOffset": null,
"startTime": "PT39.339S",
"startTimeInSeconds": 39.339,
"trackingEvents": [
  {
    "beaconUrls": [
      "https://myads.com/beaconing/event=impression"
    ],
    "duration": "PT14.982S",
    "durationInSeconds": 14.982,
    "eventId": "2698188",
    "eventProgramDateTime": null,
    "eventType": "impression",
    "startTime": "PT39.339S",
    "startTimeInSeconds": 39.339
  },
  {
    "beaconUrls": [
      "https://aws.amazon.com"
    ],
    "duration": "PT14.982S",
    "durationInSeconds": 14.982,
    "eventId": "2698188",
    "eventProgramDateTime": null,
    "eventType": "clickThrough",
    "startTime": "PT39.339S",
    "startTimeInSeconds": 39.339
  },
  {
    "beaconUrls": [
      "https://myads.com/beaconing/event=clicktracking"
    ],
    "duration": "PT14.982S",
    "durationInSeconds": 14.982,
    "eventId": "2698795",
    "eventProgramDateTime": null,
    "eventType": "clickTracking",
  }
]
```

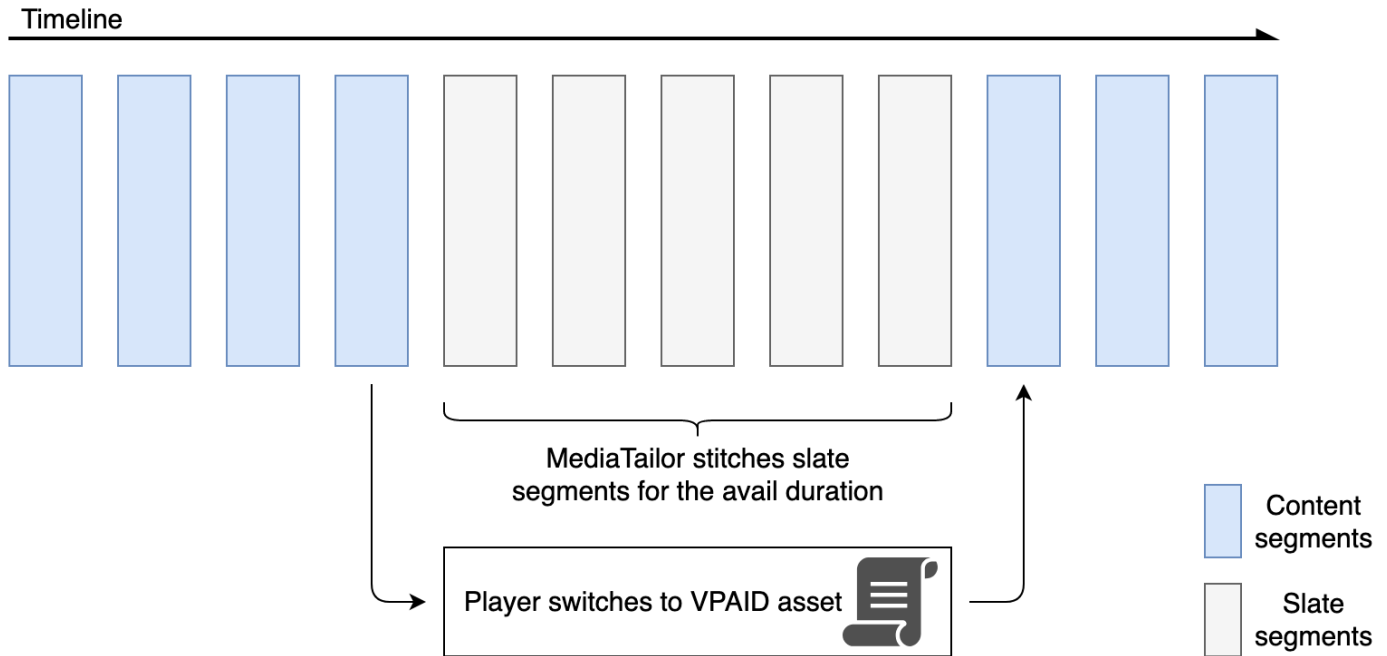
```
        "startTime": "PT39.339S",
        "startTimeInSeconds": 39.339
      }
    ],
    "vastAdId": ""
  }
],
"availId": "2698188",
"availProgramDateTime": "2023-07-31T16:53:40.577Z",
"duration": "PT14.982S",
"durationInSeconds": 14.982,
"meta": null,
"nonLinearAdsList": [],
"startTime": "PT39.339S",
"startTimeInSeconds": 39.339
}
],
"dashAvailabilityStartTime": null,
"hlsAnchorMediaSequenceNumber": null,
"nextToken": "UFQz0S4zMz1TXzIwMjMtMDctMzFUMTY6NTQ6MDQu0DA1Mzk2NTI5W18x",
"nonLinearAvails": []
}
```

インタラクティブ広告 (VPAID)

Video Player 広告インターフェイス定義 (VPAID) は、広告インタラクティブやその他の機能を有効にする広告とビデオプレーヤーの間のプロトコルを指定します。ライブストリームの場合、MediaTailor は、表示期間中にスレートセグメントを にステッチし、ビデオプレーヤーが消費するクライアント側の追跡レスポンスに VPAID クリエイティブのメタデータを配置することで、VPAID 形式をサポートします。プレーヤーは VPAID ファイルをダウンロードして線形クリエイティブを再生し、クライアントのスクリプトを実行します。プレーヤーはスレートセグメントを再生しないでください。

Note

VPAID は VAST 4.1 以降廃止されました。



次の例は、VAST レスポンスの VPAID コンテンツを示しています。

```
<?xml version="1.0"?>
<VAST xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="vast.xsd" version="3.0">
  <Ad id="1234567">
    <Inline>
      <AdSystem>GDFP</AdSystem>
      <AdTitle>VPAID</AdTitle>
      <Description>Vpaid Linear Video Ad</Description>
      <Error>http://www.example.com/error</Error>
      <Impression>http://www.example.com/impression</Impression>
      <Creatives>
        <Creative sequence="1">
          <Linear>
            <Duration>00:00:00</Duration>
            <TrackingEvents>
              <Tracking event="start">http://www.example.com/start</Tracking>
              <Tracking event="firstQuartile">http://www.example.com/firstQuartile</
Tracking>
              <Tracking event="midpoint">http://www.example.com/midpoint</Tracking>
              <Tracking event="thirdQuartile">http://www.example.com/thirdQuartile</
Tracking>
```



```

    <Tracking event="complete">http://www.example.com/complete</Tracking>
    <Tracking event="mute">http://www.example.com/mute</Tracking>
    <Tracking event="unmute">http://www.example.com/unmute</Tracking>
    <Tracking event="rewind">http://www.example.com/rewind</Tracking>
    <Tracking event="pause">http://www.example.com/pause</Tracking>
    <Tracking event="resume">http://www.example.com/resume</Tracking>
    <Tracking event="fullscreen">http://www.example.com/fullscreen</Tracking>
    <Tracking event="creativeView">http://www.example.com/creativeView</
Tracking>
    <Tracking event="acceptInvitation">http://www.example.com/
acceptInvitation</Tracking>
  </TrackingEvents>
  <AdParameters><![CDATA[ {"videos":[ {"url":"https://my-ads.com/interactive-
media-ads/media/media_linear_VPAID.mp4","mimetype":"video/mp4"}]] ]]></AdParameters>
  <VideoClicks>
    <ClickThrough id="123">http://google.com</ClickThrough>
    <ClickTracking id="123">http://www.example.com/click</ClickTracking>
  </VideoClicks>
  <MediaFiles>
    <MediaFile delivery="progressive" apiFramework="VPAID" type="application/
javascript" width="640" height="480"> https://googleads.github.io/googleads-ima-html5/
vpaid/linear/VpaidVideoAd.js </MediaFile>
  </MediaFiles>
</Linear>
</Creative>
</Creatives>
</InLine>
</Ad>
</VAST>

```

次の例は、追跡情報を示しています。

```

{
  "avails": [
    {
      "adBreakTrackingEvents": [],
      "adMarkerDuration": null,
      "ads": [
        {
          "adId": "1",
          "adParameters": "",
          "adProgramDateTime": "2023-07-31T16:53:40.577Z",
          "adSystem": "2.0",

```

```
"adTitle": "1",
"adVerifications": [],
"companionAds": [],
"creativeId": "000006",
"creativeSequence": "1",
"duration": "PT14.982S",
"durationInSeconds": 14.982,
"extensions": [],
"mediaFiles": {
  "mediaFilesList": [],
  "mezzanine": ""
},
"skipOffset": null,
"startTime": "PT39.339S",
"startTimeInSeconds": 39.339,
"trackingEvents": [
  {
    "beaconUrls": [
      "https://myads.com/beaconing/event=impression"
    ],
    "duration": "PT14.982S",
    "durationInSeconds": 14.982,
    "eventId": "2698188",
    "eventProgramDateTime": null,
    "eventType": "impression",
    "startTime": "PT39.339S",
    "startTimeInSeconds": 39.339
  },
  {
    "beaconUrls": [
      "https://aws.amazon.com"
    ],
    "duration": "PT14.982S",
    "durationInSeconds": 14.982,
    "eventId": "2698188",
    "eventProgramDateTime": null,
    "eventType": "clickThrough",
    "startTime": "PT39.339S",
    "startTimeInSeconds": 39.339
  },
  {
    "beaconUrls": [
      "https://myads.com/beaconing/event=clicktracking"
    ],
```

```

        "duration": "PT14.982S",
        "durationInSeconds": 14.982,
        "eventId": "2698795",
        "eventProgramDateTime": null,
        "eventType": "clickTracking",
        "startTime": "PT39.339S",
        "startTimeInSeconds": 39.339
    }
],
    "vastAdId": ""
}
],
    "availId": "2698188",
    "availProgramDateTime": "2023-07-31T16:53:40.577Z",
    "duration": "PT14.982S",
    "durationInSeconds": 14.982,
    "meta": null,
    "nonLinearAdsList": [],
    "startTime": "PT39.339S",
    "startTimeInSeconds": 39.339
}
],
    "dashAvailabilityStartTime": null,
    "hlsAnchorMediaSequenceNumber": null,
    "nextToken": "UFQz0S4zMz1TXzIwMjMtMDctMzFUMTY6NTQ6MDQu0DA1Mzk2NTI5Wl8x",
    "nonLinearAvails": []
}{
    "avails": [
        {
            "adBreakTrackingEvents": [],
            "adMarkerDuration": null,
            "ads": [
                {
                    "adId": "2922274",
                    "adParameters": "",
                    "adProgramDateTime": "2023-08-14T19:49:53.998Z",
                    "adSystem": "Innovid Ads",
                    "adTitle": "VPAID",
                    "adVerifications": [],
                    "companionAds": [],
                    "creativeId": "",
                    "creativeSequence": "",
                    "duration": "PT16.016S",
                    "durationInSeconds": 16.016,

```

```

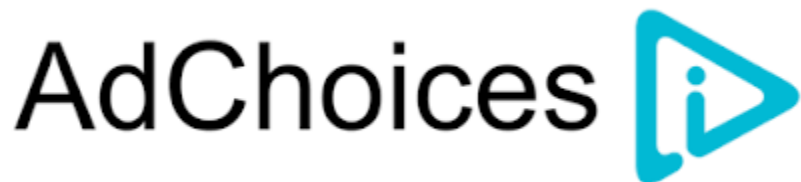
    "extensions": [],
    "mediaFiles": {
      "mediaFilesList": [
        {
          "apiFramework": "VPAID",
          "bitrate": 0,
          "codec": null,
          "delivery": "progressive",
          "height": 9,
          "id": "",
          "maintainAspectRatio": false,
          "maxBitrate": 0,
          "mediaFileUri": "http://my-ads.com/mobileapps/js/vpaid/1h41kg?
cb=178344c0-8e67-281a-58ca-962e4987cd60&deviceid=&ivc=",
          "mediaType": "application/javascript",
          "minBitrate": 0,
          "scalable": false,
          "width": 16
        }
      ],
      "mezzanine": "http://my-ads.com/mobileapps/js/vpaid/1h41kg?
cb=178344c0-8e67-281a-58ca-962e4987cd60&deviceid=&ivc="
    },
    "skipOffset": null,
    "startTime": "PT8M42.289S",
    "startTimeInSeconds": 522.289,
    "trackingEvents": [
      {
        "beaconUrls": [
          "about:blank"
        ],
        "duration": "PT16.016S",
        "durationInSeconds": 16.016,
        "eventId": "2922274",
        "eventProgramDateTime": null,
        "eventType": "impression",
        "startTime": "PT8M42.289S",
        "startTimeInSeconds": 522.289
      }
    ],
    "vastAdId": "1h41kg"
  }
],
  "availId": "2922274",

```

```
    "availProgramDateTime": "2023-08-14T19:49:53.998Z",
    "duration": "PT16.016S",
    "durationInSeconds": 16.016,
    "meta": null,
    "nonLinearAdsList": [],
    "startTime": "PT8M42.289S",
    "startTimeInSeconds": 522.289
  }
],
"dashAvailabilityStartTime": null,
"hlsAnchorMediaSequenceNumber": null,
"nextToken": "UFQ4TTQyLjI4OVNfMjAyMy0wOC0xNFQxOT01MD00MS4zOTc5MjAzODVaXzE%3D",
"nonLinearAvails": []
}
```

Google Why This Ad (WTA) のアイコン

AdChoices は、広告がどのようにターゲットにされたかなど、表示される広告に関する情報を視聴者に提供する業界標準です。



MediaTailor クライアント側の追跡 API は、VAST レスポンスの VAST 拡張機能ノードで伝送されるアイコンメタデータをサポートします。VAST レスポンスの WTA の詳細については、[この VAST XML レスポンスのサンプル](#)を参照してください。

Note

MediaTailor は現在、VAST バージョン 3 のみをサポートしています。

```
<VAST>
  <Ad>
    <Inline>
      ...
    <Extensions>
      <Extension type="IconClickFallbackImages">
```

```

    <IconClickFallbackImages program="GoogleWhyThisAd">
      <IconClickFallbackImage width="400" height="150">
        <AltText>Alt icon fallback</AltText>
        <StaticResource creativeType="image/png"><![CDATA[https://storage.googleapis.com/interactive-media-ads/images/wta_dialog.png]]></StaticResource>
      </IconClickFallbackImage>
    </IconClickFallbackImages>
    <IconClickFallbackImages program="AdChoices">
      <IconClickFallbackImage width="400" height="150">
        <AltText>Alt icon fallback</AltText>
        <StaticResource creativeType="image/png"><![CDATA[https://storage.googleapis.com/interactive-media-ads/images/wta_dialog.png?size=1x]]></StaticResource>
      </IconClickFallbackImage>
      <IconClickFallbackImage width="800" height="300">
        <AltText>Alt icon fallback</AltText>
        <StaticResource creativeType="image/png"><![CDATA[https://storage.googleapis.com/interactive-media-ads/images/wta_dialog.png?size=2x]]></StaticResource>
      </IconClickFallbackImage>
    </IconClickFallbackImages>
  </Extension>
</Extensions>
</InLine>
</Ad>
</VAST>

```

次の例は、/avails/x/ads/y/extensions リスト内のクライアント側の追跡レスポンスを示しています。

```

{
  "avails": [
    {
      "adBreakTrackingEvents": [],
      "adMarkerDuration": null,
      "ads": [
        {
          "adId": "0",
          "adParameters": "",
          "adProgramDateTime": null,
          "adSystem": "GDFP",
          "adTitle": "Google Why This Ad VAST 3 Sample",
          "adVerifications": [],

```

```

    "companionAds": [],
    "creativeId": "7891011",
    "creativeSequence": "1",
    "duration": "PT10S",
    "durationInSeconds": 10,
    "extensions": [
      {
        "content": "<IconClickFallbackImages program=\"GoogleWhyThisAd\">
          <IconClickFallbackImage height=\"150\" width=\"400\">
            <AltText>Alt icon fallback</AltText>
            <StaticResource creativeType=\"image/png\"><![CDATA[https://storage.googleapis.com/interactive-media-ads/images/wta_dialog.png]]>
          </StaticResource>
        </IconClickFallbackImage>
      </IconClickFallbackImages>
      <IconClickFallbackImages program=\"AdChoices\">
        <IconClickFallbackImage height=\"150\" width=\"400\">
          <AltText>Alt icon fallback</AltText>
          <StaticResource creativeType=\"image/png\"><![CDATA[https://storage.googleapis.com/interactive-media-ads/images/wta_dialog.png?size=1x]]>
        </StaticResource>
      </IconClickFallbackImage>
      <IconClickFallbackImage height=\"300\" width=\"800\">
        <AltText>Alt icon fallback</AltText>
        <StaticResource creativeType=\"image/png\"><![CDATA[https://storage.googleapis.com/interactive-media-ads/images/wta_dialog.png?size=2x]]>
      </StaticResource>
    </IconClickFallbackImage>
  </IconClickFallbackImages>",
        "type": "IconClickFallbackImages"
      }
    ],
    "mediaFiles": {
      "mediaFilesList": [],
      "mezzanine": ""
    },
    "skipOffset": "00:00:03",
    "startTime": "PT0S",
    "startTimeInSeconds": 0,
    "trackingEvents": [
      {
        "beaconUrls": [
          "https://example.com/view"
        ]
      }
    ],

```

```

        "duration": "PT10S",
        "durationInSeconds": 10,
        "eventId": "0",
        "eventProgramDateTime": null,
        "eventType": "impression",
        "startTime": "PT0S",
        "startTimeInSeconds": 0
      }
    ],
    "vastAdId": "123456"
  }
],
"availId": "0",
"availProgramDateTime": null,
"duration": "PT10S",
"durationInSeconds": 10,
"meta": null,
"nonLinearAdsList": [],
"startTime": "PT0S",
"startTimeInSeconds": 0
}
],
"dashAvailabilityStartTime": null,
"hlsAnchorMediaSequenceNumber": null,
"nextToken": "UFQxMFNFfMjAyMy0wNy0wNlQyMDo0MT0xNy45NDE4MDM0NDhaXzE%3D",
"nonLinearAvails": []
}

```

クライアント側のビーコン

クライアント側の追跡 `startTimeInSeconds` 要素を使用すると、MediaTailor を使用してビーコンのタイミングをサポートできます。

次の JSON レスポンスは、インプレッション、開始、四分位数、完了の主なビーコンタイプを示しています。

Note

インタラクティブ広告局 (IAB) のビデオインプレッション測定ガイドラインでは、インプレッションでは広告コンテンツをクライアント側でロードし、少なくともプレイヤーにレンダリングする時間を開始する必要があると述べています。詳細については、IAB ウェブサイトの「[デジタルビデオ広告配信テンプレート \(VAST\)](#)」を参照してください。


```
{
  "avails": [
    {
      "ads": [
        {
          "adId": "8104385",
          "duration": "PT15.100000078S",
          "durationInSeconds": 15.1,
          "startTime": "PT17.817798612S",
          "startTimeInSeconds": 17.817,
          "trackingEvents": [
            {
              "beaconUrls": [
                "http://exampleleadserver.com/tracking?event=impression"
              ],
              "duration": "PT15.100000078S",
              "durationInSeconds": 15.1,
              "eventId": "8104385",
              "eventType": "impression",
              "startTime": "PT17.817798612S",
              "startTimeInSeconds": 17.817
            },
            {
              "beaconUrls": [
                "http://exampleleadserver.com/tracking?event=start"
              ],
              "duration": "PT0S",
              "durationInSeconds": 0.0,
              "eventId": "8104385",
              "eventType": "start",
              "startTime": "PT17.817798612S",
              "startTimeInSeconds": 17.817
            },
            {
              "beaconUrls": [
                "http://exampleleadserver.com/tracking?event=firstQuartile"
              ],
              "duration": "PT0S",
              "durationInSeconds": 0.0,
              "eventId": "8104386",
              "eventType": "firstQuartile",
              "startTime": "PT21.592798631S",
              "startTimeInSeconds": 21.592
            }
          ]
        }
      ]
    }
  ]
}
```

```
    },
    {
      "beaconUrls": [
        "http://exampleleadserver.com/tracking?event=midpoint"
      ],
      "duration": "PT0S",
      "durationInSeconds": 0.0,
      "eventId": "8104387",
      "eventType": "midpoint",
      "startTime": "PT25.367798651S",
      "startTimeInSeconds": 25.367
    },
    {
      "beaconUrls": [
        "http://exampleleadserver.com/tracking?event=thirdQuartile"
      ],
      "duration": "PT0S",
      "durationInSeconds": 0.0,
      "eventId": "8104388",
      "eventType": "thirdQuartile",
      "startTime": "PT29.14279867S",
      "startTimeInSeconds": 29.142
    },
    {
      "beaconUrls": [
        "http://exampleleadserver.com/tracking?event=complete"
      ],
      "duration": "PT0S",
      "durationInSeconds": 0.0,
      "eventId": "8104390",
      "eventType": "complete",
      "startTime": "PT32.91779869S",
      "startTimeInSeconds": 32.917
    }
  ]
},
{
  "availId": "8104385",
  "duration": "PT15.100000078S",
  "durationInSeconds": 15.1,
  "startTime": "PT17.817798612S",
  "startTimeInSeconds": 17.817
}
]
```

```
}
```

サーバー側の広告ビーコンを使用したハイブリッドモード

MediaTailor は、セッション追跡用のハイブリッドモードをサポートしています。このモードでは、サービスは再生関連の広告追跡イベントを発行しますが、完全なクライアント側の追跡ペイロードをセッションで使用できるようにします。

再生プレフィックスを使用してハイブリッド追跡を有効にするには、プレイヤーから、プロトコルに従って、次のいずれかの形式でリクエストを使用して新しい MediaTailor 再生セッションを初期化します。

Example: HLS 形式

```
POST master.m3u8
{
  "adsParams": {
    "deviceType": "ipad"
  },
  "reportingMode": "server"
}
```

Example: DASH 形式

```
POST manifest.mpd
{
  "adsParams": {
    "deviceType": "ipad"
  },
  "reportingMode": "server"
}
```

MediaTailor は、次の追跡イベントをハイブリッドモードで維持します。

- インプレッション
- Start
- 最初の四分位数
- Midpoint
- 第 3 四分位数
- 完了

- `breakStart (vmap)`
- `breakEnd (vmap)`

クライアント側の広告追跡統合

このセクションでは、MediaTailor とさまざまなクライアント側の広告追跡サーバーの統合について説明します。

トピック

- [オープン測定 SDK](#)
- [Datazoom 無料プレイヤー SDKs](#)
- [Roku 広告フレームワーク \(RAF\)](#)
- [TheoPlayer](#)
- [MediaTailor SDK](#)

オープン測定 SDK

Interactive Advertising Bureau (IAB) Open Measurement SDK (OM SDK) は、ウェブビデオ環境とネイティブアプリ環境に配信される広告のサードパーティーの可視性と検証の測定を容易にします。

古い VAST バージョン 3 ドキュメントの場合、検証コードは拡張タイプの拡張ノードでロードする必要があります AdVerifications。拡張ノードのルートは、VAST 4.1 要素と同じスキーマを持つ AdVerifications ノードです。

OM SDK の導入を容易にするために、MediaTailor は Datazoom と提携して、オープン測定用に設定および検証された無料プレイヤー SDKs を提供しています。詳細については、「[Datazoom 無料プレイヤー SDKs](#)」を参照してください。

Note

MediaTailor は現在、VAST バージョン 3 のみをサポートしています。

Example: バージョン 4.1 より前の VAST 3 の検証ノード

```
...
<Extensions>
  <Extension type="AdVerifications">
```

```

    <AdVerifications>
      <Verification vendor="company.com-omid">
        <JavaScriptResource apiFramework="omid" browserOptional="true">
          <![CDATA[https://verification.com/omid_verification.js]]>
        </JavaScriptResource>
        <TrackingEvents>
          <Tracking event="verificationNotExecuted">
            <![CDATA[https://verification.com/trackingurl]]>
          </Tracking>
        </TrackingEvents>
        <VerificationParameters>
          <![CDATA[verification params key/value pairs]]>
        </VerificationParameters>
      </Verification>
    </AdVerifications>
  </Extension>
</Extensions>

```

MediaTailor は<Extensions>ノードからAdVerificationsデータを抽出し、クライアント側の追跡レスポンスのadVerifications配列に配置します。

Example: クライアント側の追跡レスポンスの adVerifications 配列

```

{
  "avails": [
    {
      "adBreakTrackingEvents": [],
      "adMarkerDuration": null,
      "ads": [
        {
          "adId": "3062770",
          "adParameters": "",
          "adProgramDateTime": "2023-08-23T16:25:40.914Z",
          "adSystem": "2.0",
          "adTitle": "AD-polarbear-15",
          "adVerifications": [
            {
              "executableResource": [],
              "javaScriptResource": [
                {
                  "apiFramework": "omid",
                  "browserOptional": "true",
                  "uri": "https://verification.com/omid_verification.js"
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}

```

[illegible]

```
        "eventId": "3062770",
        "eventProgramDateTime": null,
        "eventType": "impression",
        "startTime": "PT10.11S",
        "startTimeInSeconds": 10.11
      }
    ],
    "vastAdId": ""
  }
],
"availId": "3062770",
"availProgramDateTime": "2023-08-23T16:25:40.914Z",
"duration": "PT14.982S",
"durationInSeconds": 14.982,
"meta": null,
"nonLinearAdsList": [],
"startTime": "PT10.11S",
"startTimeInSeconds": 10.11
}
],
"dashAvailabilityStartTime": null,
"hlsAnchorMediaSequenceNumber": null,
"nextToken": "UFQxMC4xMVNfMjAyMy0wOC0yM1QxNjoyNjoyNC4yNDYxMDIxOTBaXzE%3D",
"nonLinearAvails": []
}
```

Note

IAB Tech Lab と連携して、コンプライアンスを確保するためにアプリケーションが毎年認定されるようにします。

OM SDK の詳細については、IAB Tech Lab ウェブサイトの「[Open Measurement SDK](#)」を参照してください。

Datavoom 無料プレイヤー SDKs

プレイヤー SDKs の導入を容易にするために、MediaTailor は Datavoom と提携し、[Google 広告マネージャーとのクライアント側の AWS Elemental MediaTailor 統合](#) および IAB Tech で設定およびテストされた無料プレイヤー SDKs を提供しています。[オープン測定 SDK](#)。

Datavoom プレイヤー SDK は、以下の機能をサポートしています。

- ライブプレイリストと VOD プレイリスト
- DASH および HLS の仕様
- Bitmovin、exoplayer、Android メディアプレイヤー、Apple AVPlayer、Brightcove、Chromecast Receiver、Dash.js、hls.js、JWPlayer、Shaka プレイヤー、THEO プレイヤー、Video.js、Roku などに対するプレイヤーベンダーのサポート
- 選択したデバイスで利用可能な IAB Tech Lab Open Measurement 認定
- クリックスルーイベント処理
- 広告カウントダウンタイマー、広告オーバーレイ、非線形イベント、広告ブレイク開始、広告ブレイク終了などの広告イベントディスパッチャー
- クライアント側の広告ビーコン
- オプションの設定としての Google Programmatic Access Library (PAL) SDK

Datazoom は、プレイヤー SDKs がサポートする有料の分析およびテレメトリサービスも提供します。お客様は、Datazoom マネジメントコンソールからプレイヤー SDK テレメトリをオプトインおよび制御できます。Datazoom プレイヤー SDKs にアクセスし、付加価値テレメトリおよび分析サービスの詳細については、[Datazoom サイトの](#)連絡先情報を使用します。

Roku 広告フレームワーク (RAF)

Roku 広告フレームワーク (RAF) は、Roku プラットフォーム全体で一貫した広告エクスペリエンスを維持します。動画広告を含むすべてのチャンネルは、RAF に関する Roku の認定要件を満たしている必要があります。特に、アプリは常に RAF を介したクライアント側のイベント射撃を使用する必要があります。MediaTailor は、サーバー側の広告挿入 (SSAI) プロバイダーとして、クライアント側のイベント射撃をサポートしています。RAFX SSAI Adapter は、SSAI マニフェストサーバー、またはステッカー、および RAF の両方にインターフェイスを提供します。これらのインターフェイスには以下が含まれます。

- masterURL レスポンスを解析し playURL、AdURL、および広告メタデータを抽出します。
- MediaTailor SSAI 広告メタデータを RAF が使用可能な広告メタデータに変換し、再生用に RAF を設定します。
- ストリームイベントと時間指定メタデータの監視。
- ストリームイベント、広告メタデータ、射撃イベントのピクセルを時間どおりに一致させます。
- MediaTailor SSAI マニフェストサーバーの要求に応じて AdURL に Ping/ポーリングし、RAF を解析して再設定します。

RAF 用の SSAI アダプターの詳細については、[Roku ウェブサイトの「Roku Adapters を使用したサーバー側の広告挿入の実装」](#)を参照してください。

TheoPlayer

MediaTailor との TheoPlayer 統合は、以下を実行します。

- VOD ワークフローとライブワークフローの両方で、HLS と DASH の MediaTailor クライアント側のイベント追跡をサポートする機能を提供します。
- リニア広告のみの追跡ビーコンの送信をサポートします。
- 広告中の の探索を無効にします。ただし、ユーザーが広告時間枠を過ぎようとしたときに広告を再生するロジックはありません。

TheoPlayer の SSAI の詳細、および MediaTailor のウェブ、Android、iOS、tvOS SDKs[MediaTailor](#)」を参照してください。TheoPlayer

MediaTailor SDK

AWS Elemental は、JavaScript ベースのソフトウェア開発キット (SDK) を維持します。AWS Elemental は SDK をそのまま提供しますが、黙示的な保証はありません。SDK をリファレンスデモとして使用して、MediaTailor を使用した へのオンボーディングを合理化します。SDK は、MediaTailor クライアント側の追跡 API を操作する方法を示しています。SDK は、HTML5-based プレイヤーのクライアント側の広告追跡とレポートを実装します。SDK は MediaTailor クライアント側のレポートセッションを初期化し、定期的に広告追跡情報をリクエストします。再生中、SDK は新しい広告イベントが検出されたときに広告追跡イベントを出力します。

MediaTailor SDK は、以下の機能をサポートしています。

- ライブプレイリストと VOD プレイリスト
- DASH および HLS の仕様
- クリックスルーイベント処理
- 広告イベントディスパッチャ
- カスタムイベントフック
- クライアント側の広告ビーコン。広告ビーコンの送信の詳細については、「」を参照してください [クライアント側のビーコン](#)。

Note

AWS サポートチケットを送信して、サンプルの JavaScript SDK for MediaTailor を受け取ります。パッケージとそのファイルのダウンロードリンクが届きます。

GetTracking を使用した広告ビーコンのページング

GetTracking エンドポイントを使用して、プレイヤーに返される広告の数を絞り込みます。例えば、マニフェストウィンドウが広く、長時間にわたる場合、返される広告ビーコンの数はプレイヤーのパフォーマンスに影響を与える可能性があります。

GetTracking は、返されたビーコンのリストをページングすることで、返されたビーコンの数を絞り込むために使用できるNextToken値を返します。NextToken 値を循環して、広告ビーコンの StartTimeInSecondsフィールドの目的の値を見つけることができます。

- への最初の呼び出しではGetTracking、マニフェストウィンドウに含まれる可能性のあるすべての広告が返されます。これには、それぞれの NextTokenおよび の値が含まれます。
- GetTracking リクエストに が含まれていない場合NextToken、マニフェストウィンドウ内のすべての広告が返されます。
- GetTracking リクエストに が含まれているNextTokenが、返す新しいビーコンがない場合、MediaTailor は元のリクエストで送信NextTokenした と同じ値を返します。
- 広告に対応するビーコンがなくなると、 はレスポンスから広告GetTrackingを削除します。
- からのトークンは 24 時間後にGetTracking期限切れになります。NextToken 値が 24 時間を超える場合、 の次の呼び出しは null 値 GetTrackingを返しますNextToken。

プレイヤーからの GetTracking の一般的な呼び出しシーケンス

クライアントプレイヤーからのGetTrackingリクエストは、トークンに関連する NextTokenおよび 広告とビーコンを含むリクエスト本文を含む POST です。

```
https://YouMediaTailorUrl/v1/tracking
{
    "NextToken": "value"
    .
    .
}
```

```
}
```

GetTracking で を使用する一般的な順序NextTokenは次のとおりです。

1. を最初に呼び出しますGetTracking。

すべての広告とビーコン、および後続の呼び出しNextTokenの最初の が返されます。

2. の値が null の場合、MediaTailor NextTokenはすべての広告ビーコンを返します。

3. の有効期限が切れている場合、MediaTailor NextTokenは HTTP リターンコード 400 エラーメッセージを返します。

に新しい呼び出しGetTrackingを行い、有効な を取得しますNextToken。

4. レスpons全体をスキャンして、目的の範囲内にある広告ビーコンStartTimeInSecondsの を見つけます。

5. 目的の NextTokenに関連付けられた の値GetTrackingを使用して、 に新しい呼び出しを行いますStartTimeInSeconds。

6. 必要に応じて、再生したい広告が見つかるまで、返された広告をもう一度繰り返します。

拡張例

この例では、GetTrackingの NextToken を使用して、プレイヤーに返される広告ビーコンの数を制限する方法を示します。

MediaTailor はGetTrackingリクエストを受け取ります。レスポンスには、ID 9935407 の広告とStartTimeInSeconds、値 52.286 秒と 48.332 秒の 2 つのビーコンが含まれています。

MediaTailor は、NextToken次のように で JSON レスポンスを送信します。

```
{
  "NextToken": "JF57ITe48t1441mv7TmLKuZLroxDzfIslp6BiSNL1IJmzPVMDN0lqrBYycgMbKEb
  "avails": [
    {
      "ads": [
        {
          "adId": "9935407",
          "adVerifications": [],
          "companionAds": [],
```

```
    "creativeId": "",
    "creativeSequence": "",
    "duration": "PT15S",
    "durationInSeconds": 15,
    "extensions": [],
    "mediaFiles": {
      "mediaFilesList": [],
      "mezzanine": ""
    },
    "startTime": "PT30S",
    "startTimeInSeconds": 45,
    "trackingEvents": [
      {
        "beaconUrls": [
          "http://adserver.com/tracking?event=Impression "
        ],
        "duration": "PT0S",
        "durationInSeconds": 0,
        "eventId": "9935414",
        "eventType": "secondQuartile",
        "startTime": "PT52.286S",
        "startTimeInSeconds": 52.286
      },
      {
        "beaconUrls": [
          "http://adserver.com/tracking?event=firstQuartile"
        ],
        "duration": "PT0S",
        "durationInSeconds": 0,
        "eventId": "9935412",
        "eventType": "firstQuartile",
        "startTime": "PT48.332S",
        "startTimeInSeconds": 48.332
      }
    ],
    "vastAdId": ""
  }
],
"startTime": "PT46.47S",
"startTimeInSeconds": 46.47
}
]
```

次のGetTrackingリクエストでは、MediaTailor は JF57ITe48t1441mv7TmLKuZLroxDzflslp6BiSNL1IJmzPVMDN0lqrBYycgMbKEb NextTokenの値で 応答します。

MediaTailor は、前の呼び出しNextTokenの で設定された StartTimeInSeconds に一致する広告 とビーコンで応答します。

ID 9235407 の以前の広告に加えて、ID 9935407 の別の広告がレスポンスに含まれているとします。 広告 ID 9235407 のビーコンには、StartTimeInSecondss 132.41 と 70.339 があります。

MediaTailor はセッション内のすべてのビーコンを繰り返して、ID 9235407 の広告からビーコン 3 とビーコン 4 である 52.286 秒StartTimeInSecondsを超えるビーコンを選択します。

```
{
  "NextToken": ZkfknvbfsgdfbsDFRdffg12EdffecFRvhjyjfhdfhnjtsg5SDGN
  "avails": [
    {
      "ads": [
        {
          "adId": "9235407",
          "adVerifications": [],
          "companionAds": [],
          "creativeId": "",
          "creativeSequence": "",
          "duration": "PT15.816S",
          "durationInSeconds": 19.716,
          "extensions": [],
          "mediaFiles": {
            "mediaFilesList": [],
            "mezzanine": ""
          },
          "startTime": "PT2M0S",
          "StartTimeInSeconds": 120.0,
          "trackingEvents": [
            {
              "beaconUrls": [
                "http://adserver.com/tracking?event=complete"
              ],
              "duration": "PT0S",
              "durationInSeconds": 0,
```

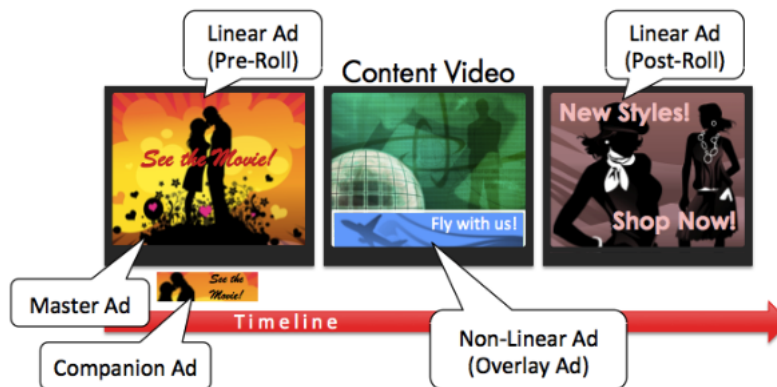
```
    "eventId": "8935414",
    "eventType": "firstQuartile",
    "startTime": "PT1M10.330S",
    "StartTimeInSeconds": 70.339
  },
  {
    "beaconUrls": [
      "http://adserver.com/tracking?event=thirdQuartile"
    ],
    "duration": "PT0S",
    "durationInSeconds": 0,
    "eventId": "8935412",
    "eventType": "secondQuartile",
    "startTime": "PT2M12.41S",
    "StartTimeInSeconds": 132.41
  }
],
"vastAdId": ""
},
],
"startTime": "PT36.47S",
"StartTimeInSeconds": 36.47
}
]
```

オーバーレイ広告

ミッドロール広告の表示エクスペリエンスを中断せずに収益化を増やすライブストリーミングワークフローでは、現在の AWS Elemental MediaTailor 統合を活用して、クライアント側でレンダリングされる広告形式をガイドできます。このタイプの広告はオーバーレイ広告と呼ばれます。オーバーレイ広告は、「L バンド広告」、「非線形動画広告」、picture-in-picture 広告、「モーションオーバーレイ」、「インコンテンツ広告」、「フレーム広告」の形式で表示される非線形動画広告です。

MediaTailor は、オーバーレイ広告挿入の機会の帯域内シグナル id=0x38 としてセグメンテーションタイプを持つ SCTE-35 マーカーを検出します。SCTE-35 マーカーにより、MediaTailor は広告決定サーバー (ADS) にリクエストを送信し、ADS は VAST レスポンスで非線形広告ペイロードで応答します。MediaTailor は、オーバーレイ広告挿入をサポートするために VAST レスポンスを解析します。MediaTailor はリニア広告のステッチングを実行するのではなく、再生可能な非リニアオーバーレイ広告があることをプレイヤーに通知します。このシグナリングにより、プレイヤーはクライアン

ト側の追跡エンドポイントから再生する非線形広告を取得して関連付けることができます。その後、プレイヤーは表示、レポート、およびそれらの広告に関連するその他のタスクを処理します。たとえば、プレイヤーの開発者は、オーバーレイ広告形式をサポートするベンダーのデバイス SDK を使用できます。クライアント側の追跡統合の詳細については、「」を参照してください[クライアント側の広告追跡統合](#)。



トピック

- [MediaTailor でオーバーレイ広告を使用するための前提条件](#)
- [MediaTailor でのオーバーレイ広告の使用開始](#)
- [MediaTailor でのオーバーレイ広告のログ記録とメトリクス](#)
- [MediaTailor でのオーバーレイ広告の請求](#)

MediaTailor でオーバーレイ広告を使用するための前提条件

MediaTailor でオーバーレイ広告を使用する場合、次の前提条件が適用されます。

- ワークフローは、ビデオオンデマンド (VOD) ではなく、ライブである必要があります。
- 広告決定サーバー (ADS) レスポンスは、VAST レスポンスで非線形広告のみを返すように設定する必要があります。MediaTailor は、広告ステッチングの目的でリニア広告を無視します。
- マニフェストは、オーバーレイアド機能呼び出す `id=0x38` ために、セグメンテーションタイプで SCTE-35 タイムシグナルメッセージを使用する必要があります。
- ストリーミングプロバイダーは、クライアント/デバイスアプリケーションを制御でき、MediaTailor クライアント側の追跡 API と統合されている必要があります。

MediaTailor でのオーバーレイ広告の使用開始

このセクションでは、MediaTailor の overlay-ads 機能の使用を開始する方法について説明します。SCTE-35 シグナリングの設定、広告決定サーバー (ADS) レスポンスの設定、セッションレベルの制御の設定を行います。

トピック

- [オーバーレイ広告の有効化](#)
- [クライアント側のメタデータを使用したオーバーレイ広告の追跡](#)

オーバーレイ広告の有効化

オーバーレイ広告の MediaTailor サポートはデフォルトで有効になっています。マニフェスト内の特定の SCTE-35 広告マーカータイプは、オーバーレイ広告の挿入をトリガーします。一部のプレイヤーはオーバーレイ広告のクライアント側のレンダリングをサポートしていない可能性があるため、セッションレベルでこの機能を無効にすることができます。

HLS または DASH 再生プレフィックスを使用して overlay-ad サポートを無効にするには:

- プレイヤーから、プロトコルに従った以下のいずれかの形式を使用して、新しい MediaTailor 再生セッションを開始します。
 - 例: HLS 形式

```
GET mediatailorURL/v1/master/hashed-account-id/origin-id/asset-id?  
aws.overlayAvails=off
```

- 例: DASH 形式

```
GET mediatailorURL/v1/master/hashed-account-id/origin-id/asset-id?  
aws.overlayAvails=off
```

session-initialization プレフィックスを使用して overlay-ad サポートを無効にするには:

- プレイヤーで、MediaTailor へのセッション初期化リクエストの JSON メッセージ本文を作成します。

- 広告オーバーレイのサポートを無効にするには、オブジェクトを の値を持つ最上位キーoverlaysとして追加しますoff。overlays デフォルト値は ですon。
- (オプション) MediaTailor がadsParamsオブジェクト内の ADS に渡すパラメータを指定します。これらのパラメータは、MediaTailor の ADS テンプレート URL にある [player_params.param] 設定に対応しています。

Example HLS:

```
POST master.m3u8
{
  "adsParams": {
    "deviceType": "ipad"
  },
  "overlayAvails": "off"
}
```

Example DASH:

```
POST manifest.mpd
{
  "adsParams": {
    "deviceType": "androidmobile"
  },
  "overlayAvails": "off"
}
```

マニフェストシグナリング

MediaTailor トリガーオーバーレイ広告は、マニフェストに特定の SCTE-35 マーカーが表示されるとサポートされます。必要なシグナルは、スプライスコマンドタイプ 6、またはプロバイダーオーバーレイ広告開始シグナルであるタイムシグナルです。このシグナルのセグメンテーションタイプ ID は です 0x38

次の例は、JSON オブジェクトの 0x38 SCTE-35 マーカーを示しています。

```
{
  "tableId": 252,
  "selectionSyntaxIndicator": false,
  "privateIndicator": false,
  "sectionLength": 53,
```

```
"protocolVersion": 0,
"encryptedPacket": false,
"encryptedAlgorithm": 0,
"ptsAdjustment": 0,
"cwIndex": 0,
"tier": 4095,
"spliceCommandLength": 5,
"spliceCommandType": 6,
"spliceCommand": {
  "specified": true,
  "pts": 1800392
},
"descriptorLoopLength": 31,
"descriptors": [
  {
    "spliceDescriptorTag": 2,
    "descriptorLength": 29,
    "identifier": "CUEI",
    "segmentationEventId": 158389361,
    "segmentationEventCancelIndicator": false,
    "programSegmentationFlag": true,
    "segmentationDurationFlag": true,
    "deliveryNotRestrictedFlag": false,
    "webDeliveryAllowedFlag": true,
    "noRegionalBlackoutFlag": true,
    "archiveAllowedFlag": true,
    "deviceRestrictions": 3,
    "segmentationDuration": 1350000,
    "segmentationUpidType": 9,
    "segmentationUpidLength": 7,
    "segmentationUpid": {
      "0": 111,
      "1": 118,
      "2": 101,
      "3": 114,
      "4": 108,
      "5": 97,
      "6": 121
    },
    "segmentationTypeId": 56,
    "segmentNum": 1,
    "segmentsExpected": 0
  }
],
```

```
"crc": 2510422713
}
```

次の例は、バイナリ (ベース 32/16 進数) 値として表される SCTE-35 シグナルを示しています。

```
0xfc303500000000000000ffff00506fe001b78c8001f021d435545490970d4717fd0000dbba009076f7665726c617
```

次の例は、HLS マニフェストと DASH マニフェストの両方の SCTE-35 マーカーを示しています。

Example: HLS マニフェスト

```
#EXTM3U
#EXT-X-VERSION:6
#EXT-X-TARGETDURATION:7
#EXT-X-MEDIA-SEQUENCE:419
#EXT-X-DISCONTINUITY-SEQUENCE:3
#EXT-X-PROGRAM-DATE-TIME:2023-08-15T04:30:09.231Z
#EXTINF:6.02,
https://aws.cloudfront.net/media/asset1/index1_00007.ts
#EXT-X-DISCONTINUITY
#EXT-X-KEY:METHOD=NONE
#EXT-X-PROGRAM-DATE-TIME:2023-08-15T04:30:15.251Z
#EXTINF:6.0,
https://aws.cloudfront.net/media/asset1/index1_00001.ts
#EXT-X-PROGRAM-DATE-TIME:2023-08-15T04:30:21.251Z
#EXTINF:4.0,
https://aws.cloudfront.net/media/asset1/index1_00002.ts
#EXT-X-DISCONTINUITY
#EXT-X-DATERANGE:ID="1692073825251-30-1",START-
DATE="2023-08-15T04:30:25.251Z",DURATION=10.0,PLANNED-DURATION=10.0,SCTE35-
OUT=0xfc303500000000000000ffff00506fe001b78c8001f021d435545490970d4717fdf00000dbba009076f7665726
#EXT-X-PROGRAM-DATE-TIME:2023-08-15T04:30:25.251Z
#EXTINF:2.0,
https://aws.cloudfront.net/media/asset1/index1_00003.ts
#EXT-X-PROGRAM-DATE-TIME:2023-08-15T04:30:27.251Z
#EXTINF:6.0,
https://aws.cloudfront.net/media/asset1/index1_00004.ts
#EXT-X-PROGRAM-DATE-TIME:2023-08-15T04:30:33.251Z
#EXTINF:2.0,
https://aws.cloudfront.net/media/asset1/index1_00005.ts
#EXT-X-DISCONTINUITY
#EXT-X-PROGRAM-DATE-TIME:2023-08-15T04:30:35.251Z
#EXTINF:4.0,
```

```
https://aws.cloudfront.net/media/asset1/index1_00006.ts
#EXT-X-PROGRAM-DATE-TIME:2023-08-15T04:30:39.251Z
#EXTINF:6.02,
https://aws.cloudfront.net/media/asset1/index1_00007.ts
```

Example: DASH マニフェスト

```
<?xml version="1.0"?>
<MPD xmlns="urn:mpeg:dash:schema:mpd:2011" xmlns:scte35="urn:scte:scte35:2013:xml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  availabilityStartTime="2023-08-15T16:34:05.911Z" minBufferTime="PT30S"
  minimumUpdatePeriod="PT2S" profiles="urn:mpeg:dash:profile:isoff-live:2011"
  publishTime="2023-08-15T16:34:17.950Z" suggestedPresentationDelay="PT20S"
  timeShiftBufferDepth="PT1M30S" type="dynamic"
  xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 http://standards.iso.org/ittf/
PubliclyAvailableStandards/MPEG-DASH_schema_files/DASH-MPD.xsd">
  <Period xmlns="urn:mpeg:dash:schema:mpd:2011" id="1692117245944_1" start="PT0.033S">
    <BaseURL>https://aws.cloudfront.net/out/v1/abc/123/def/</BaseURL>
    <EventStream schemeIdUri="urn:scte:scte35:2013:xml" timescale="90000">
      <Event duration="900000">
        <scte35:SpliceInfoSection protocolVersion="0" ptsAdjustment="0" tier="4095">
          <scte35:TimeSignal>
            <scte35:SpliceTime ptsTime="0"/>
          </scte35:TimeSignal>
          <scte35:SegmentationDescriptor segmentNum="0" segmentationDuration="900000"
segmentationEventCancelIndicator="false" segmentationEventId="1"
segmentationTypeId="56" segmentsExpected="0" subSegmentNum="0"
subSegmentsExpected="0">
            <scte35:SegmentationUpid segmentationUpidFormat="hexBinary"
segmentationUpidType="14">63736f7665726c6179</scte35:SegmentationUpid>
          </scte35:SegmentationDescriptor>
        </scte35:SpliceInfoSection>
      </Event>
    </EventStream>
    <AdaptationSet bitstreamSwitching="true" mimeType="video/mp4"
segmentAlignment="true" startWithSAP="1" subsegmentAlignment="true"
subsegmentStartsWithSAP="1">
      <Representation bandwidth="3000000" codecs="avc1.4D4028" frameRate="30/1"
height="1080" id="1" width="1920">
        <SegmentTemplate initialization="../
cf684d31ec9e451ca98d2349989f6c0a/855c733eed20493ab3cc1100750bcf0b/
index_video_1_0_init.mp4" media="../
```

```

cf684d31ec9e451ca98d2349989f6c0a/855c733eed20493ab3cc1100750bcf0b/index_video_1_0_
$Number$.mp4" presentationTimeOffset="0" startNumber="1" timescale="30000">
    <SegmentTimeline>
        <S d="60000" r="6" t="1000"/>
        <S d="30000" t="421000"/>
    </SegmentTimeline>
</SegmentTemplate>
</Representation>
<Representation bandwidth="2499968" codecs="avc1.4D4028" frameRate="30/1"
height="1080" id="2" width="1920">
    <SegmentTemplate initialization="../
cf684d31ec9e451ca98d2349989f6c0a/855c733eed20493ab3cc1100750bcf0b/
index_video_2_0_init.mp4" media="../
cf684d31ec9e451ca98d2349989f6c0a/855c733eed20493ab3cc1100750bcf0b/index_video_2_0_
$Number$.mp4" presentationTimeOffset="0" startNumber="1" timescale="30000">
    <SegmentTimeline>
        <S d="60000" r="6" t="1000"/>
        <S d="30000" t="421000"/>
    </SegmentTimeline>
</SegmentTemplate>
</Representation>
<Representation bandwidth="2200000" codecs="avc1.4D401F" frameRate="30/1"
height="720" id="3" width="1280">
    <SegmentTemplate initialization="../
cf684d31ec9e451ca98d2349989f6c0a/855c733eed20493ab3cc1100750bcf0b/
index_video_3_0_init.mp4" media="../
cf684d31ec9e451ca98d2349989f6c0a/855c733eed20493ab3cc1100750bcf0b/index_video_3_0_
$Number$.mp4" presentationTimeOffset="0" startNumber="1" timescale="30000">
    <SegmentTimeline>
        <S d="60000" r="6" t="1000"/>
        <S d="30000" t="421000"/>
    </SegmentTimeline>
</SegmentTemplate>
</Representation>
</AdaptationSet>
<AdaptationSet lang="eng" mimeType="audio/mp4" segmentAlignment="0">
    <Label>Alternate Audio</Label>
    <Representation audioSamplingRate="48000" bandwidth="128000" codecs="mp4a.40.2"
id="9">
        <AudioChannelConfiguration
schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
        <SegmentTemplate initialization="../
cf684d31ec9e451ca98d2349989f6c0a/855c733eed20493ab3cc1100750bcf0b/
index_audio_9_0_init.mp4" media="../

```

```
cf684d31ec9e451ca98d2349989f6c0a/855c733eed20493ab3cc1100750bcf0b/index_audio_9_0_
$Number$.mp4" presentationTimeOffset="0" startNumber="1" timescale="48000">
  <SegmentTimeline>
    <S d="98304" t="0"/>
    <S d="96256" t="98304"/>
    <S d="95232" t="194560"/>
    <S d="96256" r="2" t="289792"/>
    <S d="95232" t="578560"/>
    <S d="46080" t="673792"/>
  </SegmentTimeline>
</SegmentTemplate>
</Representation>
</AdaptationSet>
</Period>
</MPD>
```

広告決定サーバー (ADS) レスポンス

ADS レスポンスには、有効な追跡イベントが 1 つ含まれている必要があります。少なくとも、追跡イベントは Impression 追跡イベントにすることができます。追跡イベントには、少なくとも 1 つの NonLinear 広告が含まれている必要があります。この広告は、静的、HTML、または iFrame リソースの形式でオーバーレイ広告です。

```
<vmap AdBreak breaktype="linear" breakId="csoverlay"
```

VAST レスポンスが breakType の VMAP である場合 nonlinear、avail メタデータは nonLinearAvails ルートオブジェクト内にあります。VAST レスポンスが breakType の VMAP である場合 linear、または VMAP のないプレーン VAST レスポンスである場合、avail メタデータは avails ルートオブジェクト内にあります。

次の VAST レスポンスは、breakType 値が のラップされた VMAP レスポンスです linear。

ラップされた VMAP レスポンスに加えて、MediaTailor は breakType の値を持つラップされた VMAP レスポンス nonlinear とプレーン VAST レスポンスもサポートします。

```
<?xml version="1.0" encoding="utf-8"?>
<vmap:VMAP xmlns:vmap="http://www.iab.net/vmap-1.0" version="1.0">
  <vmap:AdBreak breakType="linear" breakId="csoverlay">
    <vmap:AdSource allowMultipleAds="true" followRedirects="true" id="1">
      <vmap:VASTAdData>
        <VAST xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="3.0"
          xsi:noNamespaceSchemaLocation="vast.xsd">
```

```

    <Ad sequence="1">
      <InLine>
        <AdSystem>2.0</AdSystem>
        <AdTitle>2</AdTitle>
        <Impression><![CDATA[https://adserver.com/beacon=impression]]></
Impression>
        <Creatives>
          <Creative>
            <NonLinearAds>
              <NonLinear width="640" height="360" id="18">
                <StaticResource creativeType="text/js_ref"><![CDATA[https://
client-side-ads.com/tags/static/ctv-generic/overlay001.json?iv\_geo\_country%3DUS%26]]></
StaticResource>
              </NonLinear>
            </NonLinearAds>
          </Creative>
        </Creatives>
      </InLine>
    </Ad>
  </VAST>
</vmap:VASTAdData>
</vmap:AdSource>
<vmap:TrackingEvents>
  <vmap:Tracking event="breakStart"><![CDATA[https://adserver.com/
beacon=breakstartimpression]]></vmap:Tracking>
  <vmap:Tracking event="breakEnd"><![CDATA[https://adserver.com/
beacon=breakendimpression]]></vmap:Tracking>
</vmap:TrackingEvents>
</vmap:AdBreak>
</vmap:VMAP>

```

Example 1: MediaTailor への DASH マニフェストソース

```

<?xml version="1.0" encoding="utf-8"?>
<MPD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="urn:mpeg:dash:schema:mpd:2011" xmlns:scte35="urn:scte:scte35:2013:xml"
  xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 http://standards.iso.org/
  ittf/PubliclyAvailableStandards/MPEG-DASH_schema_files/DASH-MPD.xsd" id="201"
  type="dynamic" publishTime="2022-11-07T19:59:05+00:00" minimumUpdatePeriod="PT2S"
  availabilityStartTime="2022-11-07T06:57:11.250000+00:00" minBufferTime="PT10S"
  suggestedPresentationDelay="PT20.000S" timeShiftBufferDepth="PT58.999S"
  profiles="urn:mpeg:dash:profile:isoff-live:2011">
  <Period start="PT46827.601S" id="0" duration="PT88.321S">

```

```

...
</Period>
<Period start="PT46915.922S" id="45" duration="PT6.006S">
  <EventStream timescale="90000" schemeIdUri="urn:scte:scte35:2014:xml+bin">
    <Event duration="540000" id="144">
      <scte35:Signal>
        <scte35:Binary>SCTE35-binary</scte35:Binary>
      </scte35:Signal>
    </Event>
  </EventStream>
  ...
</Period>
<Period start="PT46921.928S" id="49">
  ...
</Period>
</MPD>

```

Example 2: 広告 ID デコレーションを含む MediaTailor パーソナライズされた DASH マニフェスト

```

<?xml version="1.0" encoding="utf-8"?>
<MPD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="urn:mpeg:dash:schema:mpd:2011" xmlns:scte35="urn:scte:scte35:2013:xml"
  xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 http://standards.iso.org/
  ittf/PubliclyAvailableStandards/MPEG-DASH_schema_files/DASH-MPD.xsd" id="201"
  type="dynamic" publishTime="2022-11-07T19:59:05+00:00" minimumUpdatePeriod="PT2S"
  availabilityStartTime="2022-11-07T06:57:11.250000+00:00" minBufferTime="PT10S"
  suggestedPresentationDelay="PT20.000S" timeShiftBufferDepth="PT58.999S"
  profiles="urn:mpeg:dash:profile:isoff-live:2011">
  <Period start="PT46827.601S" id="0" duration="PT88.321S">
    ...
  </Period>
  <Period start="PT46915.922S" id="45" duration="PT6.006S">
    <EventStream schemeIdUri="urn:sva:advertising-wg:ad-id-signaling" timescale="90000">
      <Event presentationTime="13500000" duration="1351350">
        <![CDATA[{"version": 1,"identifiers": [{"scheme":
"urn:smpte:ul:060E2B34.01040101.01200900.00000000","value": "adId","ad_position":
"adId", "ad_type":"overlay","creative_id": "creativeId","tracking_uri":
"trackingUri"}]]]></Event>
      </EventStream>
      ...
    </Period>
    <Period start="PT46921.928S" id="49">
      ...
    </Period>
  </MPD>

```



```
</Period>
</MPD>
```

クライアント側のメタデータを使用したオーバーレイ広告の追跡

MediaTailor は、オーバーレイ広告を表示 `nonLinearAdsList` の に配置します。MediaTailor クライアント側の追跡 API には、`avails` と という 2 つのルートオブジェクトがあります `nonLinearAvails`。VAST レスポンスが `breakType` の VMAP である場合 `nonlinear`、`avail` メタデータは `nonLinearAvails` ルートオブジェクト内にあります。VAST レスポンスが の VMAP `breakType` である場合 `linear`、または VMAP を使用しないプレーン VAST レスポンスである場合、`avail` メタデータは `avails` ルートオブジェクト内にあります。

クライアント側の追跡の詳細については、「」を参照してください [クライアント側の広告追跡](#)。

次の例は、`breakType` 値が のプレーン VAST レスポンスまたは VMAP レスポンスを示しています `linear`。

```
{
  "avails": [
    {
      "adBreakTrackingEvents": [
        {
          "beaconUrls": [
            "https://adserver.com/beacon=breakstartimpression"
          ],
          "eventType": "breakStart"
        },
        {
          "beaconUrls": [
            "https://adserver.com/beacon=breakendimpression"
          ],
          "eventType": "breakEnd"
        }
      ],
      "adMarkerDuration": null,
      "ads": [],
      "availId": "828",
      "availProgramDateTime": null,
      "duration": "PT0S",
      "durationInSeconds": 0,
      "meta": null,
      "nonLinearAdsList": [
```

```
{
  "extensions": null,
  "nonLinearAdList": [
    {
      "adId": "",
      "adParameters": null,
      "adSystem": "2.0",
      "adTitle": "2",
      "apiFramework": null,
      "clickThrough": null,
      "clickTracking": null,
      "clickTrackingId": null,
      "creativeAdId": "",
      "creativeId": "18",
      "creativeSequence": "",
      "duration": null,
      "durationInSeconds": 0,
      "expandedHeight": null,
      "expandedWidth": null,
      "height": "360",
      "htmlResource": null,
      "iFrameResource": null,
      "maintainAspectRatio": false,
      "minSuggestedDuration": null,
      "scalable": false,
      "staticResource": "https://client-side-ads.com/tags/static/ctv-generic/overlay001.json?iv_geo_country%3DUS%26",
      "staticResourceCreativeType": "text/js_ref",
      "width": "640"
    }
  ],
  "trackingEvents": [
    {
      "beaconUrls": [
        "https://adserver.com/beacon=impression"
      ],
      "duration": null,
      "durationInSeconds": 0,
      "eventId": null,
      "eventProgramDateTime": null,
      "eventType": "impression",
      "startTime": null,
      "startTimeInSeconds": 0
    }
  ]
}
```

```

    ]
  }
],
"startTime": "PT1M46.08S",
"startTimeInSeconds": 106.08
}
],
"dashAvailabilityStartTime": null,
"hlsAnchorMediaSequenceNumber": null,
"nextToken": null,
"nonLinearAvails": []
}

```

次の例は、breakType値が のプレーン VMAP レスポンスを示していますnonlinear。

```

{
  "avails": [],
  "dashAvailabilityStartTime": null,
  "hlsAnchorMediaSequenceNumber": null,
  "nextToken": null,
  "nonLinearAvails": [
    {
      "adBreakTrackingEvents": [
        {
          "beaconUrls": [
            "https://adserver.com/beacon=breakstartimpression"
          ],
          "eventType": "breakStart"
        },
        {
          "beaconUrls": [
            "https://adserver.com/beacon=breakendimpression"
          ],
          "eventType": "breakEnd"
        }
      ],
      "adMarkerDuration": null,
      "ads": [],
      "availId": "828",
      "availProgramDateTime": null,
      "duration": "PT0S",
      "durationInSeconds": 0,
      "meta": null,

```

```
"nonLinearAdsList": [
  {
    "extensions": null,
    "nonLinearAdList": [
      {
        "adId": "",
        "adParameters": null,
        "adSystem": "2.0",
        "adTitle": "2",
        "apiFramework": null,
        "clickThrough": null,
        "clickTracking": null,
        "clickTrackingId": null,
        "creativeAdId": "",
        "creativeId": "18",
        "creativeSequence": "",
        "duration": null,
        "durationInSeconds": 0,
        "expandedHeight": null,
        "expandedWidth": null,
        "height": "360",
        "htmlResource": null,
        "iFrameResource": null,
        "maintainAspectRatio": false,
        "minSuggestedDuration": null,
        "scalable": false,
        "staticResource": "https://client-side-ads.com/tags/static/ctv-generic/overlay001.json?iv_geo_country%3DUS%26",
        "staticResourceCreativeType": "text/js_ref",
        "width": "640"
      }
    ],
    "trackingEvents": [
      {
        "beaconUrls": [
          "https://adserver.com/beacon=impression"
        ],
        "duration": null,
        "durationInSeconds": 0,
        "eventId": null,
        "eventProgramDateTime": null,
        "eventType": "impression",
        "startTime": null,
        "startTimeInSeconds": 0
      }
    ]
  }
]
```

```
    }
  ]
}
],
"startTime": "PT1M46.08S",
"startTimeInSeconds": 106.08
}
]
}
```

MediaTailor でのオーバーレイ広告のログ記録とメトリクス

このセクションでは、MediaTailor のオーバーレイ広告のログ記録とメトリクスについて説明します。ログ記録の設定の詳細については、「」を参照してください[AWS Elemental MediaTailor リソースのモニタリングとタグ付け](#)。

トピック

- [CloudWatch ログ](#)
- [CloudWatch メトリクス](#)

CloudWatch ログ

CloudWatch は、オーバーレイ広告に関する次のログ情報を収集します。

- VAST_RESPONSE - 非線形広告リストに関する情報を表示します。
- FILLED_PROVIDER_OVERLAY - 非線形広告に関する情報を表示します。

Note

RAW_ADS_RESPONSE は、ADS からの元のレスポンスを示すオプションのイベントです。このイベントの使用は、ステージングおよびテスト環境で特に役立ちます。設定またはアカウントでこのイベントを有効にするには、AWS サポートにチケットを送信します。

CloudWatch メトリクス

MediaTailor は、オーバーレイ広告メトリクスを他の ADS メトリクスとは別に収集します。MediaTailor は、ADS から広告を正常に取得した後、これらのメトリクスを収集します。API をポーリング `GetTracking` してメトリクスを収集する必要はありません。

次の表に、オーバーレイ広告の CloudWatch メトリクスを示します。

メトリクス	説明
<code>AdDecisionServer.OverlayAds</code>	指定した CloudWatch 期間内に ADS レスポンスに含まれるオーバーレイ広告の数。
<code>AdDecisionServer.OverlayErrors</code>	指定した CloudWatch 期間内に MediaTailor が ADS から受信した HTTP 以外の 200 ステータスコードレスポンス、空のレスポンス、タイムアウトレスポンスの数。
<code>AdDecisionServer.OverlayFilled</code>	少なくとも 1 つのオーバーレイ広告で正常に埋められた表示の数: • 1 - 有効な広告が少なくとも 1 つあります。 • 0 - MediaTailor がオーバーレイ広告を取得できなかったか、その他の障害が発生しました。 <code>SampleCount</code> は、入力された <code>avail</code> の数を追跡します。 <code>Sum</code> は、正常に埋められたオーバーレイ表示の数を追跡します。
<code>AdDecisionServer.OverlayMinSuggestedDuration</code>	指定した CloudWatch 期間内に MediaTailor が ADS から受信したすべての広告 <code>minSuggestedDuration</code> の期間の合計をミリ秒単位で表します。が指定され <code>minSuggestedDuration</code> がない場合、表示される期間は計画された期間です。

メトリクス	説明
AdDecisionServer.OverlayLatency	MediaTailor が ADS に対して行うリクエストの応答時間をミリ秒単位で表します。
AdDecisionServer.OverlayTimeouts	ユーザー指定の CloudWatch 期間中の ADS に対するタイムアウトリクエストの数。
AdsBilled	請求される広告の詳細については、「」を参照してください MediaTailor でのオーバーレイ広告の請求 。
Avail.*	MediaTailor はオーバーレイ広告の計画を行わないため、CloudWatch は Avail.X メトリクスを表示しません。
SkippedReason.*	MediaTailor はオーバーレイ広告の計画を行わないため、CloudWatch は SkippedReason.X メトリクスを表示しません。

MediaTailor でのオーバーレイ広告の請求

MediaTailor は、ADS レスポンスの非線形広告の数に基づいて顧客に請求します。この数には、ブレイク期間を超えて延長される非線形広告が含まれます。MediaTailor が表示を埋めると、その表示を埋めた広告に対して請求されます。

プリフェッチワークフローの場合、MediaTailor はプリフェッチの取得時に広告に対して課金せず、そのセッションの消費ウィンドウで互換性のある広告表示が表示される場合に課金します。

その他の請求情報については、「」を参照してください<https://aws.amazon.com/mediatailor/pricing/>。

広告 ID デコレーション

AWS Elemental MediaTailor は、コンテンツから広告ブレイクに移行するときにサーバー側の広告ステッチングを実行します。MediaTailor は、ステッチされた広告に関連付けられたメタデータでマニフェストを条件付けできます。これにより、次の利点が得られます。

⚠ Important

広告 ID デコレーションはサーバーガイド広告挿入 (SGAI) メソッドと互換性がありません。X-AD-CREATIVE-SIGNALING ヘッダーを入力するフィールドは、キャッシュ可能なマニフェストが最初書き込まれたときではなく、広告の再生中にアセットリストがフェッチされたときにのみ認識されるためです。

- ビデオ開始時間 (VST) の向上
- MediaTailor は、サーバー側の広告挿入とサーバーガイド付き広告挿入のハイブリッドモデルをサポートできます
- サーバー側のセッションは、広告位置マーカーを使用して再生タイムラインを構築できます
- MediaTailor API を使用して再生タイムラインを既に構築しているクライアント側セッションの場合、セッション VST は改善されます。これは、セッションが追跡 API を呼び出してタイムラインを構築することに依存しないためです。
- MediaTailor は、サーバー側の広告挿入やクライアント側のレンダリングされた広告をシーン内に表示するために利用できます。これにより、プレイヤーのソフトウェア開発キット (SDK) に、クライアント側の広告のために広告配信エンティティを直接呼び出すための個別の統合は必要ありません。MediaTailor は、マニフェストとクライアント側の追跡 API を通じて広告を提供できます。

各クリエイティブ広告アセットを一意的識別子に関連付けるための標準があります。この関連付けにより、広告主、エージェンシー、ベンダー、パブリッシャーは、クリエイティブ広告アセットを独立したワークフローに関連付けることができます。ストリームのメトリクスとモニタリングが継続的に改善され、より多くのディストリビューターがサーバーベースの挿入アーキテクチャを利用するにつれて、パーソナライズされたマニフェスト内など、インターリーブ/ステッチされたプレゼンテーション内で個々のクリエイティブアセットに割り当てられた識別子を正確に伝達する必要が生じます。

トピック

- [セッションの広告 ID シグナリングの有効化](#)
- [マニフェストと広告メタデータの挿入](#)
- [広告決定サーバー \(ADS\) インタラクション](#)
- [クライアント側の追跡 API](#)

セッションの広告 ID シグナリングの有効化

セッションの初期化中に、広告 ID シグナリング機能を有効にする必要があります。この機能を有効にするプロセスは、HLS/DASH 再生プレフィックス (暗黙的なセッションの初期化) を使用したセッションの作成と、セッションの初期化プレフィックス (明示的なセッションの初期化) とは異なります。

HLS/DASH 再生プレフィックスを使用してセッションの広告 ID を有効にするには

- プレイヤーから、プロトコルに従った以下のいずれかの形式を使用して、新しい MediaTailor 再生セッションを開始します。

- 例: HLS 形式

```
GET <mediatailorURL>/v1/master/<hashed-account-id>/<origin-id>/<asset-id>?
aws.adSignalingEnabled=true
```

- 例: DASH 形式

```
GET <mediatailorURL>/v1/dash/<hashed-account-id>/<origin-id>/<asset-id>?
aws.adSignalingEnabled=true
```

セッション初期化プレフィックスを使用してセッションの広告 ID を有効にするには

- プレイヤーで、MediaTailor へのセッション初期化リクエストの JSON メッセージ本文を作成します。
 - adsParams オブジェクト内で、MediaTailor が ADS に渡す必要があるパラメータを指定します。これらのパラメータは、MediaTailor の ADS テンプレート URL にある [player_params.param] 設定に対応しています。
 - 広告 ID シグナリングを有効にするには、adSignaling オブジェクトを最上位オブジェクトとして追加し、その中に enabled および というパラメータ値を追加します true。adSignaling デフォルト値は です disabled。

- 例: HLS 形式

```
POST master.m3u8
{
```

```
"adsParams": {
  "deviceType": "ipad"
},
"adSignaling": {
  "enabled": "true"
},
"reportingMode": "client"
}
```

- 例: DASH 形式

```
POST manifest.mpd
{
  "adsParams": {
    "deviceType": "ipad"
  },
  "adSignaling": {
    "enabled": "true"
  },
  "reportingMode": "client"
}
```

マニフェストと広告メタデータの挿入

広告ステッチングプロセス中に、MediaTailor は、ステッチされる各クリエイティブに関連付けられた一意の ID をマニフェストに追加します。MediaTailor は、VAST レスポンスでそのクリエイティブの id 属性値からクリエイティブの一意の ID を取得します。クリエイティブに ID 属性値がない場合、MediaTailor は空の値 () を発行します id=""。

MediaTailor は、マニフェスト内メタデータシグナルを使用して、広告クリエイティブメタデータとタイムライン全体のタイミング/配置に関するクライアント追跡 API 間の依存関係を切り離します。このデカップリングにより、プレイヤーのユーザーインターフェイス (UI) が再生を初期化する前にタイムラインで広告ブレイク位置をレンダリングする再生レイテンシー (特に VOD シナリオ) が短縮されます。

追加されたメタデータは、次の形式になります。

- HLS マニフェストの場合、追加されたメタデータは、表示期間中の各広告の DATERANGE タグの形式をとります。

- DASH マニフェストの場合、追加されたメタデータは各広告期間内の Event要素の形式になります。

次の JSON メッセージ本文は、VAST レスポンスの例を示しています。

```
{
  "version": 1,
  "identifiers": [
    {
      "scheme": "urn:smp:ul:060E2B34.01040101.01200900.00000000",
      "value": "creativeId",
      "ad_position": "adId",
      "ad_type": "adType",
      "tracking_uri": "trackingUri",
      "custom_vast_data": "customVastData"
    }
  ]
}
```

前の例では、以下のようになっています。

- *creativeId* は、広告の Creative要素のId属性値です。
- *adId* は、広告の先頭に関連付けられた HLS シーケンス番号、または広告の DASH 期間 ID のいずれかです。
- VAST レスポンスに基づく *adType* は overlayavailまたは のいずれかです。
- *trackingUri* は、MediaTailor セッションの相対追跡エンドポイントです。 `../.../.../.../tracking/hashed-account-id/origin-id/session-id`
- *customVastData* は、MediaTailor が creative_signaling VAST 拡張機能から抽出する値です。MediaTailor は、存在する場合、CDATA ノードの内容を使用します。詳細と VAST レスポンスのサンプルについては、[広告決定サーバー \(ADS\) インタラクション](#)「」セクションを参照してください。

広告メタデータを使用した HLS マニフェストのパーソナライズ

ライブ HLS ストリームの場合、MediaTailor は、ストリームに PROGRAM-DATE-TIME タグが含まれている場合にのみ、マニフェスト期間ごとに少なくとも 1 回メタデータを追加します。ビデオオンデマンド (VOD) ストリームの場合、MediaTailor はパーソナライズされたマニフェストで少なくとも 1 つのセグメント PROGRAM-DATE-TIME に追加します。各 VOD アセットの開始時間はエポックゼロ

() です1970-01-01T00:00:00Z。オリジンマニフェストに既存のPROGRAM-DATE-TIMEコンテンツがある場合、MediaTailor はそのコンテンツを保持します。

MediaTailor は、広告決定サーバー (ADS) によって返されるクリエイティブを使用してマニフェストをパーソナライズします。広告ごとに、MediaTailor には広告の期間にまたがるDATERANGEタグも含まれています。DATERANGE タグ形式は、SVA 技術出版物の 2023 バージョンの [DASH および HLS での広告クリエイティブシグナリング](#) セクションで説明されている形式と似ています。

MediaTailor が生成DATERANGEするには一意の ID 値があります。一意性を確保するために ([SCTE-35 を EXT-X-DATERANGE にマッピング](#)で指定されたガイドラインに基づく)、MediaTailor は avail の最初の広告セグメントMEDIA-SEQUENCEの数を avail 内の広告のシーケンス番号と結合します。

スレートが有効になっている設定の広告時間枠が不足している場合、MediaTailor はスレートセグメントをタグで区切って表示の末尾に追加しますがDISCONTINUITY、DATERANGEメタデータはありません。

パーソナライズされたマニフェストにステッチされた広告ごとに、MediaTailor はカスタムDATERANGEタグで base64 でエンコードされたデータとして表されるクリエイティブメタデータを追加します。

Example線形 HLS オリジン (#EXT-X-CUE-OUT):

```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-TARGETDURATION:7
#EXT-X-MEDIA-SEQUENCE:398
#EXT-X-PROGRAM-DATE-TIME:2023-02-10T19:20:01.397Z
#EXTINF:6.006,
index_1_398.ts?m=1676054627
#EXTINF:5.873,
index_1_399.ts?m=1676054627
#EXT-OATCLS-SCTE35:/DA1AAAAAyIYAP/wFAUAAAACf+//jP197P4AUmNiAAEBAQAase4/gA==
#EXT-X-CUE-OUT:59.993
#EXTINF:6.139,
index_1_400.ts?m=1676054627
#EXT-X-CUE-OUT-CONT:ElapsedTime=6.139,Duration=59.993,SCTE35=/DA1AAAAAyIYAP/wFAUAAAACf+//jP197P4AUmNiAAEBAQAase4/gA==
#EXTINF:6.006,
index_1_401.ts?m=1676054627
#EXT-X-CUE-OUT-CONT:ElapsedTime=12.145,Duration=59.993,SCTE35=/DA1AAAAAyIYAP/wFAUAAAACf+//jP197P4AUmNiAAEBAQAase4/gA==
```

```
#EXTINF:6.006,
index_1_402.ts?m=1676054627
#EXT-X-CUE-OUT-CONT:ElapsedTime=18.151,Duration=59.993,SCTE35=/DA1AAAAAyiYAP/wFAUAAAACf
+//jP197P4AUmNiAAEBAQAase4/gA==
#EXTINF:6.006,
index_1_403.ts?m=1676054627
#EXT-X-CUE-OUT-CONT:ElapsedTime=24.157,Duration=59.993,SCTE35=/DA1AAAAAyiYAP/wFAUAAAACf
+//jP197P4AUmNiAAEBAQAase4/gA==
#EXTINF:6.006,
index_1_404.ts?m=1676054627
#EXT-X-CUE-OUT-CONT:ElapsedTime=30.163,Duration=59.993,SCTE35=/DA1AAAAAyiYAP/wFAUAAAACf
+//jP197P4AUmNiAAEBAQAase4/gA==
#EXTINF:6.006,
index_1_405.ts?m=1676054627
#EXT-X-CUE-OUT-CONT:ElapsedTime=36.169,Duration=59.993,SCTE35=/DA1AAAAAyiYAP/wFAUAAAACf
+//jP197P4AUmNiAAEBAQAase4/gA==
#EXTINF:6.006,
index_1_406.ts?m=1676054627
#EXT-X-CUE-OUT-CONT:ElapsedTime=42.175,Duration=59.993,SCTE35=/DA1AAAAAyiYAP/wFAUAAAACf
+//jP197P4AUmNiAAEBAQAase4/gA==
#EXTINF:6.006,
index_1_407.ts?m=1676054627
#EXT-X-CUE-OUT-CONT:ElapsedTime=48.181,Duration=59.993,SCTE35=/DA1AAAAAyiYAP/wFAUAAAACf
+//jP197P4AUmNiAAEBAQAase4/gA==
#EXTINF:6.006,
index_1_408.ts?m=1676054627
#EXT-X-CUE-OUT-CONT:ElapsedTime=54.187,Duration=59.993,SCTE35=/DA1AAAAAyiYAP/wFAUAAAACf
+//jP197P4AUmNiAAEBAQAase4/gA==
#EXTINF:5.806,
index_1_409.ts?m=1676054627
#EXT-X-CUE-IN
#EXTINF:6.206,
index_1_410.ts?m=1676054627
#EXTINF:6.006,
index_1_411.ts?m=1676054627
#EXTINF:6.006,
index_1_412.ts?m=1676054627
```

Example 線形 HLS オリジン (#EXT-X-DATERANGE):

```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-TARGETDURATION:7
```

```
#EXT-X-MEDIA-SEQUENCE:25
#EXT-X-PROGRAM-DATE-TIME:2023-02-10T19:19:53.389Z
#EXTINF:6.006,
index_1_25.ts?m=1676056675
#EXTINF:6.006,
index_1_26.ts?m=1676056675
#EXTINF:6.006,
index_1_27.ts?m=1676056675
#EXTINF:1.869,
index_1_28.ts?m=1676056675
#EXT-X-DATERANGE:ID="2",START-DATE="2023-02-10T19:20:13.276Z",PLANNED-
DURATION=59.993,SCTE35-
OUT=0xFC302500000003289800FFF01405000000027FEFFF8CF97DECFE00526362000101010000B1EE3F80
#EXTINF:6.139,
index_1_29.ts?m=1676056675
#EXTINF:6.006,
index_1_30.ts?m=1676056675
#EXTINF:6.006,
index_1_31.ts?m=1676056675
#EXTINF:6.006,
index_1_32.ts?m=1676056675
#EXTINF:6.006,
index_1_33.ts?m=1676056675
#EXTINF:6.006,
index_1_34.ts?m=1676056675
#EXTINF:6.006,
index_1_35.ts?m=1676056675
#EXTINF:6.006,
index_1_36.ts?m=1676056675
#EXTINF:6.006,
index_1_37.ts?m=1676056675
#EXTINF:5.806,
index_1_38.ts?m=1676056675
#EXT-X-DATERANGE:ID="2",START-DATE="2023-02-10T19:20:13.276Z",END-
DATE="2023-02-10T19:21:13.269Z",DURATION=59.993
#EXTINF:6.206,
index_1_39.ts?m=1676056675
#EXTINF:6.006,
index_1_40.ts?m=1676056675
```

Example 線形 HLS パーソナライズ マニフェスト (クリエイティブ広告シグナリングを使用):

MediaTailor が生成 DATERANGE するには一意の ID 値があります。一意性を確保するために ([SCTE-35 を EXT-X-DATERANGE にマッピング](#)で指定されたガイドラインに基づく)、MediaTailor は avail の最初の広告セグメント MEDIA-SEQUENCE の数を avail 内の広告のシーケンス番号と結合します。

次の例では、MediaTailor は 421 MEDIA-SEQUENCE を広告位置番号と連結します。

```
#EXTM3U
#EXT-X-VERSION:6
#EXT-X-TARGETDURATION:7
#EXT-X-MEDIA-SEQUENCE:418
#EXT-X-DISCONTINUITY-SEQUENCE:5
#EXT-X-PROGRAM-DATE-TIME:2023-02-10T19:19:55.391Z
#EXTINF:6.006,
https://d3fch9e2fcarly.cloudfront.net/out/v1/1cc7058242a74fdd8aea14e22a9b4131/
index_1_397.ts?m=1676054627
#EXTINF:6.006,
https://d3fch9e2fcarly.cloudfront.net/out/v1/1cc7058242a74fdd8aea14e22a9b4131/
index_1_398.ts?m=1676054627
#EXTINF:5.873,
https://d3fch9e2fcarly.cloudfront.net/out/v1/1cc7058242a74fdd8aea14e22a9b4131/
index_1_399.ts?m=1676054627
#EXT-X-DISCONTINUITY
#EXT-X-PROGRAM-DATE-TIME:2023-02-10T19:19:55.391Z
#EXT-X-DATERANGE:ID="421-1",CLASS="urn:sva:advertising-wg:ad-id-signaling",START-
DATE=2019-01-01T00:02:30.000Z,DURATION=15.015,X-AD-CREATIVE-SIGNALING="base64JSON"
#EXTINF:2.002,
../../../../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/9e178fa9-
dce5-4248-83d2-5b5d98b019bf/0/1676056813
#EXTINF:2.002,
../../../../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/9e178fa9-
dce5-4248-83d2-5b5d98b019bf/0/1676056814
#EXTINF:2.002,
../../../../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/9e178fa9-
dce5-4248-83d2-5b5d98b019bf/0/1676056815
#EXTINF:2.002,
../../../../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/9e178fa9-
dce5-4248-83d2-5b5d98b019bf/0/1676056816
#EXTINF:2.002,
../../../../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/9e178fa9-
dce5-4248-83d2-5b5d98b019bf/0/1676056817
```

```
#EXTINF:2.002,
.././../././segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/9e178fa9-
dce5-4248-83d2-5b5d98b019bf/0/1676056818
#EXTINF:2.002,
.././../././segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/9e178fa9-
dce5-4248-83d2-5b5d98b019bf/0/1676056819
#EXTINF:1.001,
.././../././segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/9e178fa9-
dce5-4248-83d2-5b5d98b019bf/0/1676056820
#EXT-X-DISCONTINUITY
#EXT-X-PROGRAM-DATE-TIME:2023-02-10T19:19:55.391Z
#EXT-X-DATERANGE:ID="421-1",START-DATE="2023-02-10T19:36:13.435Z",END-
DATE="2023-02-10T19:36:43.432Z",DURATION=15.015
#EXT-X-DATERANGE:ID="421-2",CLASS="urn:sva:advertising-wg:ad-id-signaling",START-
DATE=2019-01-01T00:02:30.000Z,DURATION=15.015,X-AD-CREATIVE-SIGNALING="base64JSON"
#EXTINF:2.002,
.././../././segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/9e178fa9-
dce5-4248-83d2-5b5d98b019bf/0/1676056821
#EXTINF:2.002,
.././../././segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/9e178fa9-
dce5-4248-83d2-5b5d98b019bf/0/1676056822
#EXTINF:2.002,
.././../././segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/9e178fa9-
dce5-4248-83d2-5b5d98b019bf/0/1676056823
#EXTINF:2.002,
.././../././segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/9e178fa9-
dce5-4248-83d2-5b5d98b019bf/0/1676056824
#EXTINF:2.002,
.././../././segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/9e178fa9-
dce5-4248-83d2-5b5d98b019bf/0/1676056825
#EXTINF:2.002,
.././../././segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/9e178fa9-
dce5-4248-83d2-5b5d98b019bf/0/1676056826
#EXTINF:2.002,
.././../././segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/9e178fa9-
dce5-4248-83d2-5b5d98b019bf/0/1676056827
#EXTINF:1.001,
.././../././segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/9e178fa9-
dce5-4248-83d2-5b5d98b019bf/0/1676056828
#EXT-X-DISCONTINUITY
#EXT-X-PROGRAM-DATE-TIME:2023-02-10T19:19:55.391Z
#EXT-X-DATERANGE:ID="421-2",START-DATE="2023-02-10T19:36:13.435Z",END-
DATE="2023-02-10T19:36:43.432Z",DURATION=15.015
```



```
#EXT-X-DATERANGE:ID="421-3",CLASS="urn:sva:advertising-wg:ad-id-signaling",START-
DATE=2019-01-01T00:02:30.000Z,DURATION=15.015,X-AD-CREATIVE-SIGNALING="base64JSON"
#EXTINF:2.002,
../././././segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/9e178fa9-
dce5-4248-83d2-5b5d98b019bf/0/1676056829
#EXTINF:2.002,
../././././segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/9e178fa9-
dce5-4248-83d2-5b5d98b019bf/0/1676056830
#EXTINF:2.002,
../././././segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/9e178fa9-
dce5-4248-83d2-5b5d98b019bf/0/1676056831
#EXTINF:2.002,
../././././segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/9e178fa9-
dce5-4248-83d2-5b5d98b019bf/0/1676056832
#EXTINF:2.002,
../././././segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/9e178fa9-
dce5-4248-83d2-5b5d98b019bf/0/1676056833
#EXTINF:2.002,
../././././segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/9e178fa9-
dce5-4248-83d2-5b5d98b019bf/0/1676056834
#EXTINF:2.002,
../././././segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/9e178fa9-
dce5-4248-83d2-5b5d98b019bf/0/1676056835
#EXTINF:1.001,
../././././segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/9e178fa9-
dce5-4248-83d2-5b5d98b019bf/0/1676056836
#EXT-X-DISCONTINUITY
#EXT-X-PROGRAM-DATE-TIME:2023-02-10T19:19:55.391Z
#EXT-X-DATERANGE:ID="421-3",START-DATE="2023-02-10T19:36:13.435Z",END-
DATE="2023-02-10T19:36:43.432Z",DURATION=29.997
#EXT-X-DATERANGE:ID="421-4",CLASS="urn:sva:advertising-wg:ad-id-signaling",START-
DATE=2019-01-01T00:02:30.000Z,DURATION=15.015,X-AD-CREATIVE-SIGNALING="base64JSON"
#EXTINF:2.002,
../././././segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/9e178fa9-
dce5-4248-83d2-5b5d98b019bf/0/1676056837
#EXTINF:2.002,
../././././segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/9e178fa9-
dce5-4248-83d2-5b5d98b019bf/0/1676056838
#EXTINF:2.002,
../././././segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/9e178fa9-
dce5-4248-83d2-5b5d98b019bf/0/1676056839
#EXTINF:2.002,
../././././segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/9e178fa9-
dce5-4248-83d2-5b5d98b019bf/0/1676056840
```

```
#EXTINF:2.002,  
../../../../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/9e178fa9-  
dce5-4248-83d2-5b5d98b019bf/0/1676056841  
#EXTINF:2.002,  
../../../../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/9e178fa9-  
dce5-4248-83d2-5b5d98b019bf/0/1676056842  
#EXTINF:2.002,  
../../../../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/9e178fa9-  
dce5-4248-83d2-5b5d98b019bf/0/1676056843  
#EXTINF:1.001,  
../../../../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/9e178fa9-  
dce5-4248-83d2-5b5d98b019bf/0/1676056844  
#EXT-X-DISCONTINUITY  
#EXT-X-DATERANGE:ID="421-4",START-DATE="2023-02-10T19:36:13.435Z",END-  
DATE="2023-02-10T19:36:43.432Z",DURATION=15.015  
#EXTINF:6.206,  
https://d3fch9e2fcarly.cloudfront.net/out/v1/1cc7058242a74fdd8aea14e22a9b4131/  
index_1_410.ts?m=1676054627  
#EXTINF:6.006,  
https://d3fch9e2fcarly.cloudfront.net/out/v1/1cc7058242a74fdd8aea14e22a9b4131/  
index_1_411.ts?m=1676054627
```

Example VOD HLS オリジン (SCTE シグナルを使用):

```
#EXTM3U  
#EXT-X-VERSION:3  
#EXT-X-TARGETDURATION:7  
#EXT-X-MEDIA-SEQUENCE:1  
#EXT-X-PLAYLIST-TYPE:VOD  
#EXTINF:6,  
index_720p1500k_00001.ts  
#EXTINF:6,  
index_720p1500k_00002.ts  
#EXTINF:6,  
index_720p1500k_00003.ts  
#EXTINF:6,  
index_720p1500k_00004.ts  
#EXTINF:6,  
index_720p1500k_00005.ts  
#EXT-X-CUE-OUT:0  
#EXT-X-CUE-IN  
#EXTINF:6,  
index_720p1500k_00006.ts
```

```
#EXTINF:6,  
index_720p1500k_00007.ts  
#EXTINF:6,  
index_720p1500k_00008.ts  
#EXTINF:6,  
index_720p1500k_00009.ts  
#EXTINF:6,  
index_720p1500k_00010.ts  
#EXTINF:6,  
index_720p1500k_00011.ts  
#EXTINF:6,  
index_720p1500k_00012.ts
```

Example VOD HLS オリジン:

```
#EXTM3U  
#EXT-X-VERSION:3  
#EXT-X-TARGETDURATION:7  
#EXT-X-MEDIA-SEQUENCE:1  
#EXT-X-PLAYLIST-TYPE:VOD  
#EXTINF:6,  
index_720p1500k_00001.ts  
#EXTINF:6,  
index_720p1500k_00002.ts  
#EXTINF:6,  
index_720p1500k_00003.ts  
#EXTINF:6,  
index_720p1500k_00004.ts  
#EXTINF:4,  
index_720p1500k_00005.ts  
#EXTINF:2,  
index_720p1500k_00006.ts  
#EXTINF:6,  
index_720p1500k_00007.ts  
#EXTINF:6,  
index_720p1500k_00008.ts  
#EXTINF:6,  
index_720p1500k_00009.ts  
#EXTINF:6,  
index_720p1500k_00010.ts  
#EXTINF:6,  
index_720p1500k_00011.ts  
#EXTINF:6,
```

```
index_720p1500k_00012.ts
```

Example VOD HLS パーソナライズされたマニフェスト:

MediaTailor は、広告位置を示す HLS DATERANGE要素のアンカーとして使用するために、VOD マニフェストPROGRAM-DATE-TIMEに を追加します。

MediaTailor が生成DATERANGEする には一意の ID 値があります。一意性を確保するために ([SCTE-35](#) を [EXT-X-DATERANGE](#) にマッピングで指定されたガイドラインに基づく)、MediaTailor は、avail の最初の広告セグメントMEDIA-SEQUENCEの数を、avail 内の広告のシーケンス番号と結合します。

次の例では、MediaTailor は 421 MEDIA-SEQUENCE を広告位置番号と連結します。

```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-PLAYLIST-TYPE:VOD
#EXT-X-TARGETDURATION:7
#EXT-X-MEDIA-SEQUENCE:1
#EXT-X-DISCONTINUITY-SEQUENCE:0
#EXT-X-PROGRAM-DATE-TIME:1970-01-01T00:00:00Z
#EXTINF:6.0,
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SK0-22/asset-1/hls/
index_720p1500k_00001.ts
#EXTINF:6.0,
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SK0-22/asset-1/hls/
index_720p1500k_00002.ts
#EXTINF:6.0,
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SK0-22/asset-1/hls/
index_720p1500k_00003.ts
#EXTINF:6.0,
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SK0-22/asset-1/hls/
index_720p1500k_00004.ts
#EXTINF:4.0,
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SK0-22/asset-1/hls/
index_720p1500k_00005.ts
#EXT-X-DISCONTINUITY
#EXTINF:2.002,
../../../../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/vod-
variations/9810d863-8736-45fa-866e-be6d2c2bfa20/0/28
#EXTINF:2.002,
../../../../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/vod-
variations/9810d863-8736-45fa-866e-be6d2c2bfa20/0/29
```

```
#EXTINF:2.002,
../.../.../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/vod-
variations/9810d863-8736-45fa-866e-be6d2c2bfa20/0/30
#EXTINF:2.002,
../.../.../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/vod-
variations/9810d863-8736-45fa-866e-be6d2c2bfa20/0/31
#EXTINF:2.002,
../.../.../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/vod-
variations/9810d863-8736-45fa-866e-be6d2c2bfa20/0/32
#EXTINF:2.002,
../.../.../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/vod-
variations/9810d863-8736-45fa-866e-be6d2c2bfa20/0/33
#EXTINF:2.002,
../.../.../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/vod-
variations/9810d863-8736-45fa-866e-be6d2c2bfa20/0/34
#EXTINF:1.001,
../.../.../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/vod-
variations/9810d863-8736-45fa-866e-be6d2c2bfa20/0/35
#EXT-X-DISCONTINUITY
#EXTINF:2.002,
../.../.../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/vod-
variations/9810d863-8736-45fa-866e-be6d2c2bfa20/0/36
#EXTINF:2.002,
../.../.../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/vod-
variations/9810d863-8736-45fa-866e-be6d2c2bfa20/0/37
#EXTINF:2.002,
../.../.../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/vod-
variations/9810d863-8736-45fa-866e-be6d2c2bfa20/0/38
#EXTINF:2.002,
../.../.../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/vod-
variations/9810d863-8736-45fa-866e-be6d2c2bfa20/0/39
#EXTINF:2.002,
../.../.../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/vod-
variations/9810d863-8736-45fa-866e-be6d2c2bfa20/0/40
#EXTINF:2.002,
../.../.../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/vod-
variations/9810d863-8736-45fa-866e-be6d2c2bfa20/0/41
#EXTINF:2.002,
../.../.../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/vod-
variations/9810d863-8736-45fa-866e-be6d2c2bfa20/0/42
#EXTINF:1.001,
../.../.../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/vod-
variations/9810d863-8736-45fa-866e-be6d2c2bfa20/0/43
#EXT-X-DISCONTINUITY
```

```
#EXTINF:2.0,  
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SK0-22/asset-1/hls/  
index_720p1500k_00006.ts  
#EXTINF:6.0,  
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SK0-22/asset-1/hls/  
index_720p1500k_00007.ts  
#EXTINF:6.0,  
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SK0-22/asset-1/hls/  
index_720p1500k_00008.ts  
#EXTINF:6.0,  
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SK0-22/asset-1/hls/  
index_720p1500k_00009.ts  
#EXTINF:6.0,  
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SK0-22/asset-1/hls/  
index_720p1500k_00010.ts  
#EXTINF:6.0,  
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SK0-22/asset-1/hls/  
index_720p1500k_00011.ts  
#EXTINF:6.0,  
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SK0-22/asset-1/hls/  
index_720p1500k_00012.ts  
#EXT-X-ENDLIST  
#EXT-X-DATERANGE:ID="5-1",START-DATE="1970-01-01T00:00:28.000Z",END-  
DATE="1970-01-01T00:00:43.015Z",DURATION=15.015  
#EXT-X-DATERANGE:ID="5-2",START-DATE="1970-01-01T00:00:43.015Z",END-  
DATE="1970-01-01T00:00:58.030Z",DURATION=15.01
```

広告メタデータを使用した DASH マニフェストのパーソナライズ

MediaTailor は、広告決定サーバー (ADS) によって返されるクリエイティブを使用してマニフェストをパーソナライズします。広告ごとに、MediaTailor には広告の期間にまたがる EventStream 要素も含まれています。Event 要素形式は、SVA テクニカルパブリケーションの 2023 バージョンの [DASH および HLS での広告クリエイティブシグナリング](#) セクションで説明されているものと似ています。

スレートが有効になっている設定の広告時間枠が不足している場合、MediaTailor はスレート期間を使用可能期間の最後に追加しますが、EventStream メタデータは追加しません。

パーソナライズされたマニフェストにステッチされた広告ごとに、MediaTailor は Event 要素内の CDATA 要素として表されるクリエイティブメタデータを追加します。

Example 線形 DASH オリジン (インライン SCTE 属性):

```

<MPD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="urn:mpeg:dash:schema:mpd:2011" xmlns:scte35="urn:scte:scte35:2013:xml"
  xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 http://standards.iso.org/
  ittf/PubliclyAvailableStandards/MPEG-DASH_schema_files/DASH-MPD.xsd" id="201"
  type="dynamic" publishTime="2023-02-10T21:08:40+00:00" minimumUpdatePeriod="PT6S"
  availabilityStartTime="2023-02-09T22:47:05.865000+00:00" minBufferTime="PT10S"
  suggestedPresentationDelay="PT20.000S" timeShiftBufferDepth="PT88.999S"
  profiles="urn:mpeg:dash:profile:isoff-live:2011">
  <Period start="PT80141.456S" id="104" duration="PT304.103S">
    <AdaptationSet id="1485523442" mimeType="video/mp4" segmentAlignment="true"
    startWithSAP="1" subsegmentAlignment="true" subsegmentStartsWithSAP="1"
    bitstreamSwitching="true">
      <SegmentTemplate timescale="60000" media="index_video_$RepresentationID$_0_
      $Number$.mp4?m=1676062374" initialization="index_video_$RepresentationID$_0_init.mp4?
      m=1676062374" startNumber="151" presentationTimeOffset="4808487386">
        <SegmentTimeline>
          <S t="4824975858" d="360360" r="3"/>
          <S t="4826417298" d="316316"/>
        </SegmentTimeline>
      </SegmentTemplate>
      <Representation id="1" width="960" height="540" frameRate="30000/1001"
      bandwidth="1800000" codecs="avc1.4D401F"/>
      <Representation id="3" width="640" height="360" frameRate="30000/1001"
      bandwidth="1200000" codecs="avc1.4D401E"/>
      <Representation id="5" width="480" height="270" frameRate="30000/1001"
      bandwidth="800000" codecs="avc1.4D4015"/>
    </AdaptationSet>
    <AdaptationSet id="1377232898" mimeType="audio/mp4" segmentAlignment="0"
    lang="eng">
      <Label>eng</Label>
      <SegmentTemplate timescale="48000" media="index_audio_$RepresentationID$_0_
      $Number$.mp4?m=1676062374" initialization="index_audio_$RepresentationID$_0_init.mp4?
      m=1676062374" startNumber="151" presentationTimeOffset="3846790126">
        <SegmentTimeline>
          <S t="3859981294" d="287744"/>
          <S t="3860269038" d="288768"/>
          <S t="3860557806" d="287744"/>
          <S t="3860845550" d="288768"/>
          <S t="3861134318" d="252928"/>
        </SegmentTimeline>
      </SegmentTemplate>
    </AdaptationSet>
  </Period>
</MPD>

```

```

    <Representation id="2" bandwidth="193007" audioSamplingRate="48000"
codecs="mp4a.40.2">
    <AudioChannelConfiguration
schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
    </Representation>
    <Representation id="4" bandwidth="193007" audioSamplingRate="48000"
codecs="mp4a.40.2">
    <AudioChannelConfiguration
schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
    </Representation>
    <Representation id="6" bandwidth="193007" audioSamplingRate="48000"
codecs="mp4a.40.2">
    <AudioChannelConfiguration
schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
    </Representation>
  </AdaptationSet>
  <SupplementalProperty schemeIdUri="urn:scte:dash:utc-time"
value="2023-02-10T21:02:31.007Z"/>
</Period>
<Period start="PT80445.560S" id="155" duration="PT44.978S">
  <EventStream timescale="90000" schemeIdUri="urn:scte:scte35:2013:xml">
    <Event duration="4048044">
      <scte35:SpliceInfoSection protocolVersion="0" ptsAdjustment="207000"
tier="4095">
        <scte35:SpliceInsert spliceEventId="111" spliceEventCancelIndicator="false"
outOfNetworkIndicator="true" spliceImmediateFlag="false" uniqueProgramId="1"
availNum="1" availsExpected="1">
          <scte35:Program>
            <scte35:SpliceTime ptsTime="7239893422"/>
          </scte35:Program>
          <scte35:BreakDuration autoReturn="true" duration="4048044"/>
        </scte35:SpliceInsert>
      </scte35:SpliceInfoSection>
    </Event>
  </EventStream>
  <AdaptationSet id="1485523442" mimeType="video/mp4" segmentAlignment="true"
startWithSAP="1" subsegmentAlignment="true" subsegmentStartsWithSAP="1"
bitstreamSwitching="true">
    <SegmentTemplate timescale="60000" media="index_video_${RepresentationID$_0_
$Number$.mp4?m=1676062374" initialization="index_video_${RepresentationID$_0_init}.mp4?
m=1676062374" startNumber="156" presentationTimeOffset="4826733614">
      <SegmentTimeline>
        <S t="4826733614" d="284284"/>
        <S t="4827017898" d="360360" r="5"/>
      </SegmentTimeline>
    </AdaptationSet>
  </Period>
</Timeline>

```



```

        <S t="4829180058" d="252252"/>
    </SegmentTimeline>
</SegmentTemplate>
    <Representation id="1" width="960" height="540" frameRate="30000/1001"
bandwidth="1800000" codecs="avc1.4D401F"/>
    <Representation id="3" width="640" height="360" frameRate="30000/1001"
bandwidth="1200000" codecs="avc1.4D401E"/>
    <Representation id="5" width="480" height="270" frameRate="30000/1001"
bandwidth="800000" codecs="avc1.4D4015"/>
</AdaptationSet>
    <AdaptationSet id="1377232898" mimeType="audio/mp4" segmentAlignment="0"
lang="eng">
        <Label>eng</Label>
        <SegmentTemplate timescale="48000" media="index_audio_$RepresentationID$_0_
$Number$.mp4?m=1676062374" initialization="index_audio_$RepresentationID$_0_init.mp4?
m=1676062374" startNumber="156" presentationTimeOffset="3861387246">
            <SegmentTimeline>
                <S t="3861387246" d="227328"/>
                <S t="3861614574" d="288768"/>
                <S t="3861903342" d="287744"/>
                <S t="3862191086" d="288768"/>
                <S t="3862479854" d="287744"/>
                <S t="3862767598" d="288768"/>
                <S t="3863056366" d="287744"/>
                <S t="3863344110" d="202752"/>
            </SegmentTimeline>
        </SegmentTemplate>
        <Representation id="2" bandwidth="193007" audioSamplingRate="48000"
codecs="mp4a.40.2">
            <AudioChannelConfiguration
schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
        </Representation>
        <Representation id="4" bandwidth="193007" audioSamplingRate="48000"
codecs="mp4a.40.2">
            <AudioChannelConfiguration
schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
        </Representation>
        <Representation id="6" bandwidth="193007" audioSamplingRate="48000"
codecs="mp4a.40.2">
            <AudioChannelConfiguration
schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
        </Representation>
    </AdaptationSet>

```

```

    <SupplementalProperty schemeIdUri="urn:scte:dash:utc-time"
value="2023-02-10T21:07:35.111Z"/>
  </Period>
  <Period start="PT80490.538S" id="163">
    <AdaptationSet id="1485523442" mimeType="video/mp4" segmentAlignment="true"
startWithSAP="1" subsegmentAlignment="true" subsegmentStartsWithSAP="1"
bitstreamSwitching="true">
      <SegmentTemplate timescale="60000" media="index_video_$RepresentationID$_0_
$Number$.mp4?m=1676062374" initialization="index_video_$RepresentationID$_0_init.mp4?
m=1676062374" startNumber="164" presentationTimeOffset="4829432310">
        <SegmentTimeline>
          <S t="4829432310" d="348348"/>
          <S t="4829780658" d="360360" r="1"/>
        </SegmentTimeline>
      </SegmentTemplate>
      <Representation id="1" width="960" height="540" frameRate="30000/1001"
bandwidth="1800000" codecs="avc1.4D401F"/>
      <Representation id="3" width="640" height="360" frameRate="30000/1001"
bandwidth="1200000" codecs="avc1.4D401E"/>
      <Representation id="5" width="480" height="270" frameRate="30000/1001"
bandwidth="800000" codecs="avc1.4D4015"/>
    </AdaptationSet>
    <AdaptationSet id="1377232898" mimeType="audio/mp4" segmentAlignment="0"
lang="eng">
      <Label>eng</Label>
      <SegmentTemplate timescale="48000" media="index_audio_$RepresentationID$_0_
$Number$.mp4?m=1676062374" initialization="index_audio_$RepresentationID$_0_init.mp4?
m=1676062374" startNumber="164" presentationTimeOffset="3863546862">
        <SegmentTimeline>
          <S t="3863546862" d="278528"/>
          <S t="3863825390" d="287744"/>
          <S t="3864113134" d="288768"/>
        </SegmentTimeline>
      </SegmentTemplate>
      <Representation id="2" bandwidth="193007" audioSamplingRate="48000"
codecs="mp4a.40.2">
        <AudioChannelConfiguration
schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
      </Representation>
      <Representation id="4" bandwidth="193007" audioSamplingRate="48000"
codecs="mp4a.40.2">
        <AudioChannelConfiguration
schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
      </Representation>

```

```

    <Representation id="6" bandwidth="193007" audioSamplingRate="48000"
    codecs="mp4a.40.2">
      <AudioChannelConfiguration
    schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
    </Representation>
  </AdaptationSet>
  <SupplementalProperty schemeIdUri="urn:scte:dash:utc-time"
    value="2023-02-10T21:08:20.090Z"/>
</Period>
</MPD>

```

Example 線形 DASH パーソナライズマニフェスト (クリエイティブ広告シグナリングを使用):

```

<MPD availabilityStartTime="2023-02-09T22:47:05.865000+00:00"
  id="201" minBufferTime="PT10S" minimumUpdatePeriod="PT6S"
  profiles="urn:mpeg:dash:profile:isoff-live:2011"
  publishTime="2023-02-10T21:08:43+00:00" suggestedPresentationDelay="PT20.000S"
  timeShiftBufferDepth="PT88.999S" type="dynamic" xmlns="urn:mpeg:dash:schema:mpd:2011"
  xmlns:scte35="urn:scte:scte35:2013:xml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
  instance" xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 http://standards.iso.org/
  ittf/PubliclyAvailableStandards/MPEG-DASH_schema_files/DASH-MPD.xsd">
  <BaseURL>https://d3fch9e2fcarly.cloudfront.net/out/v1/
  f9f38deca3f14fc4b5ab3cdbc76cfb9e/</BaseURL>
  <Location>https://777788889999.mediatailor.us-west-2.amazonaws.com/
  v1/dash/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/out/v1/
  f9f38deca3f14fc4b5ab3cdbc76cfb9e/index.mpd?
  aws.sessionId=672ed481-4ffd-4270-936f-7c8403947f2e</Location>
  <Period duration="PT304.103S" id="104" start="PT80141.456S">
    <AdaptationSet bitstreamSwitching="true" id="1485523442" mimeType="video/
    mp4" segmentAlignment="true" startWithSAP="1" subsegmentAlignment="true"
    subsegmentStartsWithSAP="1">
      <SegmentTemplate initialization="index_video_$RepresentationID$_0_init.mp4?
      m=1676062374" media="index_video_$RepresentationID$_0_$Number$.mp4?m=1676062374"
      presentationTimeOffset="4808487386" startNumber="151" timescale="60000">
        <SegmentTimeline>
          <S d="360360" r="3" t="4824975858"/>
          <S d="316316" t="4826417298"/>
        </SegmentTimeline>
      </SegmentTemplate>
      <Representation bandwidth="1800000" codecs="avc1.4D401F"
      frameRate="30000/1001" height="540" id="1" width="960"/>
      <Representation bandwidth="1200000" codecs="avc1.4D401E"
      frameRate="30000/1001" height="360" id="3" width="640"/>
    </AdaptationSet>
  </Period>
</MPD>

```

```

        <Representation bandwidth="800000" codecs="avc1.4D4015"
frameRate="30000/1001" height="270" id="5" width="480"/>
    </AdaptationSet>
    <AdaptationSet id="1377232898" lang="eng" mimeType="audio/mp4"
segmentAlignment="0">
        <Label>eng</Label>
        <SegmentTemplate initialization="index_audio_$RepresentationID$_0_init.mp4?
m=1676062374" media="index_audio_$RepresentationID$_0_$Number$.mp4?m=1676062374"
presentationTimeOffset="3846790126" startNumber="151" timescale="48000">
            <SegmentTimeline>
                <S d="287744" t="3859981294"/>
                <S d="288768" t="3860269038"/>
                <S d="287744" t="3860557806"/>
                <S d="288768" t="3860845550"/>
                <S d="252928" t="3861134318"/>
            </SegmentTimeline>
        </SegmentTemplate>
        <Representation audioSamplingRate="48000" bandwidth="193007"
codecs="mp4a.40.2" id="2">
            <AudioChannelConfiguration
schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
        </Representation>
        <Representation audioSamplingRate="48000" bandwidth="193007"
codecs="mp4a.40.2" id="4">
            <AudioChannelConfiguration
schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
        </Representation>
        <Representation audioSamplingRate="48000" bandwidth="193007"
codecs="mp4a.40.2" id="6">
            <AudioChannelConfiguration
schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
        </Representation>
    </AdaptationSet>
    <SupplementalProperty schemeIdUri="urn:scte:dash:utc-time"
value="2023-02-10T21:02:31.007Z"/>
</Period>
<Period id="155_1" start="PT22H20M45.56S">
    <BaseURL>https://777788889999.mediatailor.us-west-2.amazonaws.com/
v1/dashsegment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/
emt/672ed481-4ffd-4270-936f-7c8403947f2e/155/155_1/</BaseURL>
    <EventStream schemeIdUri="urn:sva:advertising-wg:ad-id-signaling"
timescale="90000">
        <Event presentationTime="xxxxx" duration="1351350">

```

```

        <![CDATA[{"version": 1,"identifiers": [{"scheme":
"urn:smppte:ul:060E2B34.01040101.01200900.00000000","value": "155_1","ad_position":
"155_1", "ad_type":"avail","creative_id": "123","tracking_uri": "../v1/
tracking/hashed-account-id/origin-id/session-id","custom_vast_data":"123abc"}]]]>
    </Event>
</EventStream>
    <AdaptationSet bitstreamSwitching="false" frameRate="30000/1001"
mimeType="video/mp4" segmentAlignment="true" startWithSAP="1"
subsegmentAlignment="true" subsegmentStartsWithSAP="1">
        <SegmentTemplate startNumber="1" timescale="90000"/>
        <Representation bandwidth="1800000" codecs="avc1.64001f" height="540"
id="1" width="960">
            <SegmentTemplate initialization="asset_540_1_2init.mp4"
media="asset_540_1_2_${Number%09d$.mp4" startNumber="1" timescale="90000">
                <SegmentTimeline>
                    <S d="180180" r="6" t="0"/>
                    <S d="90090" t="1261260"/>
                </SegmentTimeline>
            </SegmentTemplate>
        </Representation>
        <Representation bandwidth="1200000" codecs="avc1.64001e" height="360"
id="3" width="640">
            <SegmentTemplate initialization="asset_360_1_1init.mp4"
media="asset_360_1_1_${Number%09d$.mp4" startNumber="1" timescale="90000">
                <SegmentTimeline>
                    <S d="180180" r="6" t="0"/>
                    <S d="90090" t="1261260"/>
                </SegmentTimeline>
            </SegmentTemplate>
        </Representation>
        <Representation bandwidth="800000" codecs="avc1.640015" height="270" id="5"
width="480">
            <SegmentTemplate initialization="asset_270_0_0init.mp4"
media="asset_270_0_0_${Number%09d$.mp4" startNumber="1" timescale="90000">
                <SegmentTimeline>
                    <S d="180180" r="6" t="0"/>
                    <S d="90090" t="1261260"/>
                </SegmentTimeline>
            </SegmentTemplate>
        </Representation>
    </AdaptationSet>
    <AdaptationSet lang="eng" mimeType="audio/mp4" segmentAlignment="0">
        <SegmentTemplate initialization="asset_audio_128_3init.mp4"
media="asset_audio_128_3_${Number%09d$.mp4" startNumber="1" timescale="48000"/>

```

```

        <Label>eng</Label>
        <Representation audioSamplingRate="48000" bandwidth="128000"
codecs="mp4a.40.2" id="6">
            <SegmentTemplate initialization="asset_audio_128_3init.mp4"
media="asset_audio_128_3_${Number%09d$.mp4" startNumber="1" timescale="48000">
                <SegmentTimeline>
                    <S d="98304" t="0"/>
                    <S d="96256" r="1" t="98304"/>
                    <S d="95232" t="290816"/>
                    <S d="96256" r="2" t="386048"/>
                    <S d="48128" t="674816"/>
                </SegmentTimeline>
            </SegmentTemplate>
            <AudioChannelConfiguration
schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
        </Representation>
    </AdaptationSet>
</Period>
<Period id="155_2" start="PT22H21M0.575S">
    <BaseURL>https://777788889999.mediatailor.us-west-2.amazonaws.com/
v1/dashsegment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/
emt/672ed481-4ffd-4270-936f-7c8403947f2e/155/155_2/</BaseURL>
    <EventStream schemeIdUri="urn:sva:advertising-wg:ad-id-signaling"
timescale="90000">
        <Event presentationTime="0" duration="1351350">
            <![CDATA[{"version": 1,"identifiers": [{"scheme":
"urn:smp:ul:060E2B34.01040101.01200900.00000000","value": "155_2","ad_position":
"155_2", "ad_type":"avail","creative_id": "234","tracking_uri": "../v1/
tracking/hashed-account-id/origin-id/session-id","custom_vast_data":"123abc"}]]]>
        </Event>
    </EventStream>
    <AdaptationSet bitstreamSwitching="false" frameRate="30000/1001"
mimeType="video/mp4" segmentAlignment="true" startWithSAP="1"
subsegmentAlignment="true" subsegmentStartsWithSAP="1">
        <SegmentTemplate startNumber="1" timescale="90000"/>
        <Representation bandwidth="1800000" codecs="avc1.64001f" height="540"
id="1" width="960">
            <SegmentTemplate initialization="asset_540_1_2init.mp4"
media="asset_540_1_2_${Number%09d$.mp4" startNumber="1" timescale="90000">
                <SegmentTimeline>
                    <S d="180180" r="6" t="0"/>
                    <S d="90090" t="1261260"/>
                </SegmentTimeline>
            </SegmentTemplate>

```

```

        </Representation>
        <Representation bandwidth="1200000" codecs="avc1.64001e" height="360"
id="3" width="640">
            <SegmentTemplate initialization="asset_360_1_1init.mp4"
media="asset_360_1_1_${Number%09d$.mp4" startNumber="1" timescale="90000">
                <SegmentTimeline>
                    <S d="180180" r="6" t="0"/>
                    <S d="90090" t="1261260"/>
                </SegmentTimeline>
            </SegmentTemplate>
        </Representation>
        <Representation bandwidth="800000" codecs="avc1.640015" height="270" id="5"
width="480">
            <SegmentTemplate initialization="asset_270_0_0init.mp4"
media="asset_270_0_0_${Number%09d$.mp4" startNumber="1" timescale="90000">
                <SegmentTimeline>
                    <S d="180180" r="6" t="0"/>
                    <S d="90090" t="1261260"/>
                </SegmentTimeline>
            </SegmentTemplate>
        </Representation>
    </AdaptationSet>
    <AdaptationSet lang="eng" mimeType="audio/mp4" segmentAlignment="0">
        <SegmentTemplate initialization="asset_audio_128_3init.mp4"
media="asset_audio_128_3_${Number%09d$.mp4" startNumber="1" timescale="48000"/>
        <Label>eng</Label>
        <Representation audioSamplingRate="48000" bandwidth="128000"
codecs="mp4a.40.2" id="6">
            <SegmentTemplate initialization="asset_audio_128_3init.mp4"
media="asset_audio_128_3_${Number%09d$.mp4" startNumber="1" timescale="48000">
                <SegmentTimeline>
                    <S d="98304" t="0"/>
                    <S d="96256" r="1" t="98304"/>
                    <S d="95232" t="290816"/>
                    <S d="96256" r="2" t="386048"/>
                    <S d="48128" t="674816"/>
                </SegmentTimeline>
            </SegmentTemplate>
            <AudioChannelConfiguration
schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
        </Representation>
    </AdaptationSet>
</Period>
<Period id="155_3" start="PT22H21M15.59S">

```

```

    <BaseURL>https://777788889999.mediatailor.us-west-2.amazonaws.com/
v1/dashsegment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/
emt/672ed481-4ffd-4270-936f-7c8403947f2e/155/155_3/</BaseURL>
    <EventStream schemeIdUri="urn:sva:advertising-wg:ad-id-signaling"
timescale="90000">
        <Event presentationTime="0" duration="1351350">
            <![CDATA[{"version": 1,"identifiers": [{"scheme":
"urn:smppte:ul:060E2B34.01040101.01200900.00000000","value": "155_3","ad_position":
"155_3", "ad_type":"avail","creative_id": "345","tracking_uri": "../..../v1/
tracking/hashed-account-id/origin-id/session-id","custom_vast_data":"123abc"}]]]>
        </Event>
    </EventStream>
    <AdaptationSet bitstreamSwitching="false" frameRate="30000/1001"
mimeType="video/mp4" segmentAlignment="true" startWithSAP="1"
subsegmentAlignment="true" subsegmentStartsWithSAP="1">
        <SegmentTemplate startNumber="1" timescale="90000"/>
        <Representation bandwidth="1800000" codecs="avc1.64001f" height="540"
id="1" width="960">
            <SegmentTemplate initialization="asset_540_1_2init.mp4"
media="asset_540_1_2_${Number%09d$.mp4}" startNumber="1" timescale="90000">
                <SegmentTimeline>
                    <S d="180180" r="6" t="0"/>
                    <S d="90090" t="1261260"/>
                </SegmentTimeline>
            </SegmentTemplate>
        </Representation>
        <Representation bandwidth="1200000" codecs="avc1.64001e" height="360"
id="3" width="640">
            <SegmentTemplate initialization="asset_360_1_1init.mp4"
media="asset_360_1_1_${Number%09d$.mp4}" startNumber="1" timescale="90000">
                <SegmentTimeline>
                    <S d="180180" r="6" t="0"/>
                    <S d="90090" t="1261260"/>
                </SegmentTimeline>
            </SegmentTemplate>
        </Representation>
        <Representation bandwidth="800000" codecs="avc1.640015" height="270" id="5"
width="480">
            <SegmentTemplate initialization="asset_270_0_0init.mp4"
media="asset_270_0_0_${Number%09d$.mp4}" startNumber="1" timescale="90000">
                <SegmentTimeline>
                    <S d="180180" r="6" t="0"/>
                    <S d="90090" t="1261260"/>
                </SegmentTimeline>
            </SegmentTemplate>
        </Representation>
    </AdaptationSet>
</MPD>

```



```

        </SegmentTemplate>
    </Representation>
</AdaptationSet>
<AdaptationSet lang="eng" mimeType="audio/mp4" segmentAlignment="0">
    <SegmentTemplate initialization="asset_audio_128_3init.mp4"
media="asset_audio_128_3_${Number%09d$.mp4" startNumber="1" timescale="48000"/>
    <Label>eng</Label>
    <Representation audioSamplingRate="48000" bandwidth="128000"
codecs="mp4a.40.2" id="6">
        <SegmentTemplate initialization="asset_audio_128_3init.mp4"
media="asset_audio_128_3_${Number%09d$.mp4" startNumber="1" timescale="48000">
            <SegmentTimeline>
                <S d="98304" t="0"/>
                <S d="96256" r="1" t="98304"/>
                <S d="95232" t="290816"/>
                <S d="96256" r="2" t="386048"/>
                <S d="48128" t="674816"/>
            </SegmentTimeline>
        </SegmentTemplate>
    <AudioChannelConfiguration
schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
    </Representation>
</AdaptationSet>
</Period>
<Period id="163" start="PT80490.538S">
    <AdaptationSet bitstreamSwitching="true" id="1485523442" mimeType="video/
mp4" segmentAlignment="true" startWithSAP="1" subsegmentAlignment="true"
subsegmentStartsWithSAP="1">
        <SegmentTemplate initialization="index_video_${RepresentationID$_0_init.mp4?
m=1676062374" media="index_video_${RepresentationID$_0}_${Number$.mp4?m=1676062374"
presentationTimeOffset="4829432310" startNumber="164" timescale="60000">
            <SegmentTimeline>
                <S d="348348" t="4829432310"/>
                <S d="360360" r="1" t="4829780658"/>
            </SegmentTimeline>
        </SegmentTemplate>
        <Representation bandwidth="1800000" codecs="avc1.4D401F"
frameRate="30000/1001" height="540" id="1" width="960"/>
        <Representation bandwidth="1200000" codecs="avc1.4D401E"
frameRate="30000/1001" height="360" id="3" width="640"/>
        <Representation bandwidth="800000" codecs="avc1.4D4015"
frameRate="30000/1001" height="270" id="5" width="480"/>
    </AdaptationSet>

```

```

    <AdaptationSet id="1377232898" lang="eng" mimeType="audio/mp4"
segmentAlignment="0">
      <Label>eng</Label>
      <SegmentTemplate initialization="index_audio_$RepresentationID$_0_init.mp4?
m=1676062374" media="index_audio_$RepresentationID$_0_$Number$.mp4?m=1676062374"
presentationTimeOffset="3863546862" startNumber="164" timescale="48000">
        <SegmentTimeline>
          <S d="278528" t="3863546862"/>
          <S d="287744" t="3863825390"/>
          <S d="288768" t="3864113134"/>
        </SegmentTimeline>
      </SegmentTemplate>
      <Representation audioSamplingRate="48000" bandwidth="193007"
codecs="mp4a.40.2" id="2">
        <AudioChannelConfiguration
schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
      </Representation>
      <Representation audioSamplingRate="48000" bandwidth="193007"
codecs="mp4a.40.2" id="4">
        <AudioChannelConfiguration
schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
      </Representation>
      <Representation audioSamplingRate="48000" bandwidth="193007"
codecs="mp4a.40.2" id="6">
        <AudioChannelConfiguration
schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
      </Representation>
    </AdaptationSet>
    <SupplementalProperty schemeIdUri="urn:scte:dash:utc-time"
value="2023-02-10T21:08:20.090Z"/>
  </Period>
</MPD>

```

広告決定サーバー (ADS) インタラクション

MediaTailor は、VAST レスポンスのクリエイティブ id 属性値を広告 ID シグナリングの値として使用します。id 属性値が空であるか、VAST レスポンスに存在しない場合、MediaTailor は広告 ID シグナリングに空の値を配置します。

Example VAST レスポンス:

次の VAST レスポンスのサンプルには、インライン線形 のid属性値が含まれていますCreative。MediaTailor はカスタム VAST Extension要素から値を抽出し、その値をマニフェストのクリエイティブメタデータに配置します。

```
<?xml version="1.0" encoding="utf-8"?>
<VAST version="3.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Ad sequence="3">
    <Inline>
      <AdSystem>2.0</AdSystem>
      <AdTitle>AD-caribbean2-15</AdTitle>
      <Impression><![CDATA[https://n8ljfs0xxx.execute-api.us-west-2.amazonaws.com/v1/impression]]></Impression>
      <Creatives>
        <Creative sequence="3" apiFramework="inLine" id="1234">
          <Linear>
            <Duration>00:00:15</Duration>
            <MediaFiles>
              <MediaFile id="00002" delivery="progressive" type="video/mp4" width="1280" height="720"><![CDATA[https://d3re4i3vgppxxx.cloudfront.net/Media/Bumpers/AD-caribbean2-15-HD.mp4]]></MediaFile>
            </MediaFiles>
          </Linear>
        </Creative>
      </Creatives>
      <Extensions>
        <Extension type="creative_signaling"><![CDATA[999999|TVN1DDNpFTchtpRj,E5TfTtcYd5IEzvEt,ChA050HcvWRGFY6Zp5VSS1xUEJ2B9p8GGhQIDzIQkFeQC-Ho67FR3P9qNa6khSAGKgAyAA]]></Extension>
      </Extensions>
    </Inline>
  </Ad>
</VAST>
```

クライアント側の追跡 API

次の例は、プレイヤー SDK がマニフェストの広告メタデータを、 creativeIdおよび を使用したクライアント側の追跡レスポンスペイロードの完全な追跡イベントデータとリンクする方法を示していますadId。

Example JSON メッセージ:

```
{
  "avails": [
    {
      "adBreakTrackingEvents": [],
      "ads": [
        {
          "adId": "5",
          "adParameters": "",
          "adProgramDateTime": null,
          "adSystem": "2.0",
          "adTitle": "AD-caribbean2-15",
          "adVerifications": [],
          "companionAds": [],
          "creativeId": "1234",
          "creativeSequence": "2",
          "duration": "PT15S",
          "durationInSeconds": 15,
          "extensions": [],
          "mediaFiles": {
            "mediaFilesList": [],
            "mezzanine": ""
          },
          "skipOffset": null,
          "startTime": "PT30S",
          "startTimeInSeconds": 30,
          "trackingEvents": [
            {
              "beaconUrls": [
                "https://myServer/impression"
              ],
              "duration": "PT15S",
              "durationInSeconds": 15,
              "eventId": "5",
              "eventProgramDateTime": null,
              "eventType": "impression",
              "startTime": "PT30S",
              "startTimeInSeconds": 30
            }
          ],
          "vastAdId": ""
        }
      ]
    }
  ],
}
```

```
    "availId": "5",
    "availProgramDateTime": null,
    "duration": "PT15S",
    "durationInSeconds": 15,
    "meta": null,
    "nonLinearAdsList": [],
    "startTime": "PT30S",
    "startTimeInSeconds": 30
  }
],
"nextToken": "UFQ1TTM0Ljk2N1NfMjAyMi0xMS0xOFQwNDozMzo1Mi4yNDUxOTdaXzE%3D",
"nonLinearAvails": []
}
```

AWS Elemental MediaTailor を使用して線形アセンブルストリームを作成する

AWS Elemental MediaTailor チャンネルアセンブリは、ライブコンテンツと混合した既存のビデオオンデマンド (VOD) コンテンツを使用してリニアストリーミングチャンネルを作成できるマニフェスト専用サービスです。オリジンサーバーから直接提供されるコンテンツセグメントに MediaTailor が関与することは一切ありません。その代わりに、MediaTailor はオリジンからマニフェストを取得し、それらを使用して、基盤となるコンテンツセグメントを参照するライブスライディングマニフェストウィンドウをアセンブルします。チャンネルアセンブリは、アセットからアセットへと移行する再生をスムーズにするために必要なメディアシーケンス番号といった事柄を追跡します。リニア状にアセンブルされたストリームは、マルチビットレートでエンコードされ、パッケージ化された既存の VOD コンテンツを使用することによって、低ランニングコストで作成されます。

チャンネルアセンブリのリニアストリームは、SCTE-35 マーカーでコンテンツを調節することなく、プログラムに広告ブレイクを挿入することによって、簡単に収益化できます。チャンネルアセンブリは、MediaTailor 広告挿入サービス、または任意のサーバー側広告挿入サービスで使用できます。

チャンネルアセンブリの使用を開始するには、「[the section called “MediaTailor チャンネルアセンブリの使用開始”](#)」を参照してください。

Channel Assembly をコンテンツ配信ネットワーク (CDN) と統合する方法については、「[」](#)を参照してください[チャンネルアセンブリと CDN を使用して MediaTailor リニアチャンネルを構築する](#)。

トピック

- [ソースロケーションの使用](#)
- [チャンネルの使用](#)
- [チャンネルのスケジュールにプログラムを追加する](#)
- [チャンネルストリームにパーソナライズされた広告と広告ブレイクを挿入する](#)
- [チャンネルの再生のタイムシフト](#)

ソースロケーションの使用

ソースロケーションは、ソースコンテンツが保存されているオリジンサーバーを表します。ソースの場所は、Amazon S3、標準ウェブサーバー、Amazon CloudFront などのコンテンツ配信ネットワー

ク (CDN)、または などのパッケージングオリジンです AWS Elemental MediaPackage。MediaTailor はソースロケーションからコンテンツマニフェストを取得し、それらを使用してチャンネルのリニアストリームをアSEMBルします。

このトピックでは、AWS Elemental MediaTailor コンソールを使用してソースの場所を作成および削除する方法と、VOD ソースを操作する方法について説明します。

トピック

- ソースロケーションの作成
- ソースロケーション用の認証の設定
- VOD ソースの使用
- ライブソースの使用
- パッケージ設定の使用
- マニフェストキャッシュ

ソースロケーションの作成

次の手順では、MediaTailor コンソールを使用してソースロケーションを作成する方法について説明します。MediaTailor API を使用してソースロケーションを作成する方法については、API AWS Elemental MediaTailor リファレンスの[CreateSourceLocation](#)」を参照してください。

ソースロケーションを作成する

1. MediaTailor コンソール (<https://console.aws.amazon.com/mediatailor/>) を開きます。
2. ナビゲーションペインで、[Channel assembly] (チャネルアセンブリ) > [Source locations] (ソースロケーション) とクリックします。
3. ナビゲーションバーで、[Create source location] (ソースロケーションを作成) をクリックします。
4. [Source location configuration] (ソースロケーション設定) で、オリジンサーバーの名前とベース URL を入力します。
 - [Name] (名前): ソースロケーションの識別子 (my-orig など)。
 - [Base URL] (ベース URL): コンテンツが保存されているオリジンサーバーのプロトコルとベース URL (<https://111111111111.cloudfront.net> など)。URL は標準の HTTP URL 形式で、プレフィックスを <http://> または <https://> にする必要があります。

必要に応じて、ソースロケーションが Amazon S3 バケットで、Amazon S3 アクセス認証に署名バージョン 4 を使用する場合は、Amazon S3 認証に SigV44 を使用するを選択します。Amazon S3 AWS Amazon S3 詳細については、「[ソースロケーション用の認証の設定](#)」を参照してください。

5.

[Access configuration] (アクセス設定) では、オプションでソースロケーションの認証を設定します。

- [Access type] (アクセスタイプ): ソースロケーションのオリジンに保存されているコンテンツにアクセスするために MediaTailor が使用する認証タイプを選択します。
- [SigV4 for Amazon S3] (Amazon S3 向けの SigV4) - MediaTailor は、Amazon 署名バージョン 4 (SigV4) を使用してオリジンに対するリクエストを承認します。詳細については、「[the section called “SigV4 を使用した Amazon S3 へのリクエストの認証”](#)」を参照してください。
- Secrets Manager アクセストークン認証 - MediaTailor は、ユーザーが作成、所有、管理する Secrets Manager と AWS KMS カスタマーマネージドキーを使用して、MediaTailor とオリジン間のアクセストークン認証を容易にします。[Secrets Manager access token authentication] (Secrets Manager アクセストークン認証) の設定方法については、「[the section called “AWS Secrets Manager アクセストークン認証の使用”](#)」を参照してください。
- [Header name] (ヘッダー名) - HTTP ヘッダー名を指定します。MediaTailor は、オリジンにアクセストークンを送信するためにコンテンツマニフェストリクエストの HTTP ヘッダーを使用します。ヘッダー名は、x-amz- または x-amzn- で始まらないことを条件として、任意の名前を使用できます。[MediaPackage CDN 認可](#)と統合する場合、ヘッダー値は である必要がありますX-MediaPackage-CDNIdentifier。
- [Secret string key] (シークレット文字列キー) - Secrets Manager のシークレットで指定した SecretString キーです。例えば、SecretString に {"MyHeaderName": "11111111-2222-3333-4444-111122223333"} のようなキーと値のペアが含まれている場合は、MyHeaderName がこのフィールドに入力する SecretString キーになります。
- [Secret ARN] (シークレット ARN) - アクセストークンが保持されているシークレットの ARN です。ステップバイステップガイドについては、「[ステップ 2: AWS Secrets Manager シークレットを作成する](#)」を参照してください。

6. [Segment delivery server configuration (セグメント配信サーバー設定) では、オプションでコンテンツセグメントを配信するためのサーバーを設定します。

- [Use a default segment delivery server] (デフォルトのセグメント配信サーバーを使用する): CDN など、コンテンツセグメントの配信に使用されるサーバーのベース URL を入力します。コンテンツセグメントの配信にソースロケーションサーバーとは異なるサーバーを使用する場合は、[Default segment host name] (デフォルトのセグメントホスト名) を設定します。例えば、[Base HTTP URL] (ベース HTTP URL) (MediaTailor がマニフェストへのアクセスに使用する URL) と [Default Segment Base URL] (デフォルトのセグメントベース URL) (プレイヤーがコンテンツセグメントへのアクセスに使用する URL) に異なる CDN 設定を使用することで、プレイヤーからのオリジンマニフェストへのアクセスを制限することができます。値を入力しない場合、MediaTailor はデフォルトでセグメント配信にソースロケーションサーバーを使用します。
- 名前付きセグメント配信サーバーを使用する: デフォルトのセグメント配信サーバーを設定している場合は、追加のセグメント配信サーバーを設定することもできます。それぞれに一意の名前とベース URL が必要です。ベース URL は完全な HTTP URL にすることも、 /some/path/ のような相対パスにすることもできます。名前は、MediaTailor がコンテンツセグメントのリクエストを受信するときに使用するサーバーを識別するために使用されます。リクエストにヘッダーが含まれ X-MediaTailor-SegmentDeliveryConfigurationName、ヘッダーの値が名前と一致する場合、対応するベース URL を使用してコンテンツを提供します。ヘッダーがリクエストに含まれていない場合、または名前と一致しない場合は、デフォルトのセグメント配信サーバーが使用されます。

7. [Create source location] (ソースロケーションを作成) をクリックします。

8. ソースロケーションを追加するには、ステップ 2~6 を繰り返します。

ソースロケーション用の認証の設定

[Access configuration] (アクセス設定) を使用して、ソースロケーション用の認証を設定します。アクセス設定が有効になっている場合、MediaTailor は、リクエストが MediaTailor とオリジン間で承認される場合に限り、オリジンからソースマニフェストを取得します。アクセス設定は、デフォルトで無効になっています。

MediaTailor は以下の認証タイプをサポートしています。

- Amazon S3 認証向けの Sigv4
- AWS Secrets Manager アクセストークン
- MediaPackage バージョン 2 (v2) 認証用の SigV4

この章では、SigV4 for Amazon S3、MediaPackage v2、および AWS Secrets Manager アクセストークンをソースロケーション認証に使用する方法について説明します。

詳細については、該当するトピックを選択してください。

トピック

- [SigV4 を使用した Amazon S3 へのリクエストの認証](#)
- [MediaPackage バージョン 2 での SigV4 の使用](#)
- [AWS Secrets Manager アクセストークン認証の使用](#)

SigV4 を使用した Amazon S3 へのリクエストの認証

Amazon S3 の署名バージョン 4 (SigV4) は、HTTPS 経由で Amazon S3 へのリクエストを認証するために使用される署名プロトコルです。Amazon S3 に SigV4 を使用する場合、MediaTailor はオリジンとして使用される Amazon S3 バケットへの HTTPS リクエストに署名付き認可ヘッダーを含めます。署名付きの認可ヘッダーが有効である場合は、オリジンがリクエストに対応します。有効でない場合は、リクエストが失敗します。

SigV4 for の一般的な情報については AWS Key Management Service、Amazon S3 API リファレンスの「[リクエストの認証 \(AWS 署名バージョン 4\)](#)」トピックを参照してください。

Note

MediaTailor は常に SigV4 を使用してこれらのオリジンへのリクエストに署名します。

要件

ソースロケーションのために Amazon S3 認証用の SigV4 をアクティブ化する場合は、以下の要件を満たす必要があります。

- IAM で `mediatailor.amazonaws.com` プリンシパルアクセスを許可することで、MediaTailor が Amazon S3 バケットにアクセスすることを許可する必要があります。IAM でアクセスを設定する方法については、AWS Identity and Access Management 「ユーザーガイド」の「[アクセス管理](#)」を参照してください。
- `mediatailor.amazonaws.com` サービスプリンシパルには、VOD ソースパッケージ設定によって参照されるすべてのマルチバリエーションプレイリストを読み取るアクセス許可が必要です。

- API の呼び出し元には、MediaTailor VOD ソースパッケージ設定によって参照されるすべてのマルチバリエーションプレイリストを読み取るための s3:GetObject IAM アクセス許可が必要です。
- MediaTailor のソースロケーションのベース URL が、Amazon S3 の仮想ホスティング形式のリクエスト URL 形式に従っていること。例えば、`https://bucket-name.s3.Region.amazonaws.com/key-name` などです。Amazon S3 の仮想ホスティング形式のアクセスについては、「[仮想ホスティング形式のリクエスト](#)」を参照してください。

MediaPackage バージョン 2 での SigV4 の使用

MediaPackage v2 の署名バージョン 4 (SigV4) は、HTTP 経由で MediaPackage v2 へのリクエストを認証するために使用される署名プロトコルです。MediaPackage v2 に SigV4 を使用する場合、MediaTailor はオリジンとして使用される MediaPackage v2 エンドポイントへの HTTP リクエストに署名付き認可ヘッダーを含めます。署名付きの認可ヘッダーが有効である場合は、オリジンがリクエストに対応します。有効でない場合は、リクエストが失敗します。

MediaPackage v2 用 SigV4 の一般的な情報については、MediaPackage v2 API リファレンスの「[リクエストの認証 \(AWS 署名バージョン 4\)](#)」トピックを参照してください。

要件

ソースロケーションで MediaPackage v2 認証用の SigV4 をアクティブ化する場合は、次の要件を満たす必要があります。

- エンドポイントのオリジンアクセスポリシーで `mediatailor.amazonaws.com` プリンシパルアクセスを許可することで、MediaTailor が MediaPackage v2 エンドポイントにアクセスすることを許可する必要があります。
- MediaTailor のソースロケーションベース URL は MediaPackage v2 エンドポイントである必要があります。
- API の呼び出し元には、MediaTailor ソースパッケージング設定によって参照されるすべてのマルチバリエーションプレイリストを読み取るための `mediapackagev2:GetObject` IAM アクセス許可が必要です。MediaTailor

AWS Secrets Manager アクセストークン認証の使用

MediaTailor は、Secrets Manager アクセストークン認証をサポートします。AWS Secrets Manager アクセストークン認証では、MediaTailor は AWS Key Management Service (AWS KMS) カスタマーマネージドキー AWS Secrets Manager と、オリジンへのリクエストを認証するために作成、所有、管理するシークレットを使用します。

このセクションでは、Secrets Manager アクセストークン認証の仕組みについて説明し、Secrets Manager アクセストークン認証の設定方法に関するステップバイステップの情報を提供します。Secrets Manager アクセストークン認証は、で AWS マネジメントコンソール 操作することも、AWS APIsを使用してプログラムで操作することもできます。

トピック

- [AWS Secrets Manager アクセストークン認証の設定](#)
- [CDN 認可を使用する MediaPackage エンドポイントとの統合](#)
- [MediaTailor の Secrets Manager アクセストークン認証の仕組み](#)

AWS Secrets Manager アクセストークン認証の設定

AWS Secrets Manager アクセストークン認証を使用する場合は、次の手順を実行します。

1. [AWS Key Management Service カスタマーマネージドキーを作成する](#)。
2. [AWS Secrets Manager シークレットを作成する](#)。シークレットには、暗号化されたシークレット値として Secrets Manager に保存されているアクセストークンが含まれています。MediaTailor は、AWS KMS カスタマーマネージドキーを使用してシークレット値を復号します。
3. Secrets Manager アクセストークン認証を使用するように AWS Elemental MediaTailor ソースの場所を設定します。

以下のセクションは、AWS Secrets Manager アクセストークン認証の設定方法に関するステップバイステップガイドです。

トピック

- [ステップ 1: AWS KMS 対称カスタマーマネージドキーを作成する](#)
- [ステップ 2: AWS Secrets Manager シークレットを作成する](#)
- [ステップ 3: アクセストークン認証で MediaTailor のソースロケーションを設定する](#)

ステップ 1: AWS KMS 対称カスタマーマネージドキーを作成する

を使用して AWS Secrets Manager、シークレットに SecretString 保存された の形式でアクセストークンを保存します。SecretString は、ユーザーが作成、所有、および管理する AWS KMS 対称カスタマーマネージドキーを使用することで暗号化されます。MediaTailor は、対称カスタマーマネージドキーを使用して、権限のあるシークレットへのアクセスを容易にし、シークレット値を暗号化および復号化します。

カスタマーマネージドキーを使用することで、以下のようなタスクを実行できます。

- キーポリシーの策定と維持
- IAM ポリシーとグラントの策定と維持
- キーポリシーの有効化と無効化
- 暗号化キーマテリアルのローテーション
- タグの追加

Secrets Manager が AWS KMS を使用してシークレットを保護する方法については、「[AWS Key Management Service デベロッパーガイド](#)」の「[AWS Secrets Manager が AWS KMS を使用する](#)方法」トピックを参照してください。

カスタマーマネージドキーの詳細については、AWS Key Management Service デベロッパーガイドの「[カスタマーマネージドキー](#)」を参照してください。

Note

AWS KMS カスタマーマネージドキーの使用には 料金が適用されます。料金の詳細については、[AWS Key Management Service](#) 「[料金表](#)」ページを参照してください。

対称カスタマーマネージドキーは、を使用する AWS マネジメントコンソール か AWS KMS APIs AWS KMS を使用してプログラムで作成できます。

対称カスタマーマネージドキーを作成する

AWS Key Management Service デベロッパーガイドにある[対称カスタマーマネージドキーの作成](#)ステップを実行します。

キーの Amazon リソースネーム (ARN) を書き留めておきます。これは[ステップ 2: AWS Secrets Manager シークレットを作成する](#) で必要になります。

暗号化コンテキスト

暗号化コンテキストは、データに関する追加のコンテキスト情報が含まれたキーバリューペアのオプションのセットです。

Secrets Manager は、SecretString を暗号化および復号化するときに[暗号化コンテキスト](#)を含めます。暗号化コンテキストにはシークレット ARN が含まれており、これは暗号化をその特定のシー

クレットに制限します。追加のセキュリティ対策として、MediaTailor はユーザーに代わって AWS KMS 権限を作成します。MediaTailor は、Secrets Manager 暗号化コンテキストに含まれるシークレット ARN SecretStringに関連付けられた のみを復号できるようにする [GrantConstraints](#) オペレーションを適用します。

Secrets Manager が暗号化コンテキストを使用する方法については、「AWS Key Management Service デベロッパーガイド」の「[暗号化コンテキスト](#)」トピックを参照してください。

キーポリシーの設定

キーポリシーは、カスタマーマネージドキーへのアクセスを制御します。すべてのカスタマーマネージドキーには、キーポリシーが 1 つ必要になります。このポリシーには、そのキーを使用できるユーザーとその使用方法を規定するステートメントが含まれています。カスタマーマネージドキーを作成するときは、デフォルトのキーポリシーを使用できます。詳細については、AWS Key Management Service デベロッパーガイドの「[AWS KMSの認証とアクセスコントロール](#)」を参照してください。

MediaTailor ソースロケーションリソースでカスタマーマネージドキーを使用するには、[CreateSourceLocation](#) または [UpdateSourceLocation](#) を呼び出す IAM プリンシパルに、次の API オペレーションを使用するアクセス許可を付与する必要があります。

- kms:CreateGrant - カスタマーマネージドキーにグラントを追加します。MediaTailor はカスタマーマネージドキーに対するグラントを作成します。これは、アクセストークン認証が設定されたソースロケーションの作成と更新に MediaTailor がこのキーを使用することを可能にします。[での Grants の使用の詳細については AWS KMS](#)、「AWS Key Management Service デベロッパーガイド」を参照してください。

これは、MediaTailor が以下を実行できるようにします。

- [GetSecretValue](#) Decrypt を呼び出すときに Secrets Manager シークレットを正常に取得できるように、 を呼び出します。
- ソースロケーションが削除された、またはシークレットへのアクセス権が取り消されたときにグラントを廃止するために RetireGrant を呼び出す。

以下は、MediaTailor に追加できるポリシーステートメントの例です。

```
{
  "Sid": "Enable MediaTailor Channel Assembly access token usage for the
MediaTailorManagement IAM role",
  "Effect": "Allow",
```



```
"Principal": {
  "AWS": "arn:aws:iam::account number:role/MediaTailorManagement"
},
"Action": "kms:CreateGrant",
"Resource": "*",
"Condition": {
  "StringEquals": {
    "kms:ViaService": "mediatailor.region.amazonaws.com"
  }
}
}
```

ポリシーでのアクセス許可の指定とキーアクセスのトラブルシューティングの詳細については、「AWS Key Management Service デベロッパーガイド」の「[のグラン AWS KMS ト](#)」を参照してください。

ステップ 2: AWS Secrets Manager シークレットを作成する

Secrets Manager を使用して、AWS KMS カスタマーマネージドキーで暗号化 SecretString された の形式でアクセストークンを保存します。MediaTailor は、SecretString の復号化にこのキーを使用します。Secrets Manager が AWS KMS を使用してシークレットを保護する方法については、「AWS Key Management Service デベロッパーガイド」の「[AWS Secrets Manager が AWS KMS を使用する方法](#)」トピックを参照してください。

をソースロケーションオリジン AWS Elemental MediaPackage として使用し、MediaTailor Secrets Manager アクセストークン認証を使用する場合は、「」の順に従います [the section called “CDN 認可を使用する MediaPackage エンドポイントとの統合”](#)。

Secrets Manager API を使用して、AWS マネジメントコンソール またはプログラムで Secrets Manager シークレットを作成できます APIs 。

シークレットを作成する

「AWS Secrets Manager ユーザーガイド」の「[でシークレットを作成および管理する AWS Secrets Manager](#)」の順に従います。

シークレットを作成するときは、以下の考慮事項に留意してください。

- [KmsKeyId](#) は、ステップ 1 で作成したカスタマーマネージドキーのキー [ARN](#) である必要があります。
- [SecretString](#) を指定する必要があります。SecretString は、アクセストークンが含まれる、キーと値が含まれた有効な JSON オブジェクトである必要があります。例えば、

`{"MyAccessTokenIdentifier":"112233445566"}` などです。値の長さは 8～128 文字にしてください。

アクセストークン認証を使用してソースロケーションを設定するときは、SecretString キーを指定します。MediaTailor は、SecretString に保存されているアクセストークンの検索と取得にこのキーを使用します。

シークレット ARN と SecretString キーを書き留めておきます。これらは、アクセストークン認証を使用するようにソースロケーションを設定するときに使用します。

リソースベースのシークレットポリシーのアタッチ

MediaTailor がシークレット値にアクセスできるようにするには、シークレットにリソースベースのポリシーをアタッチする必要があります。詳細については、[「ユーザーガイド」の「Secrets Manager シークレットにアクセス許可ポリシーをアタッチする」](#)を参照してください。AWS Secrets Manager

以下は、MediaTailor に追加できるポリシーステートメントの例です。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "mediatailor.amazonaws.com"
      },
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "arn:aws:secretsmanager:us-east-1:111122223333:secret:secret-name"
    }
  ]
}
```


ステップ 3: アクセストークン認証で MediaTailor のソースロケーションを設定する

Secrets Manager アクセストークン認証は、を使用する AWS マネジメントコンソール か MediaTailor APIs を使用してプログラムで設定できます。

Secrets Manager アクセストークン認証でソースロケーションを設定する

「AWS Elemental MediaTailor ユーザーガイド」の[Access configuration](#)「」の手順に従います。

CDN 認可を使用する MediaPackage エンドポイントとの統合

をソースロケーションオリジン AWS Elemental MediaPackage として使用すると、MediaTailor は CDN 認可を使用する MediaPackage エンドポイントと統合できます。

CDN 認可を使用する MediaPackage エンドポイントとの統合には、以下の手順を実行します。

MediaPackage に統合する

1. まだの場合は、AWS Elemental MediaPackage 「ユーザーガイド」の「[CDN 認可の設定](#)」のステップを完了します。
2. [the section called “ステップ 1: AWS KMS 対称カスタマーマネージドキーを作成する”](#) のステップを完了します。
3. MediaPackage CDN 認可のセットアップ時に作成したシークレットを変更します。シークレットは、以下の値で変更してください。
 - [the section called “ステップ 1: AWS KMS 対称カスタマーマネージドキーを作成する”](#) で作成したカスタマーマネージドキー ARN で KmsKeyId を更新します。
 - (オプション) SecretString については、UUID を新しい値にローテーションする、または既存の暗号化されたシークレットを使用する (それが {"MediaPackageCDNIdentifier": "112233445566778899"} といった標準 JSON 形式でのキーと値のペアである場合のみ) ことができます。
4. [the section called “リソーススペースのシークレットポリシーのアタッチ”](#) のステップを完了します。
5. 「[the section called “ステップ 3: アクセストークン認証で MediaTailor のソースロケーションを設定する”](#)」の各ステップを実行します。

MediaTailor の Secrets Manager アクセストークン認証の仕組み

アクセストークン認証を使用するようにソースロケーションを作成または更新すると、MediaTailor はオリジンからのソースコンテンツマニフェストをリクエストするときに、HTTP ヘッダーにアクセストークンを含めます。

以下は、MediaTailor がソースロケーションオリジン認証に Secrets Manager アクセストークン認証を使用する方法の概要です。

1. アクセストークン認証を使用する MediaTailor ソースロケーションを作成または更新すると、MediaTailor は Secrets Manager に [DescribeSecret](#) リクエストを送信して、シークレットに関連付けられた AWS KMS キーを決定します。シークレット ARN はソースロケーションのアクセス設定に含めます。
2. MediaTailor は、SecretString に保存されているアクセストークンへのアクセスと復号化にカスタマーマネージドキーを使用できるように、カスタマーマネージドキーに対する[グラント](#)を作成します。グラント名は MediaTailor-SourceLocation-*your AWS ##### ID-source location name* になります。

グラントへのアクセス権の取り消し、またはカスタマーマネージドキーへの MediaTailor のアクセス権の削除は、いつでも実行できます。詳細については、AWS Key Management Service API リファレンスの「[RevokeGrant](#)」を参照してください。

3. VOD ソースが作成もしくは更新される、またはプログラムで使用されると、MediaTailor は、ソースロケーションに対して HTTP リクエストを実行して、そのソースロケーション内の VOD ソースに関連付けられたソースコンテンツマニフェストを取得します。アクセストークンが設定されているソースロケーションにその VOD ソースが関連付けられている場合は、アクセストークンが HTTP ヘッダー値としてリクエストに含まれます。

VOD ソースの使用

VOD ソースは、動画やポッドキャストのエピソードなど、ソースロケーションに追加する単一のコンテンツを表します。ソースロケーションに 1 つ、または複数の VOD ソースを追加してから、チャンネルの作成後に各 VOD ソースをプログラムに関連付けます。

各 VOD ソースには、少なくとも 1 つのパッケージ設定が必要です。パッケージ設定は、VOD ソースのパッケージ形式、マニフェストの場所、およびソースグループを指定します。チャンネルを作成するときは、パッケージ設定のソースグループを使用して、チャンネル上に対応する出力を作成します。例えば、ソースが HLS と DASH の 2 つの異なる形式でパッケージ化されている場合は、DASH 用と HLS 用に 1 つずつ、合計で 2 つのパッケージ設定を作成します。その後、パッケージ設定ごとに 1

つつ、合計で 2 つのチャンネル出力を作成します。各チャンネル出力は、再生リクエストに使用されるエンドポイントを提供します。このため、上記の例を使用すると、チャンネルは HLS 再生リクエスト用のエンドポイントと、DASH 再生リクエスト用のエンドポイントを提供することになります。

マニフェスト内の広告マーカのオフセットを自動的に検出する場合、各広告マーカはすべてのパッケージ設定で同じオフセットで表示され、期間は 0 である必要があります。HLS の場合、MediaTailor は DATERANGE と EXT-X-CUE-OUT タグを検出します。DASH の場合、HLS は各タグ内の最初のイベント EventStream タグを検出します。

次の例では、期間 0.0 の DATERANGE タグのため、広告時間枠の機会は 12,000 ミリ秒のオフセットで検出されます。オフセット 0ms の最初の DATERANGE タグは、期間が 10.0 であるため、検出されません。

```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-TARGETDURATION:6
#EXT-X-MEDIA-SEQUENCE:0
#EXT-X-DATERANGE:ID="1001",START-DATE="2021-09-16T23:51:05.249Z",DURATION=10.0,SCTE35-OUT=0xFC302500000003289800FFF01405000003E97FEFFE1D381BD8FE000DBBA00001010100000FD2B275
#EXTINF:6.000,
../../../719f911124e0495cbb067c91c1d6c298/1785a16ca14d4c2884781f25333f6766/index_1_0.ts
#EXTINF:6.000,
../../../719f911124e0495cbb067c91c1d6c298/1785a16ca14d4c2884781f25333f6766/index_1_1.ts
#EXT-X-DATERANGE:ID="1001",START-DATE="2021-09-16T23:51:05.249Z",DURATION=0.0,SCTE35-OUT=0xFC302500000003289800FFF01405000003E97FEFFE1D381BD8FE000DBBA00001010100000FD2B275
#EXTINF:6.000,
../../../719f911124e0495cbb067c91c1d6c298/1785a16ca14d4c2884781f25333f6766/index_1_2.ts
```

次の例では、EXT-X-CUE-OUT タグの再生時間が 0 で、タグが直後に続くため、広告時間枠の機会は 0 ミリ秒のオフセットで検出されます EXT-X-CUE-IN。2 番目の EXT-X-CUE-OUT/EXT-X-CUE-IN ペアは、期間が 10 であるため検出されません。

```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-TARGETDURATION:6
#EXT-X-MEDIA-SEQUENCE:0
#EXT-X-CUE-OUT:0
#EXT-X-CUE-IN
#EXTINF:6.000,
../../../719f911124e0495cbb067c91c1d6c298/1785a16ca14d4c2884781f25333f6766/index_1_0.ts
#EXTINF:6.000,
../../../719f911124e0495cbb067c91c1d6c298/1785a16ca14d4c2884781f25333f6766/index_1_1.ts
```

```
#EXT-X-CUE-OUT:10
...
#EXT-X-CUE-IN
#EXTINF:6.000,
../.../719f911124e0495cbb067c91c1d6c298/1785a16ca14d4c2884781f25333f6766/index_1_2.ts
```

次の例では、の最初のイベントが PT0.000S から始まる期間に EventStream 発生するため、広告時間枠の機会は 0 ミリ秒のオフセットで検出されます。Event の 2 番目の EventStream は検出されません。

```
<Period start="PT0.000S" id="9912561" duration="PT29.433S">
<EventStream timescale="90000" schemeIdUri="urn:scte:scte35:2013:xml">
<Event duration="0">
  <scte35:SpliceInfoSection protocolVersion="0" ptsAdjustment="1241950593" tier="4095">
    <scte35:SpliceInsert spliceEventId="99" spliceEventCancelIndicator="false"
outOfNetworkIndicator="true" spliceImmediateFlag="false" uniqueProgramId="1"
availNum="1" availsExpected="1">
      <scte35:Program><scte35:SpliceTime ptsTime="3552273000"/></scte35:Program>
      <scte35:BreakDuration autoReturn="true" duration="2700000"/>
    </scte35:SpliceInsert>
  </scte35:SpliceInfoSection>
</Event>
<Event duration="0">
  <scte35:SpliceInfoSection protocolVersion="0" ptsAdjustment="1241950593" tier="4095">
    <scte35:SpliceInsert spliceEventId="99" spliceEventCancelIndicator="false"
outOfNetworkIndicator="true" spliceImmediateFlag="false" uniqueProgramId="1"
availNum="1" availsExpected="1">
      <scte35:Program><scte35:SpliceTime ptsTime="3552273000"/></scte35:Program>
      <scte35:BreakDuration autoReturn="true" duration="2700000"/>
    </scte35:SpliceInsert>
  </scte35:SpliceInfoSection>
</Event>
</EventStream>
...
</Period>
```

ソースロケーションへの VOD ソースの追加

以下は、MediaTailor コンソールを使用して VOD ソースをソースロケーションに追加し、パッケージ設定を行う方法を説明する手順です。MediaTailor API を使用して VOD ソースを追加する方法については、API AWS Elemental MediaTailor リファレンスの [CreateVodSource](#) を参照してください。

⚠ Important

VOD ソースを追加する前に、それらが以下の要件を満たしていることを確認してください。

- すべてのソースバリエーションの長さが同じであること (ソースマニフェストによって規定されている長さ)。
- パッケージ設定内の各ソースに同じ数の子ストリームがあること。

これらの要件のため、タイトルごとまたは自動化された ABR はサポートされません。これらのエンコーディングメソッドは、さまざまなマニフェストの長さと子ストリームを生成する可能性があるためです。

エンコードされたソースがこれらの要件を満たすことを確実にするため、最小セグメント長が含まれたエンコーディングテンプレートの使用をお勧めします。

VOD ソースをソースロケーションに追加する

1. MediaTailor コンソール (<https://console.aws.amazon.com/mediatailor/>) を開きます。
2. ナビゲーションペインで、[Channel assembly] (チャネルアセンブリ) > [Source locations] (ソースロケーション) とクリックします。
3. [Source locations] (ソースロケーション) ペインで、[ソースロケーションを作成する](#) の手順で作成したソースロケーションを選択します。
4. [Add VOD source] (VOD ソースを追加) をクリックします。
5. [VOD source details] (VOD ソースの詳細) で、VOD ソースの名前を入力します。
 - [Name] (名前): my-example-video などの VOD ソースの識別子です。
6. [Package configurations] (パッケージ設定) > *source-group-name* と進み、パッケージ設定に関する情報を入力します。

i Note

ソースのパッケージ設定の継続時間は、すべて同じ (ソースのマニフェストによって規定されている時間) にする必要があります。また、パッケージ設定内のすべてのソースで、子ストリームの数が同じである必要もあります。これらの要件を満たすため、アセットにはエンコードテンプレートを使用することをお勧めします。最小セグメント長が 1 秒のエンコーディングテンプレートの使用が推奨されます。タイトルごと、また

は自動アダプティブビットレート (ABR) ストリーミングはこれらの要件に違反するため、MediaTailor ではこれらのエンコードメソッドがサポートされません。

- [Source group] (ソースグループ): HLS-4K など、このパッケージ設定を説明するソースグループ名を入力します。この名前を書き留めておきます。これは、チャンネルの出力を作成するときに参照します。詳細については、「[チャンネルの出力でのソースグループの使用](#)」を参照してください。
- [Type] (タイプ): この設定のパッケージ形式を選択します。MediaTailor は、HLS と DASH をサポートします。
- [Relative path] (相対パス): ソースロケーションの ベース HTTP URL からマニフェストへの相対パスです。例えば、/my/path/index.m3u8 などです。

Note

MediaTailor は、親マルチバリエーションプレイリストに含まれるすべてのクローズドキャプションと子ストリームを自動的にインポートします。ソースのレンディション (DASH) またはバリエーションストリーム (HLS) ごとに個別のパッケージ設定を作成する必要はありません。

パッケージ設定の詳細については、「[パッケージ設定の使用](#)」を参照してください。

7. [Add VOD source] (VOD ソースを追加) をクリックします。

VOD ソースを追加する場合は、手順のステップ 4~7 を繰り返します。

ライブソースの使用

ライブソースは、ライブサッカーゲームやニュースブロードキャストなどの単一のライブストリームを表し、ソースの場所に追加します。チャンネルを作成したら、ソースの場所に 1 つ以上のライブソースを追加し、各ライブソースをプログラムに関連付けます。

MediaTailor は、以下のタイプのリニアチャンネルアセンブリをサポートしています。

- VOD-to-live コンテンツを含むチャンネルの VOD ソース
- VODlive-to-liveコンテンツと混在した Live-to-live コンテンツを含むチャンネルのライブソース VOD-to-live

VOD-to-live コンテンツの例は、VOD アセットのライブラリをライブストリームにアセンブルするチャンネルです。Livelive-to-live コンテンツと VOD-to-live コンテンツが混在する例の 1 つは、夜間のニュースイベントや事前にスケジュールされたライブスポーツイベントを除き、主に VOD コンテンツを表示するチャンネルです。Livelive-to-live コンテンツと VOD-to-live コンテンツを組み合わせたもう 1 つの例は、時刻によって異なるオリジンを持つすべての live-to-live チャンネルです。

ライブソースを使用して、主に国のプログラミングを表示するリージョンチャンネルをセットアップできますが、リージョンプログラミングオーバーライドが含まれ、VOD コンテンツが混在しています。そのためには、国内コンテンツに対してエンコーダーとパッケージャーのペアを 1 つ実行し、それらのリージョンが稼働しているときにリージョンエンコーダーを実行します。次に、リージョンチャンネルアセンブリチャンネルを作成し、それぞれに独自のスケジュールを設定します。これにより、ビューワーは必要に応じて切り替えることができます。この設定は、エンコード/パッケージ化のコストを最小限に抑えるのに役立ちます。

各ライブソースには、少なくとも 1 つのパッケージ設定が必要です。パッケージ設定では、ライブソースのパッケージ形式、マニフェストの場所、ソースグループを指定します。チャンネルを作成するときは、パッケージ設定のソースグループを使用して、チャンネル上に対応する出力を作成します。例えば、ソースが HLS と DASH の 2 つの異なる形式でパッケージ化されている場合は、DASH 用と HLS 用に 1 つずつ、合計で 2 つのパッケージ設定を作成します。次に、パッケージ設定ごとに 1 つずつ、2 つのチャンネル出力を作成します。各チャンネル出力は、再生リクエストに使用されるエンドポイントを提供します。この例では、チャンネルは HLS 再生リクエスト用のエンドポイントと DASH 再生リクエスト用のエンドポイントを提供します。

ライブソースを使用するための一般的な要件

ライブソースを使用する場合、コンテンツは次の一般的な要件に従う必要があります。

- HLS ライブソース - マニフェストウィンドウの最初のセグメントとすべての不連続性に #EXT-X-PROGRAM-DATE-TIME タグを指定する必要があります。
- HLS - 広告マーカールとして設定する必要があります DATERANGE。
- ソースマニフェストウィンドウ - MediaTailor Channel Assembly チャンネルのマニフェストウィンドウと少なくとも同じ期間を持つマニフェストウィンドウを使用することをお勧めします。ベストプラクティスとして、チャンネルアセンブリチャンネルのマニフェストウィンドウよりも 30 秒以上のマニフェストウィンドウ期間を使用することを検討してください。
- ターゲット期間を既存のソースの期間と一致させます。
- 子プレイリストの数を既存のソースの数と一致させます。

Configurations

ライブソースワークフローの一部として他の AWS Elemental メディアサービスを使用する場合は、MediaPackage 設定をセットアップするときにベストプラクティスに従うことをお勧めします。次の表は、使用するストリーミング標準に基づいて MediaPackage 設定を構成する方法を示しています。

ライブソースの MediaPackage 設定

標準	設定	値	必要性	注意事項
HLS	エンドポイントタイプ	Apple HLS	CMAF を使用しない場合は必須です	HLS ts AWS Elemental MediaConvert ジョブを照合するには
HLS	エンドポイントタイプ	CMAF	Apple HLS を使用しない場合は必須です	HLS mp4 AWS Elemental MediaConvert ジョブを照合するには
HLS	ProgramDateTimeIntervalSeconds	1	必須	不連続な場合に再生の問題を防ぐために、各セグメント#EXT-X-PROGRAM-DATE-TIME で指定する必要があります。
HLS	PlaylistWindowSeconds	チャネルアセンブリマニフェストウィンドウより 30 秒長い	必須	

標準	設定	値	必要性	注意事項
HLS	AdMarkers	DATERANGE	広告マーカースを通過するときに必要	
HLS	IncludeIframeOnlyStream	Disabled	推奨	
DASH	ManifestLayout	FULL	推奨	
DASH	SegmentTemplateFormat	NUMBER_WITH_TIMELINE、または TIME_WITH_TIMELINE	推奨	NUMBER_WITH_DURATION はサポートされていません。
DASH	ManifestWindowSeconds	チャネルアセンブリマニフェストウィンドウより 30 秒長い	必須	
DASH	PeriodTriggers	ADS	広告マーカースを通過するときに必要	

ソースロケーションへのライブソースの追加

次の手順では、MediaTailor コンソールを使用してライブソースをソースの場所に追加し、パッケージ設定を設定する方法について説明します。MediaTailor API を使用してライブソースを追加する方法については、API AWS Elemental MediaTailor リファレンスの [CreateLiveSource](#) を参照してください。

⚠ Important

ライブソースを追加する前に、パッケージ設定内で各ソースに同じ数の子ストリームがあることを確認してください。

ライブソースをソースロケーションに追加するには

1. MediaTailor コンソール (<https://console.aws.amazon.com/mediatailor/>) を開きます。
2. ナビゲーションペインで、[Channel assembly] (チャネルアセンブリ) > [Source locations] (ソースロケーション) とクリックします。
3. [Source locations] (ソースロケーション) ペインで、[ソースロケーションを作成する](#) の手順で作成したソースロケーションを選択します。
4. ライブソースタブで、ライブソースの追加を選択します。
5. ライブソースの詳細で、ライブソースの名前を入力します。
 - 名前: my-example-video など、ライブソースの識別子。
6. [Package configurations] (パッケージ設定) > *source-group-name* と進み、パッケージ設定に関する情報を入力します。

i Note

パッケージ設定内では、すべての VOD ソースとライブソースに同じ数の子ストリームが必要です。ソースストリームは同じ方法で設定することをお勧めします。

- [Source group] (ソースグループ): HLS-4K など、このパッケージ設定を説明するソースグループ名を入力します。この名前を書き留めておきます。これは、チャネルの出力を作成するときに参照します。詳細については、「[チャネルの出力でのソースグループの使用](#)」を参照してください。
- [Type] (タイプ): この設定のパッケージ形式を選択します。MediaTailor は、HLS と DASH をサポートします。
- [Relative path] (相対パス): ソースロケーションの ベース HTTP URL からマニフェストへの相対パスです。例えば、/my/path/index.m3u8 などです。

Note

MediaTailor は、親マルチバリエーションプレイリストに含まれるすべてのクローズドキャプションと子ストリームを自動的にインポートします。ソースのレンディション (DASH) またはバリエーションストリーム (HLS) ごとに個別のパッケージ設定を作成する必要はありません。

パッケージ設定の詳細については、「[パッケージ設定の使用](#)」を参照してください。

7. ライブソースの追加 を選択します。

さらにライブソースを追加する場合は、手順のステップ 4~6 を繰り返します。

パッケージ設定の使用

パッケージ設定は、異なるデバイスでの再生に必要なさまざまなパッケージ特性が含まれたソースを表します。例えば、DRM を使用する HLS、セグメントタイムラインのアドレス指定を使用する DASH、および CMAF セグメントを使用する HLS の 3 つのパッケージ化された形式を持つソースがある場合があります。

チャネルアセンブリはソースを再パッケージ化しません。所定のソースに複数のパッケージ化された形式を含める場合は、パッケージ化された形式のそれぞれをソースロケーションで利用可能にして、パッケージ化された形式それぞれへのパスを指定する必要があります。

各パッケージ設定オブジェクトには、以下が含まれている必要があります。

- [Relative path] (相対パス) - ソースロケーションに相対する、ソースのパッケージ化された形式へのフルパス。例えば、/my/path/index.m3u8 などです。
- [Source group] (ソースグループ) - パッケージ設定をチャネルの出力に関連付けるために使用されるソースグループの名前。
- [Type] (タイプ) - HLS または DASH。

チャネルを作成したら、チャネルの出力に使用する各ソースグループを宣言する必要もあります。

マニフェストキャッシュ

MediaTailor は、ソースプレイリストを定期的かつ日和見的にキャッシュして、チャンネルアセンブリのパフォーマンスと信頼性を向上させます。キャッシュされたバージョンは、ソースの場所のオリジナルバージョンと比較して古くなることがあります。MediaTailor がキャッシュされたバージョンのソースを強制的に更新するには、[UpdateVodSource](#) を呼び出します。たとえば、埋め込みパスがソースで変更されたときにこの呼び出しを使用します。MediaTailor からのリクエストがほとんどない場合でも、常にソースの up-to-date をソースの場所で利用できるようにしてください。

チャンネルの使用

チャンネルは、ソースマニフェストをリニアストリームにアセンブルします。各チャンネルには、パッケージ設定に対応する 1 つ、または複数の出力が含まれています。

最初にチャンネルを作成し、次にプログラムを作成して VOD ソースとライブソースをチャンネルのスケジュールに追加します。各プログラムは VOD ソースまたはライブソースに関連付けられます。

トピック

- [MediaTailor コンソールを使用してチャンネルを作成する](#)
- [チャンネルの出力でのソースグループの使用](#)
- [MediaTailor コンソールを使用してチャンネルを削除する](#)

MediaTailor コンソールを使用してチャンネルを作成する

次の手順では、MediaTailor コンソールを使用してチャンネルを作成する方法について説明します。

チャンネルを作成する

1. MediaTailor コンソール (<https://console.aws.amazon.com/mediatailor/>) を開きます。
2. ナビゲーションペインで、[Channel assembly] (チャンネルアセンブリ) > [Channels] (チャンネル) とクリックします。
3. ナビゲーションバーで、[Create channel] (チャンネルを作成) をクリックします。
4. [Channel details] (チャンネルの詳細) で、チャンネルに関する詳細情報を入力します。
 - [Name] (名前): チャンネルの名前を入力します。

- 階層: 階層は、チャンネルがサポートする機能と、チャンネルの実行にかかるコストを決定します。料金の詳細については、[「チャンネルアセンブリの料金」ページ](#)を参照してください。MediaTailor は、次の階層をサポートしています。
- 基本 - 基本階層はリニア再生モードとループ再生モードの両方をサポートし、ライブソースはサポートしていません。
- 標準 - 標準階層はライブソースをサポートし、線形再生モードが必要です。

チャンネルの詳細で標準を選択すると、対象者の詳細で対象者を定義できます。これらのオーディエンスは、デフォルトのプログラムの audienceMedia を作成するときに programRules に使用されます。 audienceMedia

- [Add] (追加) を選択します。
 - テキストボックスに対象者名を入力します。1~32 文字の英数字を使用する必要があります。
 - [確認] を選択します。
 - [次へ] を選択します。
- [Playback mode] (再生モード): 再生モードは、チャンネルの再生動作を設定します。MediaTailor は、以下の再生モードをサポートします。
 - [Loop] (ループ) - スケジュール内のプログラムは、連続して無限ループで再生されます。スケジュール内の最後のプログラムが再生されると、再生は折り返して最初のプログラムに戻ります。再生は、チャンネルが停止されるまでループし続けます。
 - [Linear] (リニア) - スケジュール内の各プログラムは、連続して一度だけ再生されます。

5.

[Filler slate] (フィラスレート) には、スレートの場所を参照する [Source location name] (ソースロケーション名) と、スレートとして使用する [VOD source name] (VOD ソース名) を選択します。MediaTailor は、スレートを使用してスケジュール内のプログラム間のギャップを埋めます。スレートの継続時間がプログラム間のギャップの継続時間よりも短い場合、MediaTailor はスレートをループします。チャンネルが線形再生モードを使用している場合は、フィラスレートフィールドを に設定する必要があります。MediaTailor は、ループ再生モードのフィラスレートをサポートしていません。

6. [次へ] を選択します。

7. プログラムルールで対象者の詳細を指定します。

8. チャンネルの詳細で標準を選択すると、対象者の詳細で対象者を定義できます。これらのオーディエンスは、デフォルトのプログラムの audienceMedia を作成するときに programRules に使用されます。 audienceMedia

- 追加を選択し、テキストボックスに対象者を追加し、確認を選択します。

 Note

32 文字以下の英数字の名前を入力します。

- [Output type] (出力タイプ): チャンネルのストリーミング形式を選択します。DASH と HLS がサポートされています。
 - [Source group] (ソースグループ): [ソースロケーションへの VOD ソースの追加](#) の説明にあるように、パッケージ設定で作成したソースグループの名前を入力します。
9. [次へ] を選択してください。
10. [Manifest settings] (マニフェスト設定) で、マニフェスト設定に関する追加情報を入力します。
- マニフェストウィンドウ (秒): 各マニフェストに含まれる時間ウィンドウの秒単位。最小値は 30 秒で、最大値は 3,600 秒です。
 - 広告マークアップタイプ (HLS 出力のみ): VOD プログラム広告ブレイクに表示される広告タグのタイプ。選択 Daterange すると、MediaTailor が EXT-X-DATERANGE タグ付きの VOD プログラムに広告ブレイクを挿入します。を選択すると Scte35 Enhanced、MediaTailor は EXT-X-CUE-OUT と EXT-X-CUE-IN タグを使用して VOD プログラムに広告ブレイクを挿入します。これらのタグタイプの詳細については、「」を参照してください [広告時間枠の SCTE-35 メッセージ](#)。ライブワークフローの場合、MediaTailor は常に DATERANGE タグを通過し、選択した広告マークアップタイプに関係なく、拡張 Scte35 タグを通過しません。
11. 複数のチャンネル出力を設定する場合は、[Outputs] (出力) で [Add] (追加) をクリックします。その後、この手順のステップ 6 と 7 を実行して、出力の詳細を設定します。
12. [次へ] を選択します。
13. [Channel policy] (チャンネルポリシー) で、チャンネルの IAM ポリシー設定を選択します。
- [Do not attach channel policy] (チャンネルポリシーをアタッチしない): 再生をこのアカウントの認証情報にアクセスできるユーザーのみに制限します。
 - [Attach custom policy] (カスタムポリシーをアタッチする): 独自のポリシーを定義して、アクセスを必要に応じて制限します。
 - [Attach public policy] (パブリックポリシーをアタッチする): チャンネルの出力に対するすべての着信クライアントリクエストを受け入れます。MediaTailor 広告挿入を使用する場合は、このオプションを使用する必要があります。
14. [次へ] を選択します。

15. [Review and create] (確認と作成) ペインで設定を確認します。
16. [Create channel] (チャンネルを作成) をクリックします。

 Note

チャンネルは停止状態で作成されます。MediaTailor コンソールまたは MediaTailor StartChannel API で開始するまで、チャンネルはアクティブになりません。

チャンネルの出力でのソースグループの使用

ソースグループは、パッケージ設定とチャンネル上の出力を関連付けます。ソースグループの名前は、ソースでパッケージ設定を作成するときに特定します。その後、チャンネルで出力を作成するときにそれと同じ名前を入力して、出力とパッケージ設定を関連付けます。チャンネルのプログラムに追加される VOD ソースとライブソースは、出力で識別されるソースグループに属している必要があります。

例えば、次のようになります。

- VOD ソース 1 と 2 の両方に、[HLS]、[DASH]、および [HLS-4K] のソースグループを持つ 3 つのパッケージ設定がある。
- VOD ソース 3 に、[HLS] および [DASH] のソースグループを持つ 2 つのパッケージ設定がある。

チャンネル A に [HLS] と [DASH] のソースグループを持つ 2 つの出力がある場合、チャンネル出力は 3 つのすべての VOD ソースを使用できます。これは、VOD ソース 1、2、および 3 のすべてに [HLS] および [DASH] のソースグループラベルを持つパッケージ設定があるからです。

チャンネル B に [HLS] および [HLS-4K] のソースグループを持つ 2 つの出力がある場合、このチャンネルは VOD ソース 1 と 2 を使用できますが、3 は使用できません。これは、VOD ソース 1 と 2 の両方に [HLS] および [HLS-4K] のソースグループラベルを持つパッケージ設定があるためです。

チャンネル C に [DASH] ソースグループを持つ単一の出力がある場合、このチャンネルは 3 つのすべての VOD ソースを使用できます。3 つのすべての VOD ソースには、[DASH] ソースグループを持つパッケージ設定があります。

MediaTailor コンソールを使用してチャンネルを削除する

チャンネルを削除するには、以下の手順を実行します。

チャンネルを削除する

1. MediaTailor コンソール (<https://console.aws.amazon.com/mediatailor/>) を開きます。
2. ナビゲーションペインで、[Channel assembly] (チャンネルアセンブリ) > [Channels] (チャンネル) とクリックします。
3. 削除するチャンネルを選択します。
4. チャンネルが実行中の場合は、[Actions] (アクション) ドロップダウンメニューから [Stop] (停止) を選択します。チャンネルは、停止してから削除する必要があります。
5. チャンネルが停止したら、[Actions] (アクション) ドロップダウンメニューから [Delete] (削除) を選択します。

チャンネルのスケジュールにプログラムを追加する

各プログラムには、VOD ソースまたはアカウント内のソースロケーションの一部であるライブソースが含まれています。プログラムをチャンネルのスケジュールに追加して、それらがチャンネルのストリームで再生される順序を制御します。

VOD ソースを含むプログラムは、1 つ以上の広告時間枠で設定できます。各広告ブレイクには、ソースロケーションからの VOD ソースであるスレートが含まれています。広告ブレイクを作成するには、プログラムにスレートをミリ秒単位のオフセットで追加します。

トピック

- [MediaTailor コンソールを使用したチャンネルスケジュール内でのプログラムの作成](#)
- [プログラムルールを使用して対象者のコホートと代替コンテンツを定義する](#)
- [対象者固有のマニフェストの生成](#)

MediaTailor コンソールを使用したチャンネルスケジュール内でのプログラムの作成

次の手順では、MediaTailor コンソールを使用してチャンネルのスケジュール内にプログラムを作成する方法について説明します。オプションで広告ブレイクを設定する方法も説明されています。MediaTailor API を使用してプログラムを作成する方法については、API AWS Elemental MediaTailor リファレンスの [CreateProgram](#) を参照してください。

プログラムを追加するには

1. MediaTailor コンソール (<https://console.aws.amazon.com/mediatailor/>) を開きます。
2. ナビゲーションペインで、[Channel assembly] (チャンネルアセンブリ) > [Channels] (チャンネル) とクリックします。
3. [Channels] (チャンネル) ペインで、[チャンネルを作成する](#) 手順で作成したチャンネルを選択します。
4. プログラムの詳細に、プログラムの詳細を入力します。
 - [Name] (名前): チャンネルに追加するプログラムの名前です。
 - ソースタイプ: プログラムが再生するソースビデオのタイプを決定します。このオプションは、標準チャンネルでのみ使用できます。
 - VOD - プログラムは、録画済み TV エピソードなどのvideo-on-demandソースを再生します。
 - Live - プログラムは、ライブニュースブロードキャストなどのライブソースを再生します。
 - ソースロケーション名: MediaTailor がプログラムに関連付けるソースロケーション。
 - 既存のソースロケーションを選択する を選択した場合は、ソースロケーションの選択メニューからソースロケーション名を選択します。または、ソースの場所を名前で検索します。これは、ソースロケーションの数が多い場合に便利です。
 - [Enter the source location name] (ソースロケーション名を入力する) を選択する場合は、名前でソースロケーションを検索します。
 - VOD ソース名: MediaTailor がプログラムに関連付ける VOD ソースの名前。
 - [Select an existing VOD source] (既存の VOD ソースを選択する) を選択する場合は、アカウントに関連付けられている VOD ソースのリストから VOD ソース名を選択します。または、VOD ソースを名前で検索します。この方法は、VOD ソースの数が多い場合に便利です。
 - 名前で検索 を選択した場合は、ライブソースを名前で検索します。
 - ライブソース名: プログラムに関連付けるライブソースの名前。このオプションは、Live をソースタイプとして選択した場合にのみ使用できます。
 - 既存のソースロケーションを選択する を選択した場合は、ソースロケーションの選択メニューからソースロケーション名を選択します。または、ソースの場所を名前で検索します。これは、ソースロケーションの数が多い場合に便利です。
 - [Enter the source location name] (ソースロケーション名を入力する) を選択する場合は、名前でソースロケーションを検索します。

- 既存のライブソースを選択する を選択した場合は、アカウントに関連付けられているライブソースのリストからライブソース名を選択します。または、ライブソースを名前で検索することもできます。これは、多数のライブソースがある場合に役立ちます。
 - 名前で検索 を選択した場合は、ライブソースを名前で検索します。
5. 次へ を選択して、スケジュール設定タブに移動します。
6. [Playback configuration] (再生設定) で、チャンネルのスケジュールでプログラムが再生されるタイミングを定義します。
- ミリ秒単位の期間: プログラムの期間をミリ秒単位で定義します。このオプションは、ライブソースを使用するプログラムでのみ使用できます。
 - 移行タイプ: スケジュールでプログラムからプログラムへの移行を定義します。
 - 相対: プログラムは、スケジュール内の別のプログラムの前後に再生されます。このオプションは、VOD ソースを使用するプログラムでのみ使用できます。
 - 絶対: プログラムは特定のウォールクロック時間に再生されます。MediaTailor は、ユーザー指定のクロック時間でのプログラムの再生をベストエフォートで実行します。MediaTailor は、前述のプログラムまたはスレート間の共通のセグメント境界でプログラムの再生を開始します。このオプションは、線形 を使用するように設定されたチャンネルでのみ使用できます。[\[Playback mode\] \(再生モード\): 再生モードは、チャンネルの再生動作を設定します。MediaTailor は、以下の再生モードをサポートします。](#)
 - プログラム開始時刻: 絶対遷移タイプの場合、プログラムの再生がスケジュールされているウォールクロック時間。このプログラムを実行中のリニアチャンネルに追加している場合は、現在の時刻から 15 分以降の開始時間を入力する必要があります。
 - 相対位置: 別のプログラムを基準にして、プログラムをスケジュールに挿入する場所を選択します。これには、[Before program] (プログラムの前) または [After program] (プログラムの後) を選択できます。このプログラムがチャンネルのスケジュールの最初のプログラムである場合、この設定は適用されません。
 - 既存のプログラムを選択する を選択した場合は、既存のプログラムを使用するメニューから、チャンネルによって再生される次の 100 のプログラムの事前定義されたリストからプログラム名を選択します。
 - プログラムを名前で検索 を選択した場合は、チャンネル内の既存のプログラムの名前を入力します。

プログラムに広告ブレイクを追加する場合は、次のステップに進みます。広告ブレイクは、VOD ソースを使用するプログラムでのみ設定できます。ライブソースの場合、DASH マニ

フェストの広告ブレイクと、EXT-X-DATERANGE タグを使用する HLS マニフェストの広告ブレイクは自動的に渡されます。

7. Next を選択して、Add ad breaks に移動します。
8. [Add ad break] (広告ブレイクを追加) をクリックします。[Ad breaks] (広告ブレイク) で、広告ブレイクを設定します。
 - スレートソースの場所名: 既存のソースの場所を選択し、このタスクの前半で作成したスレートが保存されているソースの場所を選択します。
 - VOD ソース名: 既存の VOD ソースを選択し、このタスクの前半で追加したスレートに使用する VOD ソースを選択します。広告ブレイクの継続時間は、スレートの継続時間によって決まります。
 - ミリ秒単位のオフセット: この値は、プログラムの開始からのオフセットとして、広告時間枠の開始時間をミリ秒単位で決定します。VOD ソースの期間より短く、プログラムの VOD ソース内のすべてのトラック (すべてのオーディオ、ビデオ、クローズドキャプショントラック) のセグメント境界と一致する値を入力します。そうしないと、広告ブレイクはスキップされます。例えば、0 を入力すると、プログラムの開始前に再生されるプレロール広告ブレイクが作成されます。
 - 利用可能な番号: MediaTailor は `splice_insert.avail_num`、SCTE-35 仕様の「[デジタルプログラム挿入キューメッセージ](#)」のセクション 9.7.3.1. で定義されているように、この値を に書き込みます。デフォルト値は 0 です。値は 0~256 の範囲である必要があります。
 - 想定どおり: MediaTailor は `splice_insert.avails_expected`、SCTE-35 仕様のセクション 9.7.3.1. で定義されているように、この値を に書き込みます。デフォルト値は 0 です。値は 0~256 の範囲である必要があります。
 - スプライスイベント ID: MediaTailor は `splice_insert.splice_event_id`、SCTE-35 仕様のセクション 9.7.3.1. で定義されているように、この値を に書き込みます。デフォルト値は 1 です。
 - 一意のプログラム ID: MediaTailor は `splice_insert.unique_program_id`、SCTE-35 仕様のセクション 9.7.3.1. で定義されているように、この値を に書き込みます。デフォルト値は 0 です。値は 0~256 の範囲である必要があります。
9. 標準リニアチャンネルの場合は、次へ を選択して代替メディアの設定に移動します。

MediaTailor を使用して代替メディアを作成する方法の詳細については、「」を参照してください [代替メディアの作成](#)。

MediaTailor を使用して広告時間枠をパーソナライズする方法の詳細については、「」を参照してください [チャンネルストリームにパーソナライズされた広告と広告ブレイクを挿入する](#)。

10. 次へ を選択して、レビューと作成に移動します。
11. プログラムの追加を選択します。

MediaTailor を使用して広告時間枠をパーソナライズする方法の詳細については、「」を参照してください [チャンネルストリームにパーソナライズされた広告と広告ブレイクを挿入する](#)。

12.

 Important

ループするチャンネルの場合、変更時点から 10 分以内にスケジュールされているプログラムのプログラムリストを変更しても、編集内容は次のループまで表示されません。

[Program details] (プログラムの詳細) で、プログラムに関する詳細情報を入力します。

- [Name] (名前): チャンネルに追加するプログラムの名前です。
- ソースタイプ: プログラムが再生するソースのタイプを決定します。このオプションは、標準チャンネルでのみ使用できます。
 - VOD - プログラムは、録画済み TV エピソードなどの VOD ソースを再生します。
 - Live - プログラムは、ライブニュースブロードキャストなどのライブソースを再生します。
- [Source location name] (ソースロケーション名): プログラムに関連付けられるソースロケーションです。

[Select an existing source location] (既存のソースロケーションを選択する) を選択する場合は、[Select an existing source location] (既存のソースロケーションを選択する) のドロップダウンメニューからソースロケーション名を選択します。別の方法として、名前でソースロケーションを検索することもできます。これは、ソースロケーションの数が多い場合に便利です。

[Enter the source location name] (ソースロケーション名を入力する) を選択する場合は、名前でソースロケーションを検索します。

- [VOD source name] (VOD ソース名): プログラムに関連付けられる VOD ソースの名前です。

[Select an existing VOD source] (既存の VOD ソースを選択する) を選択する場合は、アカウントに関連付けられている VOD ソースのリストから VOD ソース名を選択します。別の方法として、名前で VOD ソースを検索することもできます。この方法は、VOD ソースの数が多い場合に便利です。

[Search by name] (名前で検索する) を選択する場合は、名前で VOD ソースを検索します。

- **ライブソース名:** プログラムに関連付けるライブソースの名前。このオプションは、ソースタイプとして Live を選択した場合にのみ使用できます。

既存のライブソースを選択する を選択した場合は、アカウントに関連付けられているライブソースのリストからライブソース名を選択します。または、ライブソースを名前で検索することもできます。これは、多数のライブソースがある場合に役立ちます。

名前で検索 を選択した場合は、ライブソースを名前で検索します。

13. [Playback configuration] (再生設定) で、チャンネルのスケジュールでプログラムが再生されるタイミングを定義します。

- **ミリ秒単位の期間:** プログラムの期間をミリ秒単位で定義します。このオプションは、ライブソースを使用するプログラムでのみ使用できます。
- **[Transition type] (推移タイプ):** スケジュール内でのプログラムからプログラムへの遷移を定義します。
 - **[Relative] (相対的)** - プログラムは、スケジュール内の別のプログラムの前または後で再生されます。このオプションは、VOD ソースを使用するプログラムでのみ使用できます。
 - **[Absolute] (絶対的)** - プログラムは、特定のウォールクロック時間に再生されます。MediaTailor は、ユーザー指定のクロック時間でのプログラムの再生をベストエフォートで実行します。プログラムの再生は、先行するプログラムまたはスレートとの間の共通のセグメント境界で開始されます。このオプションを利用できるのは、[linear playback mode](#) を使用するように設定されたチャンネルのみです。

Note

絶対的な遷移タイプでは、以下の動作に注意してください。

- スケジュール内の先行プログラムの継続時間がウォールクロック時間を超える場合、MediaTailor は、ウォールクロック時間に最も近い共通のセグメント境界で先行プログラムをカットします。
 - スケジュール内のプログラム間にギャップがある場合、MediaTailor は [filler slate](#) を再生します。スレートの継続時間がギャップの継続時間よりも短い場合、MediaTailor はスレートをループします。
- **[Program start time] (プログラム開始時間)** - 絶対的な推移タイプの場合は、プログラムの再生がスケジュールされているウォールクロック時間です。このプログラムを実行中のリニアチャンネルに追加している場合は、現在の時刻から 15 分以降の開始時間を入力する必要があります。

- [Relative position] (相対位置): プログラムをスケジュールに挿入する、別のプログラムに相対的な位置を選択します。これには、[Before program] (プログラムの前) または [After program] (プログラムの後) を選択できます。このプログラムがチャンネルのスケジュールの最初のプログラムである場合、この設定は適用されません。
- [Relative program] (相対プログラム): 新しいプログラムをその前または後に挿入するために使用されるプログラムの名前です。このプログラムがチャンネルのスケジュールの最初のプログラムである場合、この設定は適用されません。

[Select an existing program] (既存のプログラムを選択する) を選択する場合は、[Select an existing program] (既存のプログラムを使用する) のドロップダウンメニューから、チャンネルで再生される次の 100 個のプログラムの事前定義されたリストからプログラム名を選択します。

[Search for a program by name] (名前でプログラムを検索する) を選択する場合は、チャンネル内の既存のプログラムの名前を入力します。

プログラムに広告ブレイクを追加する場合は、次のステップに進みます。広告ブレイクは、VOD ソースを使用するプログラムでのみ設定できます。ライブソースの場合、DASH マニフェストの広告ブレイクと、EXT-X-DATERANGE タグを使用する HLS マニフェストの広告ブレイクは自動的に渡されます。

14. [Add ad break] (広告ブレイクを追加) をクリックします。[Ad breaks] (広告ブレイク) で、広告ブレイクを設定します。

- [Slate source location name] (スレートのソースロケーション名): [Select an existing source location] (既存のソースロケーションを選択する) を選択して、このチュートリアルで先ほど作成したスレートが保存されているソースロケーションを選択します。
- [VOD source name] (VOD ソース名): [Select an existing VOD source] (既存の VOD ソースを選択する) を選択して、このチュートリアルで先ほど追加したスレート用に使用している VOD ソースを選択します。広告ブレイクの継続時間は、スレートの継続時間によって決まります。
- [Offset in milliseconds] (ミリ秒単位のオフセット): この値は、ミリ秒単位での広告ブレイクの開始時間を、プログラムの開始時点に相対するオフセットとして指定します。VOD ソースの期間より短く、プログラムの VOD ソース内のすべてのトラック (すべてのオーディオ、ビデオ、クローズドキャプショントラック) のセグメント境界と一致する値を入力します。そうしないと、広告ブレイクはスキップされます。例えば、0 を入力すると、プログラムの開始前に再生されるプレロール広告ブレイクが作成されます。

 Note

MediaTailor が HLS の場合は、EventStreamDASH EXT-X-CUE-OUTの場合はなどのDATERANGE広告マーカを検出し、VOD ソース内で期間は 0 の場合、ドロップダウンメニューからそれらの広告マーカのオフセットを選択して、広告ブレイクのオフセットとして使用できます。広告オポチュニティを検出するには、VOD ソース内のすべてのパッケージ設定で同じオフセットに存在し、その期間はゼロである必要があります。

- メッセージタイプ: SCTE-35 広告挿入タイプ。SPICE_INSERT または TIME_SIGNAL を選択します。
- SPICE_INSERT: スプライス挿入パラメータを使用して、広告ブレイクに関する基本的なメタデータを提供します。
- TIME_SIGNAL: セグメンテーション記述子を使用して、より高度なメタデータを提供します。メッセージタイプの違いの詳細については、「」を参照してください[the section called “広告時間枠の SCTE-35 メッセージ”](#)。

SPICE_INSERT メッセージタイプの場合:

- [Avail number] (Avail 番号) は、SCTE-35 仕様のセクション 9.7.3.1 で定義されているとおり、splice_insert.avail_num に書き込まれます。デフォルト値は 0 です。値は、0 以上 256 以下にする必要があります。
- [Avail expected] (期待される avail 数) は、SCTE-35 仕様のセクション 9.7.3.1 で定義されているとおり、splice_insert.avails_expected に書き込まれます。デフォルト値は 0 です。値は、0 以上 256 以下にする必要があります。
- [Splice event ID] (スプライスイベント ID) は、SCTE-35 仕様のセクション 9.7.3.1 で定義されているとおり、splice_insert.splice_event_id に書き込まれます。デフォルト値は 1 です。
- [Unique program ID] (一意のプログラム ID) は、SCTE-35 仕様のセクション 9.7.3.1 で定義されているとおり、splice_insert.unique_program_id に書き込まれます。デフォルト値は 0 です。値は、0 以上 256 以下にする必要があります。

TIME_SIGNAL メッセージタイプの場合:

- セグメンテーションイベント ID の場合、これは SCTE-35 仕様のセクション 10.3.3.1 で定義されているsegmentation_descriptor.segmentation_event_idように、に書き込まれます。デフォルト値は 1 です。

- セグメンテーションタイプ ID の場合、これは SCTE-35 仕様のセクション 10.3.3.1 で定義されているように `segmentation_descriptor.segmentation_type_id`、に書き込まれます。デフォルト値は 48 (0x30) です。値は、0 以上 256 以下にする必要があります。
- Segmentation UPID の場合、これは SCTE-35 仕様のセクション 10.3.3.1 で定義されている `segmentation_descriptor.segmentation_upid` ように、に書き込まれます。値は、文字 0-9 とを含む 16 進数の文字列である必要があります A-F。デフォルト値は空の文字列です。
- Segmentation UPID タイプの場合、これは SCTE-35 仕様のセクション 10.3.3.1 で定義されている `segmentation_descriptor.segmentation_upid_type` ように、に書き込まれます。デフォルト値は 14 (0x0E) です。値は、0 以上 256 以下にする必要があります。
- セグメント番号の場合、これは SCTE-35 仕様のセクション 10.3.3.1 で定義されている `segmentation_descriptor.segment_num` ように、に書き込まれます。デフォルト値は 0 です。値は、0 以上 256 以下にする必要があります。
- 予想されるセグメントについては、SCTE-35 仕様のセクション 10.3.3.1 で定義されているように `segmentation_descriptor.segments_expected`、これはに書き込まれます。デフォルト値は 0 です。値は、0 以上 256 以下にする必要があります。

15. [Add program] (プログラムを追加) をクリックします。

広告ブレイクをパーソナライズするための MediaTailor の使用に関する詳細については、「[チャンネルストリームにパーソナライズされた広告と広告ブレイクを挿入する](#)」を参照してください。

Note

チャンネルに Enhanced SCTE-35 広告マークアップタイプの出力が少なくとも 1 つある場合は、広告ブレイクメタデータを送信できます。MediaTailor は、送信されたキーと値のペアを広告時間枠の EXT-X-ASSET タグに書き込みます。

プログラムルールを使用して対象者のコホートと代替コンテンツを定義する

プログラムルールを使用すると、チャンネルのオーディエンスコホートを定義し、それらのオーディエンスに対して再生する代替メディアを指定できます。1 つ以上の代替コンテンツソースをプログラムの対象者に関連付けることができます。プログラムが終了すると、追加の代替メディアを指定しない限り、デフォルトのオーディエンスコンテンツが再生されます。

プログラムルールは、LINEAR 再生モードの STANDARD 階層チャンネルで使用できます。MediaTailor チャンネルは、すべての VOD ソースとライブソースの代替メディアをサポートしています。

使用例については、[AWS MediaTailor でのプログラムルールの使用](#)を参照してください。

対象者の定義

MediaTailor チャンネルを設定するときにオーディエンスを 1 つずつ入力して、チャンネルのオーディエンスを定義します。これを行うには、MediaTailor コンソールまたは MediaTailor CreateChannel API を使用します。各対象者は 1~32 文字の英数字である必要があります。対象者に指定された値が無効である場合、リクエストは失敗します。

LINEAR 再生モードの STANDARD 階層チャンネルでのみオーディエンスを定義できます。

対象者を更新する必要がある場合は、MediaTailor コンソールまたは MediaTailor API を使用して更新できます。UpdateChannel

ProgramRules 機能を使用している場合は、CreateProgramまたは UpdateProgramリクエストでAudienceMedia定義された に、チャンネルで定義された既存のオーディエンスが含まれていることを確認してください。

代替メディアの作成

次のタスクでは、MediaTailor コンソールを使用して代替メディアを定義する方法について説明します。MediaTailor API を使用して代替メディアを定義する方法については、API AWS Elemental MediaTailor リファレンスの[CreateProgram](#)「」を参照してください。

新しいプログラムで代替メディアを定義するには:

1. MediaTailor コンソール (<https://console.aws.amazon.com/mediatailor/>) を開きます。
2. ナビゲーションペインで、チャンネルアセンブリ > チャンネルを選択します。
3. 代替メディアを追加するチャンネル名を選択します。
4. プログラムを作成します。詳細については、「[MediaTailor コンソールを使用したチャンネルスケジュール内でのプログラムの作成](#)」を参照してください。
5. 代替メディアを設定します。
 - オーディエンスボックスに「追加」を選択して、代替メディアを定義するオーディエンスを選択します。
 - オーディエンスメニューからチャンネルで定義されたオーディエンスを選択します。

- 代替メディアの追加を選択して、プログラムの代替メディアの定義を開始します。
- MediaTailor は代替メディア 1 ボックスを作成します。これは、MediaTailor がプログラムで代替メディアとして再生する最初のコンテンツです。
- 代替メディア 1 ボックス内:
 - ソースの場所を選択します。
 - VOD またはライブソースタイプを選択します。

VOD の場合

- ソースタイプの VOD を選択します。
- (オプション) クリップ範囲を指定します。代替メディア VOD ソースを含む VOD ソースでは、再生する VOD ソースの一部を指定し、ソースの開始または終了からクリッピングできます。指定 開始オフセットと終了オフセットはミリ秒単位です。
- (オプション) a\広告時間枠の追加。これは、プログラムを作成するときと同じ方法で行われます。詳細については、「[MediaTailor コンソールを使用したチャネルスケジュール内のプログラムの作成](#)」を参照してください。

ライブの場合

- ソースタイプのライブを選択します。
- ライブソースを選択します。
- このライブソースが開始する wall-clock 時刻の開始時間をミリ秒単位で入力します。ライブソースは、定義されているデフォルトのプログラムの時間枠内にのみ再生されます。開始時刻がデフォルトプログラムの開始時刻より前である場合、デフォルトプログラムが開始されるまで開始されません。開始時刻がデフォルトのプログラム終了後である場合、MediaTailor はライブソースを再生しません。
- 期間をミリ秒単位で入力します。期間は 10 分以上の長さである必要があります。
- 代替メディアを再度追加を選択すると、対象者のためにこのプログラムに追加の代替メディアを追加できます。これにより、代替メディア 2 というラベルが付いた別のボックスが作成されます。プログラムごとに、対象者ごとに最大 5 つの代替メディアソースを指定できます。
- 必要なすべての対象者の代替メディアの定義が完了したら、次へ を選択し、プログラムの作成を続行します。

詳細については、「[MediaTailor コンソールを使用したチャネルスケジュール内のプログラムの作成](#)」を参照してください。

Note

代替メディアは、それが定義されているプログラムの時間枠でのみ再生されます。すべての代替コンテンツがデフォルトコンテンツをオーバーランした場合、MediaTailor はそれを切り捨てます。MediaTailor は、定義された順序で代替メディアを再生します。ライブ代替メディアの開始時間は常に優先され、以前にスケジュールされた VOD ソースまたはライブソースは切り捨てられます。対象者の代替メディアで満たされていない場合は、チャンネル定義のフィラースレートで満たされます。

- 他のオーディエンスのオーディエンスメディアを定義するには、オーディエンスの横にもう一度追加を選択します。上記の説明に従って、新しく作成した対象者を選択し、対象者 ID を設定し、代替メディアを追加します。最大 5 人のオーディエンスが、1 つのプログラムで代替メディアを使用できます。

対象者固有のマニフェストの生成

特定の対象者のマニフェストを取得するには、`aws.mediatailor.channel.audienceId` クエリパラメータを使用します。このクエリパラメータは、CDN によって動的に追加すること、コンテンツまたはカスタマー管理システムへの呼び出しを通じて追加することもできます。特定の再生セッションと MediaTailor の `audienceId` 外部との関連付けを維持する必要があります。これにより、デフォルトのコンテンツの代わりに、その対象者に定義された代替メディアを含む対象者固有のマニフェストが取得されます。特定の対象者に対してマニフェストがリクエストされると、プレイヤーは常に同じ対象者 ID でマニフェストをリクエストするか、再生エラーが発生する可能性があります。

チャンネルに存在しないオーディエンスに対してリクエストが行われた場合、MediaTailor は 404 エラーを返します。

Example 対象者のマニフェストの取得

```
https:// prefix>.channel-assembly.mediatailor.us-west-2.amazonaws.com/v1/channel/ExampleChannel/index_dash.mpd?aws.mediatailor.channel.audienceId=Seattle
```

チャンネルストリームにパーソナライズされた広告と広告ブレイクを挿入する

MediaTailor を使用すると、SCTE-35 マーカーでコンテンツを条件付けすることなく、プログラムに広告ブレイクを挿入することで、チャンネルアセンブリのリニアストリームを収益化できます。MediaTailor 広告挿入サービス、または任意のサーバー側の広告挿入 (SSAI) でチャンネルアセンブリを使用できます。

以下のトピックでは、チャンネルのリニアストリームにパーソナライズされた広告と広告ブレイクを挿入する方法を示します。

トピック

- [MediaTailor での広告挿入の設定](#)
- [広告時間枠の SCTE-35 メッセージ](#)

MediaTailor での広告挿入の設定

パーソナライズされた広告をチャンネルのストリームに挿入するには、チャンネルのエンドポイント URL がコンテンツソースになります AWS Elemental MediaTailor。このガイドでは、広告挿入用に MediaTailor を設定する方法について説明します。

前提条件

開始する前に、以下の前提条件を満たしていることを確認してください。

- MediaTailor 広告挿入用の HLS ストリームと DASH ストリームを準備します。
 - コンテンツストリームをまだ準備していない場合は、「MediaTailor [ステップ 2: ストリームを準備する](#) 広告挿入の開始方法」トピックの「」を参照してください。 MediaTailor
- 広告決定サーバー (ADS) がある。
- プログラムで広告時間枠の設定を行います。詳細については、[Configuring ad breaks for your program](#) 「」の手順を参照してください。

ベストプラクティスとして、チャンネルアセンブリと MediaTailor 広告挿入の間にコンテンツ配信ネットワーク (CDN) を使用することを検討してください。MediaTailor 広告挿入サービスは、追加のオリジンリクエストを生成できます。したがって、チャンネルアセンブリからマニフェストをプロキシする

ように CDN を設定し、コンテンツソース URLs で CDN プレフィックス付き URL を使用するのがベストプラクティスです。

広告挿入用に MediaTailor を設定する

以下は、チャンネルのストリームにパーソナライズされた広告を挿入できるように MediaTailor コンソール設定を構成する方法を示しています。

広告挿入用に MediaTailor を設定するには

1. MediaTailor コンソール (<https://console.aws.amazon.com/mediatailor/>) を開きます。
2. ナビゲーションペインで [Configurations] (設定) をクリックします。
3. [Required settings] (必須の設定) で、設定に関する基本的な必須情報を入力します。
 - [Name] (名前): 設定の名前です。
 - [Content source] (コンテンツソース): チャンネルの出力からの再生 URL (ファイル名と拡張子を除く) を入力します。MediaTailor の設定に関する詳細については、「[必須の設定](#)」を参照してください。
 - 広告決定サーバー: ADS の URL を入力します。
4. オプションで、[Configuration aliases] (設定エイリアス)、[Personalization details] (パーソナライゼーションの詳細)、および [Advanced settings] (アドバンスド設定) を設定できます。これらの設定については、「[オプションの設定](#)」を参照してください。
5. ナビゲーションバーで [Create configuration] (設定を作成) をクリックします。

広告挿入用に MediaTailor をセットアップしたので、広告時間枠を設定することもできます。詳細な手順については、「[MediaTailor 広告挿入の使用開始](#)」を参照してください。

広告時間枠の SCTE-35 メッセージ

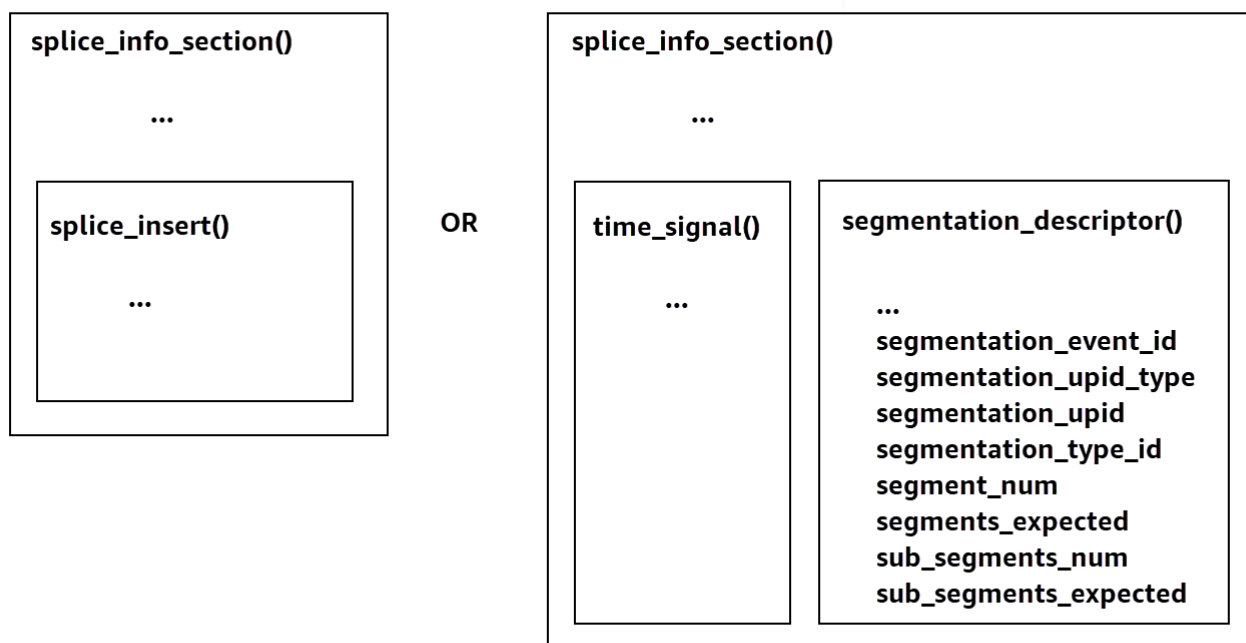
MediaTailor を使用すると、ソースの場所と VOD ソースリソースに基づいてコンテンツチャンネルを作成できます。その後、チャンネルのスケジュールに従って、プログラムごとに 1 つ以上の広告時間枠を設定できます。SCTE-35 仕様に基づくメッセージを使用して、コンテンツに広告時間枠の条件を設定します。たとえば、SCTE-35 メッセージを使用して、広告時間枠に関するメタデータを提供できます。SCTE-35 仕様の詳細については、「[デジタルプログラム挿入キューメッセージ](#)」を参照してください。

広告時間枠は、次の 2 つの方法のいずれかでセットアップします。

- `time_signal` SCTE-35 メッセージを `segmentation_descriptor` メッセージでアタッチします。 `segmentation_descriptor` このメッセージには、広告時間枠に関する詳細情報を伝えるコンテンツ識別子など、より高度なメタデータフィールドが含まれています。MediaTailor は、(HLS) または EXT-X-DATERANGE (EventStreamDASH) 広告マーカの SCTE-35 データの一部として、広告メタデータを出カマニフェストに書き込みます。
- 広告時間枠に関する基本的なメタデータを提供する `splice_insert` SCTE-35 メッセージを添付します。
- HLS:
 - 広告マークアップタイプが の場合 `Daterange`、MediaTailor はマニフェストで広告ブレイクを `EXT-X-DATERANGE` タグとして指定します。
 - 広告マークアップタイプが の場合 `Scte35 Enhanced`、MediaTailor は次のタグを使用して広告ブレイクを指定します。
 - MediaTailor は、広告スレートの最初のセグメント `EXT-X-CUE-OUT` に を配置し、コンテンツから広告時間枠へのカットを示します。これには、 など、広告時間枠の予想期間が含まれます `EXT-X-CUE-OUT:Duration=30`。
 - `>EXT-X-ASSET`: このタグは 同じセグメントに表示され `EXT-X-CUE-OUT`、プログラムの作成時または更新時に `AdBreak` で提供される広告ブレイクメタデータが含まれます。常に が含まれます `CAID`。
 - `EXT-0ATCLS-SCTE35`: このタグは 同じセグメントに表示され `EXT-X-CUE-OUT`、SCTE-35 メッセージの base64 エンコードされたバイトが含まれます。
 - `EXT-X-CUE-OUT-CONT`: このタグは、広告スレート内の後続の各セグメントに表示され、期間と経過時間情報が含まれます。また、base64 でエンコードされた SCTE-35 メッセージとも含まれています `CAID`。
 - `EXT-X-CUE-IN`: このタグは、広告時間枠が終了した後、コンテンツの最初のセグメントに表示され、広告時間枠からコンテンツへのカットを示します。

次の図は、SCTE-35 メッセージを使用してチャンネルで広告ブレイクを設定する 2 つの方法を示しています。

- `splice_insert()` メッセージを使用して、基本的なメタデータで広告時間枠を設定します。
- `time_signal()` メッセージと `segmentation_descriptor()` メッセージを使用して、より詳細なメタデータを含む広告時間枠を設定します。



の使用の詳細についてはtime_signal、2022 SCTE-35 仕様の「[デジタルプログラム挿入キューメッセージ](#)」のセクション 9.7.4 を参照してください。

広告時間枠情報は、出力 splice_info_section SCTE-35 データに表示されます。MediaTailor では、1つのsegmentation_descriptorメッセージを1つのtime_signalメッセージとペアリングできます。

Note

segmentation_descriptor メッセージを送信する場合は、time_signalメッセージタイプの一部としてメッセージを送信する必要があります。time_signal メッセージには、MediaTailor が特定のタイムスタンプを使用して構築するsplice_timeフィールドのみが含まれます。

次の表は、MediaTailor が各segmentation_descriptorメッセージに必要とするフィールドを示しています。詳細については、[ANSI Webstore ウェブサイト](#)で購入できる 2022 SCTE-35 仕様のセクション 10.3.3.1 を参照してください。

segmentation_descriptor メッセージの必須フィールド

フィールド	タイプ	デフォルトの値	説明
segmentation_event_id	integer	1	これは に書き込まれますsegmentation_descriptor.segmentation_event_id 。
segmentation_upid_type	整数	14 (0x0E)	これは に書き込まれますsegmentation_descriptor.segmentation_upid_type 。値は 0～256 の間でなければなりません。
segmentation_upid	string	"" (空の文字列)	これは に書き込まれますsegmentation_descriptor.segmentation_upid 。値は、文字 0-9と を含む 16 進数の文字列である必要がありますA-F。
segmentation_type_id	整数	48 (0x30)	これは に書き込まれますsegmentation_descriptor.segmentation_type_id 。値は 0～256 の間でなければなりません。

フィールド	タイプ	デフォルトの値	説明
segment_num	整数	0	これは に書き込まれますsegmentation_descriptor.segment_num 。値は 0 ~ 256 の間でなければなりません。
segments_expected	整数	0	これは に書き込まれますsegmentation_descriptor.segments_expected 。値は 0 ~ 256 の間でなければなりません。
sub_segment_num	整数	null	これは に書き込まれますsegmentation_descriptor.sub_segment_num 。値は 0 ~ 256 の間でなければなりません。
sub_segments_expected	整数	null	これは に書き込まれますsegmentation_descriptor.sub_segments_expected 。値は 0 ~ 256 の間でなければなりません。

次の表は、MediaTailor が segmentation_descriptor メッセージの一部のフィールドに自動的に設定する値を示しています。

segmentation_descriptor メッセージのフィールドに MediaTailor によって設定された値

フィールド	タイプ	値
segmentation_event_cancel_indicator	ブール値	正
program_segmentation_flag	ブール値	正
delivery_not_restricted_flag	ブール値	正

MediaTailor は常に `segmentation_duration_flag`を に設定しますTrue。MediaTailor は、状態の内容の期間をティックで`segmentation_duration`フィールドに入力します。

 Note

MediaTailor が `time_signal` メッセージを送信すると、`splice_info_section` メッセージの `splice_command_type` フィールドが 6 (0x06) に設定されます。

HLS 出力では、time_signalメッセージAdBreakを含む の場合、出力EXT-X-DATERANGEタグにはsplice_info_sectionメッセージのシリアル化されたバージョンに設定された SCTE-35 フィールドが含まれます。たとえば、次のEXT-X-DATERANGEタグはsplice_info_sectionメッセージのシリアル化されたバージョンを示しています。

```
#EXT-X-DATERANGE:ID="\1",START-DATE="\2020-09-25T02:13:20Z",DURATION=300.0,SCTE35-OUT=0xFC002C0000000000000000FF05068000000000001602144355454900000000100E000019BFCC00E003000000000
```

DASH 出力では、time_signalメッセージAdBreakを含む の場合、出力EventStream要素には子として scte35:TimeSignal および scte35:SpliceInfoSection要素を持つ scte35:SegmentationDescriptor 要素が含まれます。scte35:TimeSignal 要素には子scte35:SpliceTime要素があり、 scte35:SegmentationDescriptor要素には

子scte35:SegmentationUpid要素があります。たとえば、次の DASH 出力は、EventStream要素の子要素構造を示しています。

```
<EventStream schemeIdUri="urn:scte:scte35:2013:xml" timescale="90000">
  <Event duration="27000000">
    <scte35:SpliceInfoSection protocolVersion="0" ptsAdjustment="0" tier="4095">
      <scte35:TimeSignal>
        <scte35:SpliceTime ptsTime="0" />
      </scte35:TimeSignal>
      <scte35:SegmentationDescriptor segmentNum="0"
segmentationDuration="27000000" segmentationEventCancelIndicator="false"
segmentationEventId="1" segmentationTypeId="48" segmentsExpected="0">
        <scte35:SegmentationUpid segmentationUpidFormat="hexBinary"
segmentationUpidType="14">012345</scte35:SegmentationUpid>
      </scte35:SegmentationDescriptor>
    </scte35:SpliceInfoSection>
  </Event>
</EventStream>
```

SCTE-35 メッセージを使用してチャンネルアセンブリの広告ブレイクを設定する方法、それらのメッセージの構造と必須フィールド、SCTE-35 メッセージを含むサンプル HLS および DASH 出力について学びました。

チャンネルの再生のタイムシフト

MediaTailor チャンネルアセンブリを使用すると、チャンネルの再生をタイムシフトできます。タイムシフトを使用すると、ビューワーは元のブロードキャスト時間以外の時間にチャンネルのコンテンツを視聴できます。たとえば、視聴者は、開始後にブロードキャストに参加した場合でも、最初からプログラムの視聴を開始できます。または、ビューワーはプログラムを一時停止し、後で視聴を再開できます。

チャンネルの再生をタイムシフトするには、チャンネルの再生 URL にstartパラメータを追加します。start パラメータは、再生を開始する時刻を、現在の時刻を基準にして指定します。たとえば、現在の時刻が午後 2 時で、午後 1 時から再生を開始する場合は、startパラメータを-3600秒単位で -1 時間である に設定します。

start パラメータは、正の値と負の値の両方を秒単位で受け入れます。

- 負の値は、現在の時刻を基準とした過去の時刻を示します。たとえば、 は「1 時間前から再生を開始するstart=-3600」を意味します。

- 正の値は、現在の時刻を基準にした将来の時刻を示します。例えば、 は「今後 1 時間から再生を開始するstart=3600」ことを意味します。

次の例は、チャンネルの再生 URL に startパラメータを追加する方法を示しています。

```
https://a1b2c3d4e5f6.mediapackage.us-west-2.amazonaws.com/out/v1/examplechannel/index.m3u8?start=-3600
```

この例では、再生は 1 時間前から開始されます。

start パラメータには以下の制限があります。

- 最大タイムシフトウィンドウは、過去または将来の 24 時間です。
- タイムシフトは、線形再生モードを使用するチャンネルでのみ使用できます。
- タイムシフトは、ループ再生モードを使用するチャンネルでは使用できません。

使用可能なタイムシフトウィンドウ外のstartパラメータを指定すると、MediaTailor はエラーを返します。

CDN を使用して MediaTailor の広告パーソナライゼーションとコンテンツ配信を最適化する

AWS Elemental MediaTailor はスタンドアロンサービスとして効果的に機能しますが、Amazon CloudFront やその他のサードパーティー CDN などのコンテンツ配信ネットワーク (CDNs) と統合することで、ストリーミングワークフローを大幅に強化できます。CDN 統合は、地理的に分散された大規模な対象者にコンテンツを提供する必要がある場合や、異なる AWS リージョン間で一貫した広告配信を確保したい場合に特に役立ちます。

CDN がない場合、ビューワーはパーソナライズされたマニフェストと広告セグメントのために MediaTailor に直接接続するため、特に MediaTailor 設定がデプロイされている AWS リージョンから遠くにあるビューワーの場合、レイテンシーが増加する可能性があります。さらに、トラフィックの多いイベント中に、MediaTailor への直接接続で負荷が増加し、パフォーマンスに影響する可能性があります。

MediaTailor の概念とワークフローの詳細については、「」を参照してください [とは AWS Elemental MediaTailor](#)。

CDN を MediaTailor と統合する場合、ウェブベースのプレイヤーで再生失敗を引き起こす可能性のある問題を防ぐために、適切な CORS (Cross-Origin Resource Sharing) 処理を設定することが重要です。広告セグメントとコンテンツセグメントの両方に適切な CORS 設定が不可欠です。広告セグメントは CORS の問題の影響を受けやすくなりますが、すべてのセグメントタイプに一貫した CORS 処理を適用すると、最も信頼性の高い再生エクスペリエンスが保証されます。適切な CORS 処理で CDN ルーティング動作を設定する詳細なガイダンスについては、「」を参照してください [MediaTailor の本番稼働対応 CloudFront 設定](#)。

CDN 統合により、高度なパラメータの受け渡しと動的ルーティング機能も可能になります。認可とルーティングのために CDNs 「」を参照してください [MediaTailor マニフェストクエリパラメータ](#)。設定エイリアスを使用した動的広告サーバーとオリジンルーティングについては、「」を参照してください [ADS リクエストの MediaTailor 動的広告変数](#)。

ビューワーと MediaTailor の間に CDN を配置すると、次の利点があります。

- 視聴者に近いエッジロケーションからコンテンツを提供することでレイテンシーを低減する
- CDN のグローバルインフラストラクチャ全体に負荷を分散することでスケーラビリティを向上させる
- 冗長な配信パスによる信頼性の向上

- オリジントラフィックを削減してコストを最適化する
- Media Quality-Aware Routing (MQAR) などの高度な機能を実装してストリーミング品質を向上させる

トピック

- [ニーズに適した CDN を選択する](#)
- [の CDN 統合を計画する AWS Elemental MediaTailor](#)
- [MediaTailor との CDN 統合を設定する](#)
- [パーソナライズされた動画広告用に CDN で SSAI をセットアップする](#)
- [チャンネルアセンブリと CDN を使用して MediaTailor リニアチャンネルを構築する](#)
- [MediaTailor を MediaPackage および CDN と統合する](#)
- [Amazon CloudFront AWS Elemental MediaTailor との統合](#)
- [MediaTailor 広告配信用のサードパーティー CDN を設定する](#)
- [CDN と MediaTailor の統合のパフォーマンス最適化ガイド](#)
- [MediaTailor CDN オペレーションとパフォーマンスのモニタリング](#)
- [CDN と MediaTailor の統合のテストと検証](#)
- [MediaTailor と CDN の統合に関する問題のトラブルシューティング](#)
- [MediaTailor の CDN 統合ログ分析とエラーコードリファレンス](#)
- [で MediaTailor と CDN を自動化する CloudFormation](#)
- [MediaTailor の本番稼働対応 CloudFront 設定](#)
- [CDN と MediaTailor の統合のサポートとトラブルシューティングのヘルプを取得する](#)

ニーズに適した CDN を選択する

適切なコンテンツ配信ネットワーク (CDN) プロバイダーを選択することは、でのコンテンツ配信のパフォーマンス、コスト、視聴者エクスペリエンスに影響を与える可能性のある重要な決定です。AWS Elemental MediaTailor。MediaTailor 実装の CDN を選択するときは、次の要素を考慮してください。

地理的カバレッジ

対象者がいるリージョンに強いプレゼンスを持つ CDN を選択します。異なる CDN プロバイダーは、異なる地理的リージョンでさまざまな強みを持っています。

AWS サービスとの統合

Amazon CloudFront は MediaTailor やその他の AWS サービスとの最も緊密な統合を提供するため、セットアップと管理を簡素化できます。サードパーティー CDN、特殊なビデオ配信機能や特定のリージョンでのプレゼンスの強化など、他の利点を提供する可能性があります。

動画固有の機能

アダプティブビットレートの最適化、ビデオ圧縮、視聴者エクスペリエンスに焦点を当てた分析など、ビデオ配信専用に設計された機能を提供する CDN を探します。

コスト構造

トラフィック量、地理的分布、機能要件などの要因を考慮して、プロバイダー間で料金モデルを比較します。一部の CDN では、使用量パターンに沿ったボリューム割引またはコミットされた使用割引が提供されます。

高度な機能のサポート

選択した CDN が、トークン認証、地理的制限、リクエストの折りたたみ、適切なヘッダー転送など、必要な機能をサポートしていることを確認します。

特定の CDN プロバイダーとその MediaTailor との統合の詳細については、以下のリソースを参照してください。

- [CloudFront 統合](#) Amazon CloudFront 用
- [サードパーティー CDN のセットアップ](#) サードパーティー CDN プロバイダー向け

以下のトピックでは、パフォーマンスを最適化するために CDN で MediaTailor を設定するための包括的なガイダンスを提供します。

の CDN 統合を計画する AWS Elemental MediaTailor

の CDN 統合により、ビューワーエクスペリエンスを向上させ、レイテンシーを短縮できます AWS Elemental MediaTailor。コンテンツ配信ネットワーク (CDN) を実装すると、ビューワーに近い場所からコンテンツを配信できます。これにより、ロード時間が短縮され、スケーラビリティが向上し、さまざまな地理的リージョンで一貫した広告配信が可能になります。

で CDN を実装する前に、適切な計画が必要です AWS Elemental MediaTailor。このセクションでは、主要な計画領域について説明します。実際の設定を開始する前に、これらの領域に対処します。これらのステップは、対象者に最適な視聴エクスペリエンスを作成するのに役立ちます。

計画が完了したら、step-by-step [「CDN と MediaTailor の統合」](#) を参照してください。

CDN 計画に影響する可能性のある MediaTailor クォータについては、「」を参照してください [のクォータ AWS Elemental MediaTailor](#)。CloudFront クォータの詳細については、CloudFront デベロッパーガイド」の「クォータ」を参照してください。

計画を開始する前に、MediaTailor が CDN とやり取りする方法を理解してください。

1. ビューワーは MediaTailor から直接ではなく CDN 経由でコンテンツをリクエストします
2. CDN はマニフェストリクエストを MediaTailor に転送してパーソナライズします。
3. CDN は、エッジロケーションからコンテンツセグメントと広告セグメントをキャッシュして提供します。

このアーキテクチャは MediaTailor の負荷を軽減し、視聴者が最小限のレイテンシーでパーソナライズされた広告を受信できるようにします。

マニフェストの用語を理解すると、CDN を正しく設定するのに役立ちます。異なるストリーミングプロトコルは、キャッシュとルーティングの設定方法に影響する特定のマニフェスト構造を使用します。

- HLS マニフェスト - HLS ストリームを使用する場合は、以下を処理します。
 - 多変量プレイリスト: これらの最上位マニフェストをパーソナライズのために MediaTailor にルーティングするように CDN を設定します。
 - メディアプレイリスト: コンテンツセグメントへのリンクを含むこれらのマニフェストに適切なキャッシュルールを設定します。
- DASH マニフェスト - DASH ストリームを使用する場合は、以下を処理します。
 - MPD (メディアプレゼンテーションの説明): パーソナライズの要件に従ってこれらのマニフェストを処理するように CDN を設定します。

CDN 計画プロセスには、それぞれ特定のタスクに焦点を当てた以下の主要なステップが含まれます。

- [CDN と MediaTailor の統合のトラフィック要件を見積もる](#): 予想されるビューワーの同時実行数と帯域幅要件を計算します。
- [CDN と MediaTailor の統合の最適化戦略を設定する](#): 最適なコンテンツ配信と広告のパーソナライズのために CDN を設定します。

- [CDN と MediaTailor の統合の計画をカスタマイズする](#): 特定の MediaTailor ワークフローに基づいて CDN 戦略を調整します。
- [CDN と MediaTailor の統合のモニタリングとスケーリングを設定する](#): 信頼性の高いパフォーマンスのためのモニタリングおよびスケーリング戦略を実装します。
- [CDN と MediaTailor の統合のコストを最適化する](#): パフォーマンスとコスト効率のバランスを取ります。
- [CDN と MediaTailor の統合の実装をテストする](#): 本番デプロイの前に CDN 統合を徹底的にテストします。

CDN と MediaTailor の統合のトラフィック要件を見積もる

とのコンテンツ配信ネットワーク (CDN) 統合を正確にサイズ設定するには AWS Elemental MediaTailor:

1. 履歴データまたは同様のイベントを使用して、予想されるビューワーの同時実行数を計算します。予期しないスパイクに対応するため、ベースラインを超える追加の容量を計画します。現在のスケーリングの推奨事項については、AWS アカウントチームにお問い合わせください。また、[のクォータ AWS Elemental MediaTailor](#)を表示することもできます。
2. コンテンツスケジュールのピークトラフィックパターンと潜在的なスパイクを特定します。次のような要素を考慮してください。
 - ライブスポーツイベントまたはシーズン初心者向け
 - マーケティングキャンペーンまたはプロモーションイベント
 - グローバルオーディエンスのタイムゾーンの違い
 - 祝祭日または季節的な表示パターン
3. ビューワー数にストリームビットレートを乗算して、帯域幅要件を決定します。CDN プロバイダーと協力して、適切な容量を計算します。この計算は、特定のコンテンツのビットレートと予想されるオーディエンスサイズに基づいて行われます。プロバイダーが推奨する広告セグメントとマニフェストリクエストのオーバーヘッドを追加します。
4. CDN プロバイダーと協力して、ターゲットリージョンに十分なエッジキャパシティを確保します。

以下の特定のアクションを実行することで、広告挿入容量が視聴者の需要を満たしていることを確認します。

1. [Service Quotas コンソール](#)で現在の広告挿入リクエストのクォータを確認します。現在のサービス制限を確認して、設定でサポートできる同時ビューワーの数を確認します。
2. トラフィックの多いイベントの場合は、[Service Quotas コンソール](#)を使用してクォータの引き上げをリクエストします。
3. 500,000 人を超える同時視聴が予想される場合は、イベントの少なくとも 2 週間前に [AWS サポート](#) にお問い合わせください。これにより、AWS は広告パーソナライゼーションのニーズに十分な容量を確保できます。

ワークフローでのキャパシティプランニングの実装の詳細については、[「プリフェッチスケジューリングを使用してトラフィックの多いイベントの広告配信を最適化する」](#)を参照してください。

CDN と MediaTailor の統合の最適化戦略を設定する

トラフィックの推定が完了したら、コンテンツ配信ネットワーク (CDN) を設定して、コンテンツ配信と広告パーソナライゼーションを最適化します AWS Elemental MediaTailor。これらの最適化は、ターゲットを絞った広告を維持しながら、スムーズな再生を確保するのに役立ちます。

以下の特定の CDN 最適化を実装します。

1. MediaTailor の負荷を軽減し、キャッシュ効率を向上させるために、CDN でオリジンシールド機能を設定します。オリジンシールドは、次の中間キャッシュレイヤーとして機能します。
 - 複数のビューワーリクエストを 1 つのオリジンリクエストに統合する
 - MediaTailor への冗長リクエストの数を減らす
 - キャッシュされたコンテンツの応答時間を改善する

CloudFront でオリジンシールドを設定する方法の実装の詳細については、CloudFront デベロッパーガイドの [「オリジンシールドの使用」](#)を参照してください。

2. さまざまなコンテンツタイプに適切な有効期限 (TTL) 値を設定します。TTL は、CDN がコンテンツをキャッシュする期間を決定します。この時間が経過すると、CDN はオリジンから新しいコピーをリクエストします。
 - マニフェスト:
 - 広告挿入に 0 秒
 - チャネルアセンブリの場合は 5 ~ 10 秒

広告挿入では、MediaTailor はビューワーにパーソナライズされた広告を含むマニフェストを提供します。プレイリストまたは MPD がキャッシュされ、間違った再生デバイスに提供されると、デバイスで再生または追跡の問題が発生する可能性があります。

- コンテンツセグメント: 24 時間以上 (ほとんど変更されず、オリジン負荷を減らすために積極的にキャッシュできます)
- 広告セグメント: 24 時間以上 (広告コンテンツは通常ビューワー間で再利用され、長期間キャッシュできます)

すべての MediaTailor ワークフローにわたる包括的な TTL レコメンデーションとキャッシュ最適化戦略については、「」を参照してください[CDN と MediaTailor の統合のキャッシュ最適化](#)。

CloudFront でキャッシュ動作を設定する詳細な手順については、CloudFront デベロッパーガイドの「[キャッシュ動作設定](#)」を参照してください。

3. CDN エッジノードをビューワー母集団の近くにデプロイします。CDN プロバイダーと協力して以下を行います。
 - ビューワーの属性に基づいて最適なエッジノードの場所を特定する
 - 各リージョンで十分な容量を確保する
 - エッジパフォーマンスをモニタリングし、必要に応じて調整する

実装ガイダンスについては、[CloudFront エッジロケーション](#)」を参照して、対象者のリージョンで利用可能なエッジロケーションを特定します。

4. グローバルオーディエンスには、マルチ CDN 戦略の実装を検討してください。このアプローチ:
 - 複数の CDN プロバイダーを使用して信頼性を向上させる
 - ビューワーをその場所の最もパフォーマンスの高い CDN にルーティングします
 - CDN の停止中にフェイルオーバーオプションを提供します
 - さまざまな料金モデルを活用してコストを最適化できる

実装の詳細については、AWS 「ネットワークとコンテンツ配信ブログ」の「[マルチ CDN 戦略](#)」を参照してください。

CDN と MediaTailor の統合の計画をカスタマイズする

ワークフローごとに、コンテンツ配信ネットワーク (CDN) の計画に影響する固有の要件 AWS Elemental MediaTailor があります。ワークフロー固有のガイダンスについては、「[設定の使用](#)」を参照してください。

特定の MediaTailor ワークフローに基づいてキャパシティプランを調整します。

MediaTailor 広告挿入ワークフローの場合

1. キャッシュなしでパーソナライズされたマニフェストを処理するように CDN を設定します。これにより、各ビューワーはターゲットを絞った一意の広告を受信できます。広告は、そのプロファイルと表示コンテキストに基づいています。
2. ピークリクエストボリュームを処理するように広告決定サーバー (ADS) のサイズを設定します。ADS 設定のガイダンスについては、[MediaTailor を使用した広告挿入](#)を参照してください。次の点を考慮してください。
 - ユースケースの応答時間要件
 - 予想される同時ビューワー容量
 - 冗長性とフェイルオーバーの要件
 - 地理的分散のニーズ
3. CDN レベルでリクエストの折りたたみを実装して、同期された広告時間枠リクエストを処理します。リクエストの折りたたみは、複数の同一のリクエストを 1 つのオリジンリクエストに結合します。これは、次の場合に重要です。
 - 多数の視聴者が同時に広告ブレイクにヒットしたときのライブスポーツイベント
 - 人気テレビ番組が同期商用ブレイクで公開
 - 突然の視聴者の急増を引き付ける重大なニュースイベント

実装の詳細については、「[オリジンフェイルオーバー](#)」を参照して、ピーク負荷時のリクエスト処理を設定します。

MediaTailor チャンnelアセンブリワークフローの場合

1. チャンネル数とそのビットレートに基づいて容量要件を計算します。チャンネルアセンブリ容量計画のガイダンスについては、[MediaTailor の「チャンネルアセンブリ」](#)を参照してください。次の点を考慮してください。
 - チャンネルの合計数
 - チャンネルあたりのビットレート要件
 - 予想される同時ビューワーロード
 - 地理的分散のニーズ

2. 公開されたスケジュールに基づいて予測可能なトラフィックパターンを処理するように CDN を設定します。チャンネルアセンブリには通常、広告挿入よりも予測可能なパターンがあります。その理由は次のとおりです。
 - プログラミングスケジュールが事前にわかっている
 - ビューワーの動作は確立されたパターンに従います
 - コンテンツはビューワーごとに動的に変更されません
3. オリジンに、一貫したチャンネル出力を維持するのに十分な帯域幅があることを確認します。実装:
 - 高可用性チャンネル用の冗長オリジンサーバー
 - プライマリオリジンとバックアップオリジン間の自動フェイルオーバー
 - オリジンパフォーマンスの問題を検出するためのモニタリング

実装ガイダンスについては、「[オリジン冗長性を設定](#)」して回復力のあるオリジンインフラストラクチャを作成する」を参照してください。

MediaTailor ワークフローを組み合わせた場合

1. 両方のサービスを組み合わせたトラフィックパターンを処理するようにインフラストラクチャのサイズを設定します。ワークフローの組み合わせに関するガイダンスについては、「」を参照してください。[AWS Elemental MediaTailor を使用して広告を挿入する](#)。次の点を考慮してください。
 - チャンネルアセンブリのベースライン要件
 - 広告挿入のオーバーヘッド要件
 - ピークトラフィックパターン
 - 冗長性のニーズ
2. 線形コンテンツ配信と動的広告挿入用に個別の CDN 動作を設定します。この分離により、次のことが可能になります。
 - コンテンツタイプごとにキャッシュポリシーを個別に最適化する
 - コンテンツタイプに基づいてリクエストを適切なオリジンにルーティングする
 - ワークフローごとにパフォーマンスメトリクスを個別にモニタリングする
3. 最適なパフォーマンスを維持するために、エッジ CDN 間の適切なルーティングを設定します。以下を使用することを検討してください。
 - コンテンツセグメント (/content/*) と広告セグメント (/ads/*) の異なるオリジンパス
 - マニフェストとセグメントのキャッシュ動作を分離する

実装の詳細については、「パースベースのルーティングとキャッシュルールを設定するためのキャッシュ [動作](#) の設定」を参照してください。

CDN と MediaTailor の統合のモニタリングとスケーリングを設定する

AWS Elemental MediaTailor コンテンツ配信ネットワーク (CDN) 統合で最適なパフォーマンスと視聴者エクスペリエンスを維持するには、効果的なモニタリングとスケーリング戦略が不可欠です。CDN 統合が大規模に確実に機能するように、これらのアプローチを実装します。

以下のモニタリングおよびスケーリング戦略を実装します。

1. これらの主要メトリクスのモニタリングを設定します。適切なターゲット値に関するガイダンスについては、[Amazon CloudWatch による MediaTailor のモニタリング](#) および「CDN プロバイダーのベストプラクティス」を参照してください。
 - CDN キャッシュヒット率 (コンテンツタイプと配信パターンに基づいてベースラインメトリクスとターゲットを確立する)
 - オリジンリクエストボリューム (異常検出のベースラインを確立するために、通常のオペレーション中にパターンをモニタリングする)
 - エラータイプ別のエラー率 (サービスレベルの目標と MediaTailor のベストプラクティスに基づいてしきい値を定義する)
 - 応答時間 (ビューワーエクスペリエンスの要件と地理的分布に基づいて適切なレイテンシーターゲットを設定する)

実装手順の詳細については、「[MediaTailor と CDN メトリクスを一緒に視覚化する CloudWatch ダッシュボードの作成](#)」を参照してください。MediaTailor

2. 予期しないトラフィックパターンやパフォーマンスの低下に関するアラートを設定します。ベースラインメトリクスとサービスレベルの目標に基づいてしきい値を設定します。アラートの設定に関するガイダンスについては、[Amazon CloudWatch アラームの作成](#)」を参照してください。モニタリングを検討してください。
 - ベースラインキャッシュヒット率からの大幅な偏差 (通常は 85~90% を下回ると警告)
 - オリジンリクエストボリュームの急激な増加 (ベースラインから 30% 以上の増加に関するアラート)
 - 定義されたしきい値を超えるエラー率の急増 (通常、4xx エラーの場合は 1~2%、5xx エラーの場合は 0.5%)

- 許容レベルを超える応答時間の低下 (通常はマニフェストの場合は >500 ミリ秒、セグメントの場合は >200 ミリ秒)

実装例については、効果的なモニタリングダッシュボードとアラートを作成するための [CloudWatch の概念](#) を参照してください。

3. 予測可能なトラフィックの多いイベントのスケーリングプランを作成します。計画には、次の主要な要素を含める必要があります。

- イベント前のキャパシティーの増加 (イベント開始の 24 ~ 48 時間前)
- 段階的な視聴者ランプアップスケジュール (通常、5 分間隔で予想される視聴者の 10 ~ 20%)
- 対象者に基づくリージョン別キャパシティー分布 (予想されるリージョン別視聴率に比例してキャパシティーを割り当てる)
- イベント後のスケーリング手順 (イベント終了後 30 ~ 60 分間ピーク容量を維持する)

トラフィックの多いイベントのスケーリングに関する実装ガイダンスについては、AWS メディアブログの「[回復力のある end-to-end のライブワークフローのセットアップ](#)」を参照してください。

4. 次のような重要なストリームにフェイルオーバーと冗長性対策を実装します。

- マルチリージョン CDN デプロイ (重要なコンテンツの場合は少なくとも 2 つのリージョン)
- バックアップオリジンサーバー (30 秒ごとに自動ヘルスチェックで設定)
- ヘルスチェックに基づく自動フェイルオーバートリガー (通常はチェックが 2 ~ 3 回失敗した後)
- さまざまな障害シナリオの復旧手順 (特定の応答時間目標とともに文書化)

実装手順の詳細については、[CloudFront オリジンフェイルオーバーによる高可用性の最適化](#)」を参照してください。

CDN と MediaTailor の統合のコストを最適化する

コンテンツ配信ネットワーク (CDN) のコストは、トラフィックパターン、地理的分布、による機能の使用状況によって大きく異なる場合があります AWS Elemental MediaTailor。現在の料金情報については、[CloudFront の料金](#)」を参照してください。CDN プロバイダーのドキュメントを参照することもできます。

以下の戦略を使用して、パフォーマンスとコスト効率のバランスを取ります。

1. CDN トラフィックパターンを分析して、最も費用対効果の高い料金階層を選択します。CDN プロバイダーで次の要因を確認します。

- リージョンと期間別のデータ転送ボリューム
- マニフェストとセグメントのリクエストパターン
- 地理的分散の要件
- ピーク使用パターンと平均使用パターン

コスト分析については、[AWS 料金計算ツールを使用して](#)、特定の使用パターンに基づいて CloudFront のコストを見積もります。

2. 予測可能なワークロードの場合は、CDN プロバイダーとのリザーブドキャパシティ契約を評価します。これらの契約には次のような利点があります。

- コミットされた使用量の割引料金
- 予算の予測可能な月額コスト
- 優先度のサポートと容量の割り当て

CDN プロバイダーに相談して、リザーブドキャパシティが使用パターンに適しているかどうかを判断します。CloudFront については、[CloudFront のプレミアム機能](#)を参照してください。これにより、リザーブドキャパシティオプションに関する情報が提供されます。

3. MediaTailor と CDN プロバイダー間のトラフィックを分散することで、エグレスコストを最適化します。戦略には以下が含まれます。

- キャッシュヒット率を最大化してオリジンリクエストを減らす
- オリジンシールドを使用してリクエストを統合する
- 圧縮を実装してデータ転送ボリュームを減らす
- リージョンの料金に合った CDN AWS リージョンを選択する

圧縮に関する実装ガイダンスについては、CloudFront デベロッパーガイドの[「圧縮ファイルの供給」](#)を参照してください。

4. オリジンリクエストを減らすために、さまざまなコンテンツタイプに適したキャッシュ戦略を実装します。キャッシュ最適化のガイダンスについては、[「キャッシュヒット率の向上」](#)を参照してください。以下に焦点を当てます。

- コンテンツセグメント (長期間キャッシュできます)
- 広告セグメント (通常はビューワー間で再利用)
- プレイヤーファイルやイメージなどの静的アセット

キャッシュヒット率の向上により、オリジンコストが大幅に削減されます。CDN プロバイダーと協力して、特定のコンテンツパターンのキャッシュ設定を最適化します。詳細な実装手順については、[「キャッシュ動作の設定」](#)を参照してください。

CDN と MediaTailor の統合の実装をテストする

本番デプロイ前に AWS Elemental MediaTailor コンテンツ配信ネットワーク (CDN) 統合を徹底的にテストすることで、信頼性の高い視聴者エクスペリエンスを確保します。適切なテストは、対象者に影響を与える前に問題を特定して解決するのに役立ちます。テスト方法のガイダンスについては、[CloudFront ディストリビューションのテスト](#)を参照してください。CDN プロバイダーのテストドキュメントを参照することもできます。

CDN 統合を検証するテストステップに従います。

1. 本番稼働用設定をミラーリングするテスト環境を作成します。インクルード:

- 同一の CDN 設定とキャッシュ動作
- さまざまなビットレートと形式の代表的なコンテンツを含める
- サンプル広告レスポンスを使用してテスト広告決定サーバーを設定する
- モニタリングとアラートの設定をセットアップする

step-by-step実装ガイダンスについては、CloudFront [デベロッパーガイド](#)の「[ステージングディストリビューションの作成](#)」を参照してください。

2. 負荷テストを実行して、容量の見積もりを検証します。負荷テストのガイダンスについては、[MediaPackage のモニタリング](#)を参照してください。テストシナリオには以下が含まれます。

- ビューワーステージの段階的な増加 (通常は 5 分ごとに予想されるピークの 10~20%)
- 予想されるピーク負荷に基づく突然のトラフィックの急増 (ピークオーディエンスの 50% が 60 秒以内に参加することをシミュレート)
- 延長ピーク負荷期間 (30~60 分以上ピーク負荷を維持する)
- 対象者に一致する地理的分散 (予想される視聴者の場所に従ってテストトラフィックを分散)

応答時間が目標しきい値を下回っていることを検証します。通常、これはマニフェストの場合は 500 ミリ秒未満、セグメントの場合は 200 ミリ秒未満を意味します。エラー率は 1% 未満に維持する必要があります。負荷テストツールと方法の実装の詳細については、AWS「ネットワークとコンテンツ配信ブログ」の[CloudFront を使用した負荷テスト](#)を参照してください。

3. フェイルオーバーシナリオをテストして信頼性を確保します。シミュレート:

- オリジンサーバーの障害 (完全な停止シナリオと部分的な低下シナリオ)
- CDN エッジロケーションの停止 (バックアップロケーションへのトラフィックルーティングでテスト)
- 広告決定サーバーの非可用性 (5 ~ 10 秒のタイムアウトでテスト)
- ネットワーク接続の問題 (パケット損失とレイテンシーの増加をシミュレート)

CDN プロバイダーと協力して、ユースケースに適したフェイルオーバー応答時間目標を設定します。通常、これはフェイルオーバーが完了するまで 3 秒未満であることを意味します。フェイルオーバーテストの実装ガイダンスについては、CloudFront デベロッパーガイドの「[オリジンフェイルオーバー](#)」を参照してください。

4. 主要なイベントについては、[AWS 負荷テストガイドライン](#)に基づいて段階的なランプアップ戦略を実装します。

- 異なるオーディエンスセグメントの開始時間をずらします。たとえば、オーディエンスグループ間で 15 分間隔でプレミアムサブスクライバーを最初に許可します。
- 事前ウォーミング手法を使用して負荷を徐々に増やします。事前ウォーミングには以下が含まれます。
 - イベントの 2 ~ 4 時間前に CDN キャッシュに一般的なコンテンツを入力する
 - 合成トラフィックを予想ピークの 20 ~ 30% まで徐々に増やしてシステムをウォームアップする
 - 実際のコンテンツを使用して、現実的な負荷条件ですべてのコンポーネントをテストする
- ランプアップ期間中のシステムパフォーマンスをモニタリングし、以下を追跡します。
 - キャッシュヒット率と応答時間 (ターゲット >90% ヒット率、<500ms 応答)
 - エラー率とオリジン負荷 (エラー率 <1%、オリジン CPU <70%)
 - 広告パーソナライゼーションの成功率 (ターゲット >98% の成功パーソナライゼーション)
 - ビューワーエクスペリエンスメトリクス (ターゲット <2 秒の起動時間、<0.5% の再バッファリング)
- 予期しないトラフィックの急増に備えて、緊急時対応計画を立てます。プランには、以下の重要なコンポーネントを含める必要があります。
 - 15 分以内に容量を 50 ~ 100% 増やす手順を文書化した緊急キャパシティスケーリング手順
 - トラフィックの 20 ~ 50% をセカンダリ CDN に移行する機能を備えたバックアップ CDN アクティベーションプロトコル

- 広告挿入のフォールバックを簡素化し、ターゲットパラメータを 10 個以上から 3~5 個の必須パラメータに減らす
- 事前定義された通知テンプレートと連絡先リストを持つステークホルダー向けのコミュニケーションプラン

テストが完了したら、本番環境のデプロイ手順の [CDN 統合の実装](#)に進みます。

CDN と MediaTailor の統合に関する一般的な問題のトラブルシューティング

ビューワーに影響を与える AWS Elemental MediaTailor 前に、一般的なコンテンツ配信ネットワーク (CDN) 統合の課題に対処します。このセクションでは、この CDN 統合中に発生する一般的な問題を特定して解決するのに役立ちます AWS Elemental MediaTailor。

包括的なトラブルシューティングガイダンスについては、[MediaTailor のトラブルシューティング](#)と[CloudFront デイストリビューションのトラブルシューティング](#)を参照してください。

MediaTailor マニフェスト配信の問題の解決

視聴者に再生の問題が発生したり、誤った広告が表示されたりする場合は、以下の一般的なマニフェスト関連の問題を確認してください。

- 不正なキャッシュ設定: CDN がパーソナライズされたマニフェストをキャッシュする場合、ビューワーには他のユーザー向けの広告が表示されることがあります。

解決策: MediaTailor へのマニフェストリクエストのキャッシュ TTL を 0 に設定します。

- オリジンリクエストの失敗: CDN が MediaTailor に到達できない場合、マニフェストリクエストは失敗します。

解決策: CDN と MediaTailor 間のネットワーク接続を確認します。CDN が正しいヘッダーを適切に転送していることを確認します。

- セッションパラメータの問題: セッションパラメータがないか、正しくないと、パーソナライゼーションが失敗する可能性があります。

解決策: プレイヤーがマニフェストリクエストに必要なすべてのセッションパラメータを正しく追加していることを確認します。

MediaTailor セグメント配信の問題の修正

コンテンツまたは広告セグメントが正しくロードされない場合は、次の一般的な問題を調査してください。

- セグメントパスの書き換え: CDN 設定が正しくないと、セグメント URLs 不適切に書き換えられる可能性があります。

解決策: CDN がセグメント URLs。参照を壊すような方法でパスを変更しないことを確認します。

- CORS 設定: CORS ヘッダーがないか正しくないと、ブラウザがセグメントをロードできなくなる可能性があります。

解決策: セグメントリクエストに適切な CORS ヘッダーを渡すように CDN を設定します。

- キャッシュミスストーム: トラフィックの多いイベントでは、複数のキャッシュミスがオリジンサーバーを圧迫する可能性があります。

解決策: リクエストの折りたたみとオリジンシールド機能を実装して、トラフィックの急増時のオリジン負荷を軽減します。

MediaTailor CDN パフォーマンスの問題に対処する

ビューワーがバッファリングや読み込みが遅い場合は、以下のパフォーマンス関連の問題を確認してください。

- キャッシュヒット率が低い: CDN がオリジンからコンテンツを頻繁にリクエストする場合、パフォーマンスが低下します。

解決策: キャッシュヒット率をコンテンツタイプ別に分析し、TTL 設定を調整してキャッシュ効率を向上させます。

- 地理的分布: CDN エッジロケーションから離れたビューワーでは、レイテンシーが増加する可能性があります。

解決策: CDN エッジロケーションディストリビューションを確認します。ビューワーの濃度が高いリージョンに容量を追加します。

- オリジン容量の制限: オリジンサーバーが過負荷になっている場合、応答時間が長くなります。

解決策: オリジンリクエストの制限を実装します。オリジン容量を増やしたり、キャッシュを改善してオリジンの負荷を減らすこともできます。

トラブルシューティングに関するその他のサポートについては、[MediaTailor のトラブルシューティング](#)」を参照するか、AWS サポートにお問い合わせください。一般的な CDN 問題の解決に関する実装ガイダンスについては、AWS メディアブログの「[コンテンツ配信ネットワークのデバッグ](#)」を参照してください。

MediaTailor との CDN 統合を設定する

このセクションでは、コンテンツ配信ネットワーク (CDN) AWS Elemental MediaTailor と統合するためのガイダンスを提供します。

MediaTailor との効果的な CDN 統合は、パーソナライズされた広告で高品質のストリーミングエクスペリエンスを大規模に提供するために不可欠です。このガイドでは、CDN 統合をセットアップ、設定、最適化する完全なプロセスについて説明します。

詳細については、次のリンクを参照してください。

- 認可と広告ターゲティングのために CDN 「」を参照してください[MediaTailor マニフェストクエリパラメータ](#)。
- 動的変数と設定エイリアスを使用した高度なルーティングについては、「」を参照してください[ADS リクエストの MediaTailor 動的広告変数](#)。
- MediaTailor 設定の作成については、「」を参照してください[AWS Elemental MediaTailor を使用して広告を挿入する](#)。
- CloudFront ディストリビューションの作成の詳細については、CloudFront [デベロッパーガイド](#)の「[ディストリビューションの作成](#)」を参照してください。

このガイドを明確にするために、さまざまなタイプのマニフェストを説明するために次の用語が使用されます。

- HLS マニフェスト:
 - 多変量プレイリスト: メディアプレイリストへのリンクを含む最上位マニフェスト
 - メディアプレイリスト: コンテンツセグメントへのリンクを含む第 2 レベルのマニフェスト
- DASH マニフェスト:
 - MPD (メディアプレゼンテーションの説明): DASH マニフェストの標準用語

MediaTailor を CDN と統合すると、次の利点があります。

- ビューワーのレイテンシーの短縮
- トラフィックの多いイベントのスケラビリティの向上
- 冗長な配信パスによる信頼性の向上
- オリジントラフィックを削減してコストを最適化
- DDoS 攻撃に対する保護の強化

CDN 統合コンポーネントと要件

MediaTailor との CDN 統合を成功させるには、次の主要コンポーネントを設定する必要があります。

CDN ルーティングの動作

さまざまなタイプのリクエスト (マニフェスト、コンテンツセグメント、広告セグメント) が CDN 経由でルーティングされる方法を決定するルール。

MediaTailor での CDN マッピング

オリジンサーバーを直接参照するのではなく、マニフェストが CDN ドメインを参照することを保証する MediaTailor の設定。

セキュリティ設定

トランスポートセキュリティ、アクセスコントロール、モニタリングなど、コンテンツとインフラストラクチャを保護する設定。

テストと検証

本番環境にデプロイする前に CDN 統合が正しく動作していることを検証する手順。

CDN 統合の前提条件

CDN 統合を設定する前に、以下があることを確認してください。

1. 次の設定の MediaTailor 設定。

- コンテンツソースとしてのコンテンツオリジン
- 広告決定サーバーとしての ADS

CDN 統合ステップにもオリジン URL と ADS URLs が必要です。

2. CDN の設定インターフェイスへのアクセス
3. 動作、ルール、キャッシュ設定に関する CDN 固有の用語を理解する
4. セグメントに使用されるファイル拡張子 (.ts、.mp4、.m4s など) を含むコンテンツ構造に関する知識

CDN 統合のセットアップ手順

MediaTailor を CDN と統合するプロセスは、以下の大まかなステップに従います。

1. CDN ルーティング動作を設定する - さまざまなタイプのリクエストを適切にルーティングするように CDN を設定します。
2. MediaTailor で CDN マッピングを設定する - CDN ドメイン名を使用するように MediaTailor 設定を更新します。
3. セキュリティのベストプラクティスを実装する - コンテンツとインフラストラクチャを保護するためにセキュリティ設定を構成します。
4. 統合をテストする - CDN 統合が正しく動作していることを確認します。

MediaTailor CDN 統合に必要なヘッダー

MediaTailor が CDN で正しく機能するには、特定の HTTP ヘッダーを転送するように CDN を設定する必要があります。これらのヘッダーは、圧縮、デバイス検出、広告パーソナライゼーション、ジオターゲティングなどの適切な機能に不可欠です。

次のヘッダーを MediaTailor に転送するように CDN を設定します。

Accept-Encoding

目的: 圧縮機能に必要

詳細: このヘッダーは、クライアントがサポートする圧縮方法を MediaTailor に伝えます。MediaTailor はこの情報を使用して、可能な限りマニフェストを圧縮し、帯域幅の使用量を減らし、パフォーマンスを向上させます。圧縮をサポートしていないレガシーデバイスはこのヘッダーを送信せず、MediaTailor は非圧縮マニフェストを返します。

User-Agent

目的: デバイスの検出と広告のパーソナライズに必要

詳細: MediaTailor は User-Agent ヘッダーを分析して、クライアントデバイスタイプ、ブラウザ、機能を特定します。この情報は、広告ターゲティング、デバイス固有の最適化、さまざまな再生クライアントとの互換性の確保に使用されます。

Host

目的: 適切なリクエストルーティングに必要です

詳細: Hostヘッダーにより、リクエストが正しい MediaTailor エンドポイントにルーティングされます。これは、マルチテナント環境やカスタムドメイン設定を使用する場合に特に重要です。

Amazon CloudFront を含む多くの CDNs、デフォルトでHostヘッダーを転送しません。CloudFront ユーザーの場合: 設定手順[キャッシュ動作の設定](#)については、「」を参照してください。

X-Forwarded-For

目的: クライアント IP 検出とジオターゲティングに必要

詳細: このヘッダーは、リクエストが CDN を通過するときに元のクライアント IP アドレスを保持します。MediaTailor は、この情報を地理的広告のターゲティング、分析、リージョンのコンテンツ制限への準拠に使用します。

Important

MediaTailor の全機能には、4 つのヘッダーすべてが必要です。これらのヘッダーがないと、次のような機能が低下する可能性があります。

- マニフェストを圧縮できない (Accept-Encoding がない)
- 不適切な広告ターゲティングとデバイスの互換性の問題 (User-Agent の欠落)
- リクエストルーティングの失敗 (ホストの欠落)
- 不正確なジオターゲティングと分析 (X-Forwarded-For の欠落)

CDN 固有の設定手順については、このヘッダーリストを参照するルーティング動作とキャッシュセクションを参照してください。

以下のトピックでは、CDN 統合の各側面の詳細な手順について説明します。

トピック

- [MediaTailor の CDN ルーティング動作を設定する](#)
- [MediaTailor で CDN マッピングを設定する](#)
- [MediaTailor の CDN 統合セキュリティのベストプラクティス](#)

MediaTailor の CDN ルーティング動作を設定する

このセクションでは、AWS Elemental MediaTailor 統合のためにさまざまなタイプのリクエストを適切にルーティングするようにコンテンツ配信ネットワーク (CDN) を設定する方法について説明します。適切なルーティング設定により、マニフェストリクエスト、コンテンツセグメント、広告セグメントが正しく処理されます。

CDN ルーティング動作の設定は、効率的なコンテンツ配信パイプラインを作成する上で重要なステップです。さまざまなコンテンツタイプに特定のルーティングルールを設定することで、キャッシュを最適化し、配信パフォーマンスを向上させ、パーソナライズされた広告挿入が正しく機能するようにできます。

動変数と設定エイリアスを使用する高度なルーティングシナリオについては、「」を参照してください [ADS リクエストの MediaTailor 動的広告変数](#)。CDN ルーティング全体でクエリパラメータを保持する方法については、「」を参照してください [MediaTailor マニフェストクエリパラメータ](#)。

Important

キャッシュされたオブジェクトまたはビューワーへの CDN レスポンスに CORS ヘッダーを含めると、再生が失敗する可能性があります。

CDN ルーティング動作設定

さまざまなタイプのリクエストを適切にルーティングするように CDN を設定します。

コンテンツセグメントのルーティング

コンテンツセグメントルーティングは、実際のコンテンツセグメントのリクエストをオリジンサーバーに送信します。広告セグメントルーティングと同様に、コンテンツセグメントルーティングには、ウェブベースのプレイヤーでスムーズに再生できるように適切な CORS 設定も必要です。

詳細な設定ガイダンスについては、の CloudFront の例を参照してください [優先順位 4: コンテンツオリジンパスの動作](#)。この例では、CloudFront で従うか、他の CDNs。

コンテンツセグメントルーティングの主な設定要件は次のとおりです。

- コンテンツセグメントファイル拡張子 (*.ts、*.mp4など*.m4s) に一致するパスパターンを使用する
- リクエストをコンテンツオリジン (Amazon S3 バケットや MediaPackage エンドポイントなど) にルーティングする
- 最適なキャッシュヒット率を得るには、オリジンがキャッシュキーのレスポンスを変更し、オリジンリクエストを転送する原因となるクエリ文字列パラメータのみを含めます。
- 24 時間を超える TTL 値を持つ適切なキャッシュポリシーを適用する
- CORS レスポンスヘッダーをビューワーに含める

広告セグメントのルーティング

広告セグメントルーティングは、パーソナライズされた広告を視聴者に配信するために不可欠です。広告セグメントルーティングを設定するときは、ウェブベースのプレーヤーで再生失敗の原因となる問題を防ぐために、適切な CORS (Cross-Origin Resource Sharing) 処理を実装する必要があります。

詳細な設定ガイダンスについては、の CloudFront の例を参照してください[優先順位 0: 広告セグメントのパス動作](#)。この例では、CloudFront で従うか、他の CDNs。

広告セグメントルーティングの主な設定要件は次のとおりです。

- MediaTailor 広告セグメント/tm/*専用のパスパターンを使用する
- へのリクエストのルーティング `segments.mediatailor.region.amazonaws.com`
- 最適なキャッシュヒット率を得るには、ビューワーリクエストヘッダー、Cookie、またはクエリ文字列パラメータをキャッシュキーまたはオリジンリクエストに含めないでください。
- 24 時間を超える TTL 値を持つ適切なキャッシュポリシーを適用する
- CORS レスポンスヘッダーをビューワーに含める

マニフェストリクエストルーティング

多変量プレイリスト、メディアプレイリスト、MPD リクエストを MediaTailor にルーティングするには、次の一般的な設定を使用します。CloudFront の設定については、「」を参照してください[マニフェストキャッシュ動作の設定](#)。

1. CDN 設定インターフェイスで、さまざまなマニフェストタイプの動作を作成します。

2. マルチバリエーションプレイリストとメディアプレイリストのファイル拡張子 (*.m3u8HLS の場合) と MPD ファイル拡張子 (*.mpdDASH の場合) に一致するようにパスパターンを設定します。
3. MediaTailor 設定エンドポイントを指すように CDN のオリジン設定を構成します。
4. 広告挿入の場合、パーソナライズされたマルチバリエーションプレイリスト、メディアプレイリスト、MPDs。広告挿入はパーソナライズされたマニフェストを提供するため、CDN はマニフェストをキャッシュしないでください。意図したものとは異なる再生デバイスがキャッシュされたプレイリストまたは MPD を受信すると、再生または追跡の問題が発生する可能性があります。すべてのコンテンツタイプの TTL レコメンデーションを含む包括的なキャッシュガイドランスについては、「」を参照してください[CDN と MediaTailor の統合のキャッシュ最適化](#)。
5. すべてのヘッダーのヘッダー転送を設定します。最小要件については、「」を参照してください[MediaTailor CDN 統合に必要なヘッダー](#)。
6. クエリ文字列転送を有効にして、広告ターゲティングパラメータを渡します。

HLS multivariant playlist

HLS マルチバリエーションプレイリストリクエストは、次の形式に従います。

```
https://<playback-endpoint>/v1/index/<hashed-account-id>/<origin-id>/<index>.m3u8
```

例:

```
https://777788889999.mediataylor.us-east-1.amazonaws.com/v1/master/  
a1bc06b59e9a570b3b6b886a763d15814a86f0bb/Demo/assetId.m3u8
```

HLS media playlist

HLS メディアプレイリストリクエストは次の形式に従います。

```
https://<playback-endpoint>/v1/manifest/<hashed-account-id>/<session-id>/  
<manifestNumber>.m3u8
```

へのプレーヤーリクエスト `https://CDN_Hostname/some/path/asset.m3u8` は、リクエストのキーワード `https://mediataylor.us-west-2.amazonaws.com/v1/session/configuration/endpoint` に基づいて MediaTailor パス *.m3u8 にルーティングされます。

例:

```
https://777788889999.mediatailor.us-east-1.amazonaws.com/v1/manifest/  
a1bc06b59e9a570b3b6b886a763d15814a86f0bb/c240ea66-9b07-4770-8ef9-7d16d916b407/0.m3u8
```

DASH MPD

DASH MPD リクエストは次の形式に従います。

```
https://<playback-endpoint>/v1/dash/<hashed-account-id>/<origin-id>/<assetName>.mpd
```

へのプレイヤーリクエスト `https://CDN_Hostname/some/path/asset.mpd` は、リクエストのキーワード `https://mediatailor.us-west-2.amazonaws.com/v1/dash/configuration/endpoint` に基づいて MediaTailor パス*.mpd にルーティングされます。

例:

```
https://777788889999.mediatailor.us-east-1.amazonaws.com/v1/dash/  
a1bc06b59e9a570b3b6b886a763d15814a86f0bb/Demo/0.mpd
```

CDN ルーティングのベストプラクティス

CDN ルーティング動作を設定するときは、次のベストプラクティスに従って最適なパフォーマンスと信頼性を確保します。

特定のパスパターンを使用する

コンテンツ構造に正確に一致する特定のパスパターンを作成して、適切なルーティングを確保します。

動作順序の優先順位付け

ほとんどの CDNs、動作は順番に評価されます。より一般的な動作の前に、より具体的な動作を配置します。

テスト動作パターン

本番環境にデプロイする前に、パスパターンが予想されるリクエストと正しく一致していることを確認します。

設定を文書化する

CDN ルーティング動作のドキュメントを維持し、トラブルシューティングと今後の更新を容易にします。

次の手順

CDN ルーティング動作を設定したら、次のステップとして MediaTailor で CDN マッピングを設定します。手順については、「[MediaTailor で CDN マッピングを設定する](#)」を参照してください。

MediaTailor で CDN マッピングを設定する

このセクションでは、コンテンツ配信ネットワーク (CDN) ドメイン名を使用する AWS Elemental MediaTailor 用に を設定する方法について説明します。CDN ルーティング動作を設定したら、オリジンサーバーを直接参照するのではなく、マニフェストが CDN ドメインを参照するように MediaTailor 設定を更新する必要があります。

MediaTailor で CDN マッピングを設定すると、マニフェスト内のすべてのコンテンツと広告セグメント URLs、オリジンサーバーに直接ではなく CDN を指すようになります。このステップは、完全な CDN 配信チェーンを作成し、CDN 統合の利点を最大化するために不可欠です。

MediaTailor での CDN マッピング設定

CDN ルーティング動作を設定したら、CDN ドメイン名を使用するように MediaTailor を設定します。

1. [MediaTailor コンソール](#)を開きます。
2. 更新する設定を選択します。
3. CDN 設定の詳細設定セクションで、CDN コンテンツセグメントプレフィックスフィールドに CDN ドメイン名を入力します。
4. 広告セグメントに別の CDN ドメインを使用している場合は、CDN 広告セグメントプレフィックスフィールドに入力します。
5. 変更内容を保存します。

この設定により、MediaTailor はオリジンまたは広告セグメントストレージを直接参照するのではなく、CDN を指す URLs を持つマニフェストを生成します。

基本 URL の動作について

MediaTailor URLs を決定します。

- CDN 広告セグメントプレフィックスの設定: 広告セグメントは CDN プレフィックスをベース URL として使用します。

- CDN 広告セグメントプレフィックスが設定されていません: 広告セグメントは直接 MediaTailor の場所をベース URL として使用します。
- CDN コンテンツセグメントプレフィックスの設定: コンテンツセグメントは CDN プレフィックスをベース URL として使用します。
- CDN コンテンツセグメントプレフィックスが設定されていません: コンテンツセグメントは元のコンテンツオリジンを参照します。

DASH BaseURL の処理

DASH マニフェストの場合、MediaTailor はコンテンツセグメントと広告セグメントBaseURLの設定を異なる方法で管理します。

コンテンツセグメント:

- CDN コンテンツセグメントプレフィックスの場合: MediaTailor は、指定したプレフィックスを使用してMPDレベルBaseURLに 1 つだけを設定します。
- CDN コンテンツセグメントプレフィックスなし: MediaTailor はオリジンマニフェストから既存のBaseURL設定を保持するか、存在しない場合はオリジン MPD URL に基づいて設定を追加します。

広告セグメント:

- CDN 広告セグメントプレフィックスの場合: 各広告期間は、設定されたプレフィックスBaseURLを使用して 1 つだけ取得されます。
- CDN 広告セグメントプレフィックスなし: 各広告期間は MediaTailor 広告コンテンツサーバーBaseURLを指す 1 つだけ取得されます。

Example CDN マッピングの例

コンテンツオリジンが `http://origin.com/contentpath/` で、CDN コンテンツセグメントプレフィックスが の場合 `https://cdn.example.com/`、通常は として参照されるコンテンツセグメント `http://origin.com/contentpath/subdir/content.ts` がマニフェストに として表示されます `https://cdn.example.com/subdir/content.ts`。

重要な考慮事項

MediaTailor で CDN マッピングを設定するときは、次の重要な考慮事項に注意してください。

CDN プレフィックスに HTTPS を使用する

CDN プレフィックスには常に HTTPS URLs を使用して、安全なコンテンツ配信を確保します。

CDN 動作パスの一致

MediaTailor で設定した CDN プレフィックスが、CDN 動作で設定したパスパターンと一致していることを確認します。

リージョン CDN を検討する

リージョンごとに異なる CDN ドメインを使用している場合は、リージョンごとに個別の MediaTailor 設定を作成する必要があります。

ドメインの所有権を検証する

MediaTailor で設定する CDN ドメインを制御できることを確認します。

CDN マッピング設定の検証

MediaTailor で CDN マッピングを設定したら、設定が正しく機能していることを確認します。

1. CDN を介してマニフェストをリクエストします。
2. マニフェストコンテンツを調べて、セグメント URLs CDN ドメインを参照していることを確認します。
3. マニフェスト内のコンテンツセグメント URLs が CDN ドメインを指していることを確認します。
4. マニフェストの広告セグメント URLs が CDN ドメインを指していることを確認します。

包括的なテストと検証の手順については、「」を参照してください[CDN と MediaTailor の統合のテストと検証](#)。

次の手順

MediaTailor で CDN マッピングを設定した後の次のステップは、CDN 統合のセキュリティのベストプラクティスを実装することです。手順については、「[MediaTailor の CDN 統合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。

MediaTailor の CDN 統合セキュリティのベストプラクティス

このセクションでは、AWS Elemental MediaTailor コンテンツ配信ネットワーク (CDN) 統合のセキュリティのベストプラクティスを実装する方法について説明します。適切なセキュリティ設定は、不正アクセスや潜在的な脅威からコンテンツとインフラストラクチャを保護するのに役立ちます。

CDN 統合を保護することは、コンテンツを保護し、不正アクセスを防止し、セキュリティ要件を確実に順守するために不可欠です。これらのベストプラクティスを実装することで、ストリーミングワークフローの堅牢なセキュリティ体制を構築できます。

CDN セキュリティ設定

CDN ルーティングとキャッシュ設定を構成する前に、以下のセキュリティのベストプラクティスを実装して、コンテンツとインフラストラクチャを保護します。

トランスポートセキュリティ

MediaTailor はすべての通信に HTTPS のみを使用し、HTTP 接続を許可しません。ストリーミングワークフロー内のすべてのコンポーネント間の安全な通信を確保するには、次の手順を実行します。

1. CDN と MediaTailor エンドポイント間のすべての通信に HTTPS を設定します。MediaTailor は HTTPS を必要とし、HTTP 接続を受け入れません。
2. すべての HTTPS 接続に TLS 1.2 以降を使用します。
3. ビューワーからの HTTPS 専用接続を適用するように CDN を設定します。

アクセスコントロール

コンテンツとインフラストラクチャを不正アクセスから保護するには、次のアクセスコントロールを実装します。

1. 特定のリージョンへのコンテンツアクセスを制限する必要がある場合は、地理的制限を設定します。
2. ビューワー認証が必要なコンテンツには、署名付き URLs または署名付き Cookie を実装します。
3. CDN 設定への管理アクセス用に IP ベースの許可リストを設定します。

セキュリティモニタリング

セキュリティイベントを効果的に検出して対応するには、次のモニタリングプラクティスを実装します。

1. CDN ディストリビューションのアクセスログ記録を有効にします。
2. 異常なトラフィックパターンやアクセス試行のアラートを設定します。
3. セキュリティ設定を定期的に確認し、必要に応じて更新します。

高度なセキュリティ機能

次の高度なセキュリティ機能は、CDN 統合の保護を強化します。

ウェブアプリケーションファイアウォール (WAF)

CDN が WAF 機能をサポートしている場合は、一般的なウェブの脆弱性や攻撃から保護するように設定してください。

DDoS 保護

CDN が提供する DDoS 保護機能を有効にして、分散型denial-of-service攻撃を軽減します。

オリジンアクセスコントロール

直接アクセスの試行を防ぐために、CDN からのリクエストのみを受け入れるようにオリジンサーバーを設定します。

コンテンツの暗号化

機密性の高いコンテンツの場合は、トランスポートセキュリティを超えて追加の暗号化メカニズムを実装することを検討してください。

次の手順

セキュリティのベストプラクティスを実装した後の次のステップは、CDN 統合をテストしてトラブルシューティングすることです。包括的なテストとトラブルシューティングの手順[CDN 統合のトラブルシューティング](#)については、「」を参照してください。

パーソナライズされた動画広告用に CDN で SSAI をセットアップする

このセクションでは、AWS Elemental MediaTailor サーバー側の広告挿入 (SSAI) をコンテンツ配信ネットワーク (CDN) と統合するための包括的なガイダンスを提供します。SSAI CDN 統合をセットアップ、設定、最適化するには、次の手順に従います。

サーバー側の広告挿入 (SSAI) は、パーソナライズされた広告をクライアントレベルではなくサーバーレベルでビデオストリームにシームレスに挿入するテクノロジーです。CDN と組み合わせると、最小限のレイテンシーでパーソナライズされた広告を世界中の視聴者に配信するための堅牢でスケーラブルなソリューションが作成されます。

このトピックでは、マニフェストという用語を使用して、マルチバリエーションプレイリスト、メディアプレイリスト、MPDs。

必要なもの

CDN で MediaTailor 広告挿入を設定する前に、以下の必要なリソースを収集します。

AWS アカウントとアクセス許可

MediaTailor リソースを作成および管理するための適切なアクセス許可を持つ AWS アカウント

MediaTailor、CloudFront (使用している場合)、および関連サービスの IAM アクセス許可

アクセス許可の要件の詳細については、「」を参照してください [のセキュリティ AWS Elemental MediaTailor](#)。

必要なサービス

AWS Elemental MediaTailor 設定および実行

コンテンツ配信ネットワーク (CDN) アカウント (Amazon CloudFront またはサードパーティー CDN)

コンテンツのオリジンサーバー (HLS または DASH)

VAST または VMAP をサポートする広告決定サーバー (ADS)

コンテンツ要件

コンテンツが適切にエンコードされ、HLS または DASH 形式でパッケージ化されている

コンテンツ内の広告ブレイクマーカー (VOD 用) または SCTE-35 マーカー (ライブ用)

[開始する前に]

CDN で MediaTailor 広告挿入を実装する前に、以下のセットアップタスクを完了します。

1. CDN、MediaTailor、オリジンサーバー間のネットワーク接続を設定する

2. 安全なコンテンツ配信のために HTTPS を設定する
3. CDN ドメインの DNS 設定を構成する
4. MediaTailor の基本的なセットアップについては、[設定](#)「」および「」の手順を実行します[MediaTailor 広告挿入の使用開始](#)。

ナレッジの前提条件

このソリューションを正常に実装するには、以下が必要です。

- ストリーミングプロトコル (HLS/DASH) の理解
- CDN 設定原則の基本知識
- 広告挿入の概念に精通している

CDN 統合の利点

SSAI を CDN と統合すると、次の重要な利点があります。

ビューワーエクスペリエンスの向上

CDNs を適切に設定することで、広告移行中のバッファリング、起動時間、再生エラーを削減できます。これにより、視聴者のエンゲージメントと満足度が向上します。

コスト削減

効率的なキャッシュ戦略は、オリジンリクエストを最小限に抑えます。これにより、データ転送コストとオリジンサーバーの負荷が軽減され、特に大量の広告サポートコンテンツにとって重要です。

スケーラビリティ

最適化された CDN 設定は、パフォーマンスを低下させることなく、一般的なイベント中のトラフィックの急増を処理し、ピーク視聴時でもパーソナライズされた広告を確実に配信します。

グローバルリーチ

適切に設定された CDNs、場所に関係なく、世界中の視聴者に低レイテンシーでコンテンツを配信し、潜在的な視聴者を拡大します。

シームレスな広告移行

最適化された CDN 設定により、コンテンツと広告間のスムーズな移行が可能になり、ブロードキャスト品質の視聴エクスペリエンスが実現します。

以下のトピックでは、パフォーマンスを最適化するために CDN で MediaTailor を設定するための包括的なガイダンスを提供します。

トピック

- [CDN と MediaTailor の統合のための広告挿入アーキテクチャを理解する](#)
- [最適な広告配信のために CDN を使用して基本的な MediaTailor SSAI を設定する](#)
- [MediaTailor SSAI を収益化されたリニアチャンネルのチャンネルアセンブリと統合する](#)
- [CDN と MediaTailor の統合のパフォーマンスを最適化する](#)
- [CDN と MediaTailor の統合のオペレーションをモニタリングする](#)
- [中断のない広告配信のための CDN を使用した MediaTailor SSAI のトラブルシューティング](#)

CDN と MediaTailor の統合のための広告挿入アーキテクチャを理解する

このセクションでは、コンテンツ配信ネットワーク (CDNs) の概念とアーキテクチャについて説明します AWS Elemental MediaTailor。動的な広告挿入とマニフェスト操作が連携して効果的な動画収益化を実現する方法について説明します。

MediaTailor を使用したサーバー側の広告挿入 (SSAI) では、次のことが可能になります。

- 定義された広告ブレイクポイントでパーソナライズされた広告をビデオストリームに挿入する
- ビューワーデータに基づいて正確に広告をターゲットにする
- クライアント側の広告挿入テクノロジーが不要に

CDN と組み合わせると、パフォーマンスとスケーラビリティが向上し、ビデオ収益化戦略を強化して、これらのパーソナライズされたストリームを視聴者に配信できます。

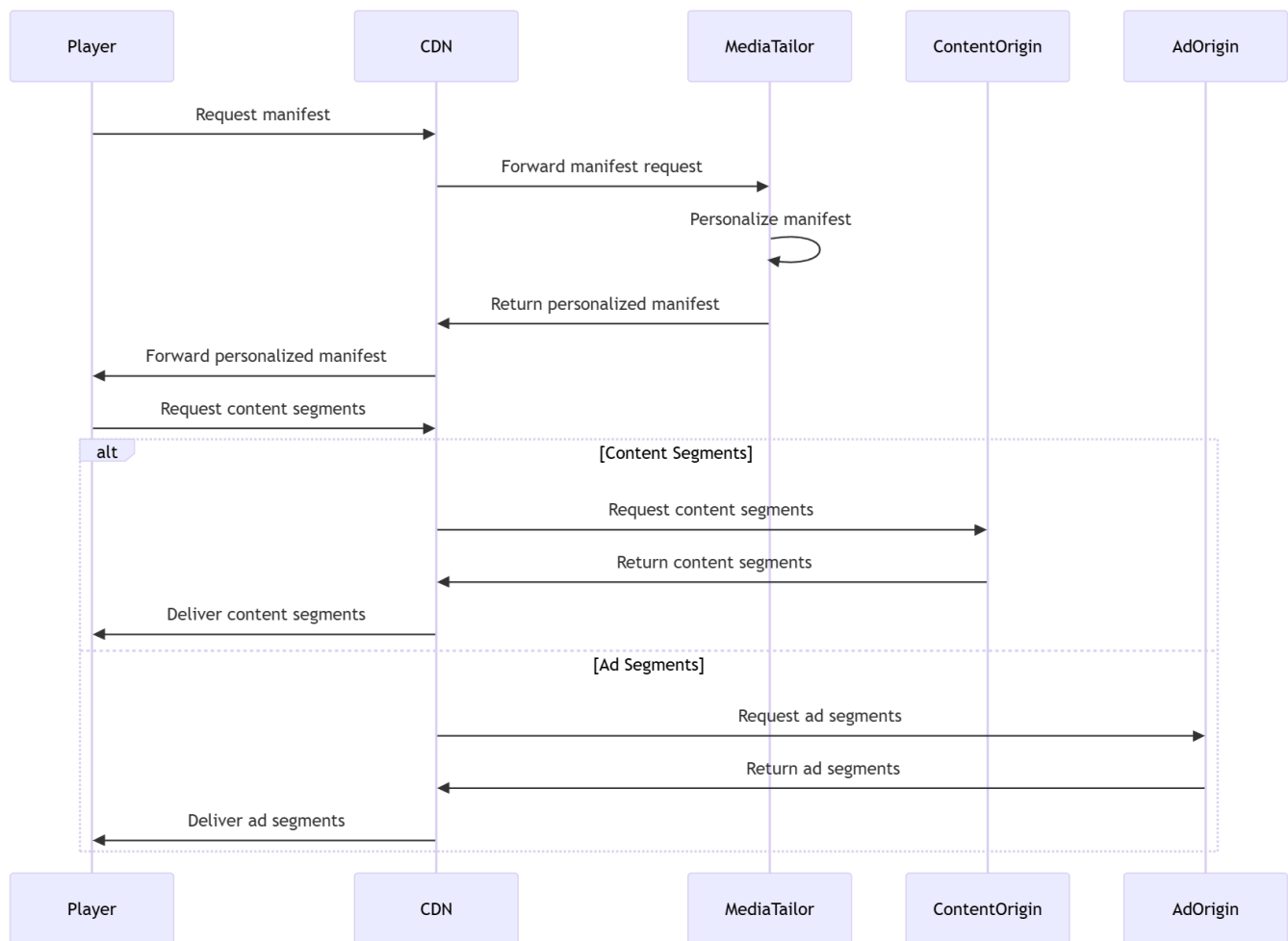
CDN を使用した広告挿入の推奨アーキテクチャは、ビューワーと広告挿入の間に CDN を配置し、広告挿入はオリジンから直接コンテンツにアクセスします。このアーキテクチャでは、コンテンツ配信とビデオ収益化の両方に次の利点があります。

- コンテンツと広告セグメントの効果的なキャッシュ
- MediaTailor でのリクエスト負荷の軽減
- ビューワーへの配信速度の向上
- URL 管理の簡素化
- デバイス間でパーソナライズされた広告を一貫して配信する

この推奨アーキテクチャでは、次のようになります。

1. CDN からのビューワーリクエストマニフェスト
2. CDN が広告挿入にリクエストを転送する
3. オリジンからの広告挿入リクエストコンテンツマニフェスト
4. 広告挿入が広告決定サーバー (ADS) から広告をリクエストする
5. 広告挿入は、広告マーカ (オリジンマニフェストから) を、特定のビューワー (ADS から) のターゲット広告セグメントを指す URLs に置き換えることで、マニフェストをパーソナライズします。
6. 広告挿入は、広告セグメント URLsを含むパーソナライズされたマニフェストを CDN に返し、それをビューワーに転送します。
7. ビューワーが CDN を介してセグメントをリクエストする
8. CDN は、セグメントタイプに基づいてセグメントリクエストをルーティングします。
 - コンテンツセグメントリクエストはコンテンツオリジンに送信されます
 - 広告セグメントリクエストは MediaTailor に移動します

このアーキテクチャは、CDN を使用するセキュリティと柔軟性の利点を維持しながら、最適なパフォーマンスを保証します。



Note

このフローは、VOD とライブコンテンツの間でわずかに異なります。VOD の場合、マニフェストはより長くキャッシュできますが、ライブコンテンツでは、ストリームの継続性を維持するためにより頻繁なマニフェスト更新が必要です。

VOD とライブコンテンツのキャッシュの主な違い:

VOD コンテンツ

マニフェストは頻繁に変更されないため、より長い TTL 値 (分から時間) を設定します。

ライブコンテンツ

マニフェストの TTL 値 (秒) を短く設定して、ビューワーが最新のストリームセグメントを受信できるようにします。

コンテンツオリジンとの間に CDN を配置することはお勧めしません AWS Elemental MediaTailor。これを行うと、いくつかの技術的な課題が発生する可能性があります。

キャッシュキーの衝突

クエリパラメータを適切に処理するように CDN を設定します。これにより、異なるクエリパラメータで同じマニフェストをリクエストするときに MediaTailor が誤ったマニフェストを受信できなくなります。

Gzip 圧縮の問題

マニフェスト解析エラーが発生した場合は、CDN が適切にフォーマットされたマニフェストを MediaTailor に配信していることを確認します。一部の CDN、破損した gzip ペイロードが配信され、解析に失敗する可能性があります。この場合、ワークフローの他の場所でコスト削減のために圧縮を維持しながら、CDN と MediaTailor 間の圧縮を無効にする必要がある場合があります。

マニフェストの鮮度

ライブストリームの場合は、現在のマニフェストを MediaTailor に配信するように CDN を設定します。これにより、コンテンツと広告間の同期の問題を防ぐことができます。

パフォーマンスの最適化

ネットワークホップと潜在的なキャッシュミスを最小限に抑えて、再生の起動時間を短縮します。

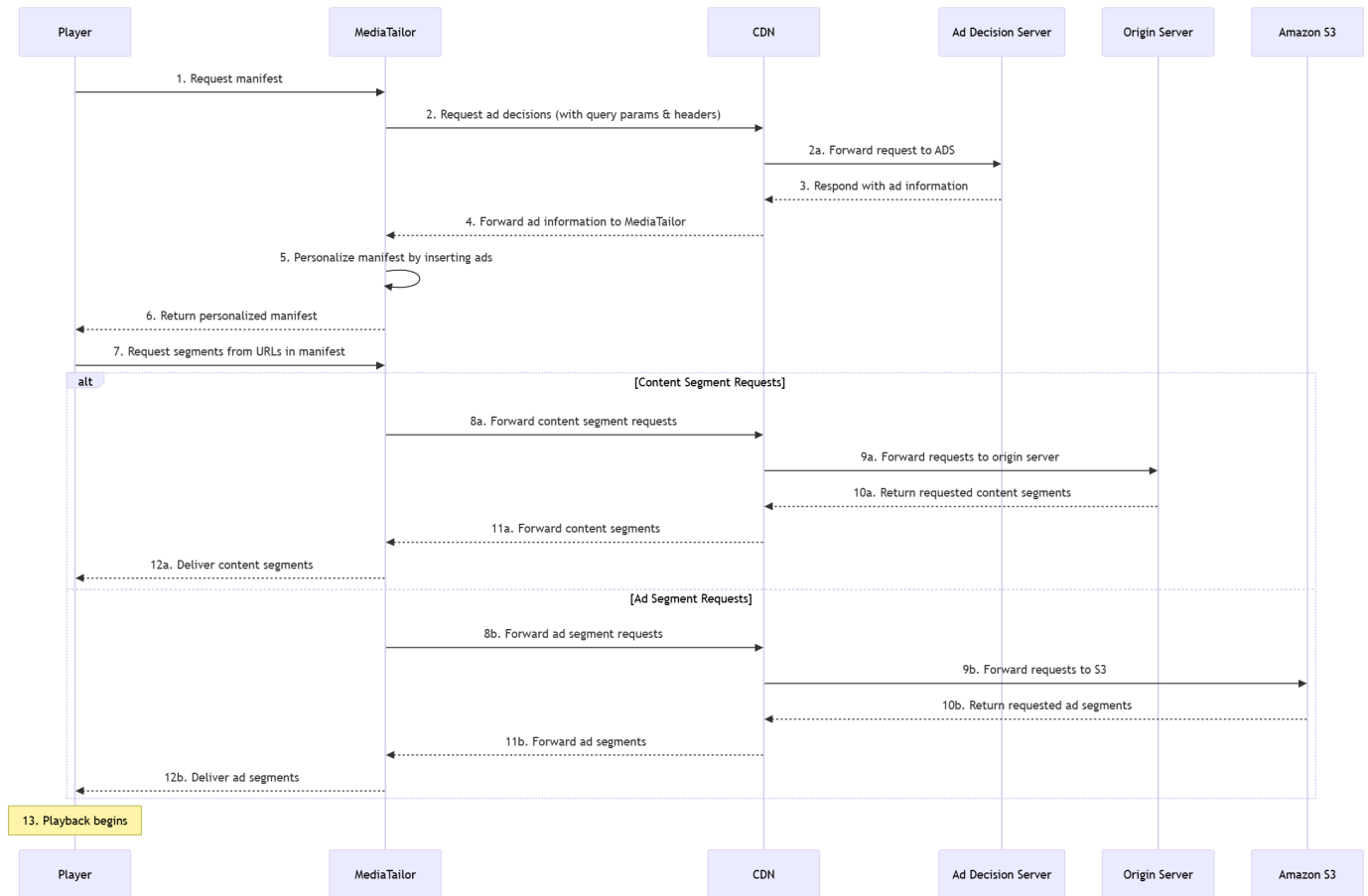
キャッシュ管理

特にマニフェストが頻繁に更新されるライブコンテンツに対して、簡素化されたキャッシュ無効化戦略を実装します。

この最適ではないアーキテクチャでは、次のようになります。

1. ビューワーは、多変量プレイリスト、メディアプレイリスト、または MPDs から直接リクエストします AWS Elemental MediaTailor。

2. MediaTailor は、CDN を介してコンテンツマニフェスト (多変量プレイリスト、メディアプレイリスト、または MPDs) をリクエストします。
3. CDN はリクエストをオリジンサーバーに転送します。
4. オリジンサーバーは、CDN に多変量プレイリスト、メディアプレイリスト、または MPDsを返します。
5. CDN は、多変量プレイリスト、メディアプレイリスト、または MPDsを MediaTailor に転送します。
6. MediaTailor は、広告決定サーバー (ADS) から広告をリクエストします。
7. MediaTailor は、マルチバリエーションプレイリスト、メディアプレイリスト、または MPDs に広告を挿入してマニフェストをパーソナライズし、ビューワーに直接配信します。
8. このアーキテクチャでは、追加のレイテンシー、潜在的なキャッシュの問題が発生し、トラブルシューティングが複雑になります。



リクエストとレスポンスのフロー

CDN で動的広告挿入を実装する場合は、このリクエストとレスポンスのフローをサポートするようにシステムを設定します。

- MediaTailor をマニフェストオリジンとして CDN からマルチバリエーションプレイリスト (HLS) または MPDs (DASH) をリクエストするようにプレイヤーを設定します。
- すべてのクエリパラメータとヘッダーを含む、すべての多変量プレイリスト、メディアプレイリスト、MPD リクエストを MediaTailor に転送するように CDN を設定します。
- MediaTailor が広告決定サーバー (ADS) と通信し、クエリパラメータとヘッダーを渡せることを確認します。
- クエリパラメータを使用して挿入する広告を決定するように ADS を設定します。

- MediaTailor 再生設定で CDN プレフィックスを設定して、MediaTailor が CDN ドメイン名をコンテンツと広告セグメント URL プレフィックスに置き換えられるようにします。
- MediaTailor からリクエスト元のプレイヤーにパーソナライズされたマルチバリエーションプレイリスト、メディアプレイリスト、MPDs を転送するように CDN を設定します。
- セグメント URLs を翻訳するように CDN を設定し、コンテンツセグメントリクエストをオリジンサーバーに転送し、広告リクエストを MediaTailor がトランスコードされた広告を保存する Amazon S3 バケットに転送します。

広告挿入の CDN 用語

これらの重要な用語を理解することは、広告挿入 CDN 統合の実装とトラブルシューティングに役立ちます。

オリジン CDN とエッジ CDN

オリジン CDN: MediaTailor とコンテンツオリジンの間に配置される CDN。コンテンツセグメントをキャッシュして、オリジンサーバーの負荷を軽減します。マルチ CDN アーキテクチャでは、これはオリジンと直接インターフェイスする最初の CDN レイヤーです。

Edge CDN: ビューワーと MediaTailor の間に配置される CDN。パーソナライズされたマニフェストとコンテンツをビューワーに配信します。マルチ CDN アーキテクチャでは、これはビューワーと直接インターフェイスする最も外側の CDN レイヤーです。

CDN 設定条件

キャッシュ動作: CDN がさまざまなタイプのリクエストを処理する方法を決定するルール。これらのルールには以下が含まれます。

- キャッシュ期間の設定
- オリジンルーティング設定
- リクエスト処理パラメータ

TTL (有効期限): オリジンから更新する必要があるコンテンツが CDN キャッシュで有効である期間。

キャッシュキー: CDN がキャッシュされたコンテンツを保存および取得するために使用する一意の識別子。通常、以下が含まれます。

- URL パス

- クエリパラメータ
- 選択したヘッダー

オリジンシールド: CDN エッジロケーションとオリジンサーバー間の中間キャッシュレイヤー。これにより、オリジンへのリクエストの数が減ります。

リクエストの折りたたみ: 同じコンテンツに対する複数の同時リクエストを 1 つのオリジンリクエストに結合する CDN 機能。

MediaTailor 固有の CDN 用語

CDN コンテンツセグメントプレフィックス: マニフェストでコンテンツセグメントURLs を生成するときに が AWS Elemental MediaTailor 使用する CDN ドメイン名。

CDN 広告セグメントプレフィックス: MediaTailor がマニフェストで広告セグメントURLs を生成するときに使用する CDN ドメイン名。

MediaTailor での CDN 設定の詳細については、「」を参照してください[CDN 統合をセットアップする](#)。

Note

これらの用語は、チャネルアセンブリドキュメントで使用されている用語と一致しています。チャネルアセンブリの用語については、「」を参照してください[チャネルアセンブリの CDN 用語](#)。

最適な広告配信のために CDN を使用して基本的な MediaTailor SSAI を設定する

このセクションでは、コンテンツ配信ネットワーク (CDN) を使用して AWS Elemental MediaTailor 動的広告挿入を設定して動画収益化ワークフローを最適化するstep-by-step。

動的変数を使用した高度な広告サーバー設定については、「」を参照してください[ADS リクエストの MediaTailor 動的広告変数](#)。広告ターゲティングのために CDNs 「」を参照してください[MediaTailor マニフェストクエリパラメータ](#)。

CDNs 「」を参照してください[CDN と MediaTailor の統合のための広告挿入アーキテクチャを理解する](#)。

前提条件

CDN で広告挿入を設定する前に、以下を確認してください。

- アクティブな AWS Elemental MediaTailor 設定
- 適切な広告マーカースを使用して HLS または DASH コンテンツを配信するコンテンツオリジンサーバー

広告マーカースの詳細については、「」を参照してください[広告挿入動作について](#)。

- 広告ターゲティングの VAST または VMAP をサポートする広告決定サーバー (ADS)
- CDN アカウント (Amazon CloudFront や別の CDN プロバイダーなど)
- マニフェスト操作と動的広告挿入の概念に関する基本的な知識

ステップ 1: 最適な広告配信のために CDN キャッシュを設定する

動画収益化ワークフローのパフォーマンスを最適化するには、適切な CDN キャッシュ設定が不可欠です。キャッシュ要件は、サーバー側の広告挿入 (SSAI) とサーバーガイド付き広告挿入 (SGAI) で異なります。以下の推奨設定を使用して、コンテンツとパーソナライズされた広告の両方を効率的に配信します。

SSAI CDN キャッシュ設定

サーバー側の広告挿入ワークフローでは、最適なパフォーマンスを得るには、適切なキャッシュ設定が不可欠です。SSAI では、セグメントが効率的にキャッシュされている間にパーソナライズされたマニフェストがキャッシュされないように、特定の TTL 値とキャッシュキー設定が必要です。

TTL 値、パスパターン、キャッシュキー設定を含む詳細な SSAI キャッシュ設定については、CDN 最適化ガイド[サーバー側の広告挿入 \(SSAI\) キャッシュ](#)の「」を参照してください。

SSAI の主なキャッシュ原則:

- マニフェスト: TTL を 0 秒に設定して、パーソナライズされたコンテンツのキャッシュを防止します
- セグメント: オリジンの負荷を軽減するために積極的にキャッシュする (24 時間以上)
- キャッシュキー: マニフェストのすべてのクエリパラメータ、セグメントの URL パスのみを含める

SGAI CDN キャッシュ設定

サーバーガイド広告挿入ワークフローの場合、SGAI マニフェストはパーソナライズされた広告エクスペリエンスを提供しながら短期間キャッシュできるため、キャッシュ要件は SSAI とは異なります。

VOD やライブ TTL 値を含む包括的な SGAi キャッシュ設定については、最適化ガイドのキャッシュテーブルを参照してください。SGAI を使用すると、広告パーソナライゼーション機能を維持しながら、SSAI よりもキャッシュ効率が向上します。

SGAI キャッシュの主な違い:

- マニフェスト: 短期間キャッシュ可能 (VOD の場合は 5~30 分、ライブの場合は 2~10 秒)
- セグメント: SSAI のように積極的にキャッシュする (ほとんどのコンテンツでは 24 時間以上)
- パフォーマンス上の利点: キャッシュ可能なマニフェストにより、SSAI よりもキャッシュヒット率が向上します

Amazon CloudFront では、TTL 値とキャッシュキーポリシーが異なるキャッシュ動作を使用して、これらの設定を実装できます。その他の CDN については、同様のキャッシュルールの実装に関する特定のドキュメントを参照してください。

ステップ 2: ハイブリッドアプローチを実装する (必要な場合)

アーキテクチャで、コンテンツオリジンと MediaTailor の間に別の CDN またはキャッシュレイヤーを持つハイブリッドアプローチが必要な場合:

1. CDN 設定で懸念を明確に分離します。
2. 前のセクションで説明した技術的な問題を防ぐため、特定の CDN 設定を構成します。
3. 設定を徹底的にテストして、マニフェストのパーソナライゼーションが正しく機能することを確認します。
4. パフォーマンスメトリクスをモニタリングして、多変量プレイリスト、メディアプレイリスト、MPDs、セグメントの最適な配信を確保します。

ハイブリッドアプローチを実装するときは、以下の特定の設定を考慮してください。

- コンテンツオリジンと MediaTailor 間の CDN の場合:
 - マニフェストファイルの圧縮パススルーを設定して、オリジンからの元の圧縮状態を保持する

- キャッシュキーにすべてのクエリパラメータを含める
- ライブコンテンツマニフェストの短い TTL 値を設定する
- MediaTailor とビューワー間の CDN の場合:
 - 広告セグメントのキャッシュ時間を長く設定する
 - パーソナライズされたマニフェストに適切な TTLs を設定する
 - コンテンツセグメントと広告セグメントに適切なオリジンルーティングを実装する

ステップ 3: CDN セットアップを完了する

アーキテクチャを選択し、リクエストフローを理解したら、「」の詳細な設定手順に従ってセットアップを完了します [CDN 統合をセットアップする](#)。

特定の CDN プロバイダーについては、以下の追加リソースを参照してください。

- Amazon CloudFront: CloudFront 固有の設定手順 [CloudFront 統合](#) については、「」を参照してください。
- その他の CDNs: このガイドで概説されている一般的な原則を適用し、特定の CDN の設定オプションに適応させます。

ステップ 4: 設定を確認する

CDN のセットアップが完了したら、動的広告挿入ワークフローが正しく機能していることを確認します。

1. サンプルプレイヤーを使用して CDN 経由で再生をテストする
2. パーソナライズされた広告が指定された広告ブレイクポイントに正しく挿入されていることを確認します。
3. CDN ログをチェックして、適切なリクエストルーティングを確認する
4. キャッシュヒット率をモニタリングして、コンテンツセグメントと広告セグメントの両方に最適なパフォーマンスを確保する
5. 広告ターゲティングパラメータがワークフローに適切に渡されていることを確認します。

包括的なテストと検証の手順については、「」を参照してください [CDN と MediaTailor の統合のテストと検証](#)。SSAI 実装のモニタリングの詳細については、「」を参照してください [CDN と](#)

[MediaTailor の統合のオペレーションをモニタリングする](#)。パフォーマンスを最適化するには、「」を参照してください[CDN と MediaTailor の統合のパフォーマンスを最適化する](#)。

MediaTailor SSAI を収益化されたリニアチャンネルのチャンネルアセンブリと統合する

このトピックでは、AWS Elemental MediaTailor サーバー側の広告挿入をチャンネルアセンブリおよびコンテンツ配信ネットワーク (CDN) 統合と組み合わせる方法について説明します。この統合により、次のことが可能になります。

- パーソナライズされた広告を使用して収益化された線形チャンネルを作成する
- 同じコンテンツを見ているさまざまな視聴者にターゲットを絞った広告を配信する
- ブロードキャスト品質の視聴エクスペリエンスを維持する

SSAI とチャンネルアセンブリを組み合わせるメリット

SSAI とチャンネルアセンブリを統合すると、いくつかの重要な利点があります。

線形チャンネルの収益化

パーソナライズされた広告をリニアチャンネルに挿入して、コンテンツライブラリから収益を生成します。ライブコンテンツと VOD コンテンツの両方を単一の線形ストリーム内で収益化できます。

パーソナライズされた広告

同じチャンネルコンテンツを見ているさまざまな視聴者にさまざまな広告を配信します。このターゲットを絞ったアプローチでは、従来のブロードキャスト広告と比較して、広告の関連性と潜在的な収益が増加します。

広告時間枠管理の簡素化

SCTE-35 マーカーでコンテンツを条件付けすることなく、チャンネルアセンブリプログラムで広告ブレイクポイントを定義します。これにより、コンテンツ内の自然なブレイクポイントに広告を簡単に挿入できます。

ブロードキャスト品質のエクスペリエンス

MediaTailor は、コンテンツと広告のシームレスな移行により、高品質の視聴エクスペリエンスを維持します。サーバー側の広告挿入は、次のような多くの一般的な問題を排除します。

- 広告遷移中のバッファリング
- 収益化を妨げる広告ブロッカー
- 再生品質の不整合

スケーラブルな配信

CDN と組み合わせると、この統合はパフォーマンスやパーソナライゼーション機能を低下させることなく、数百万の同時視聴者にスケールできます。

アーキテクチャの概要

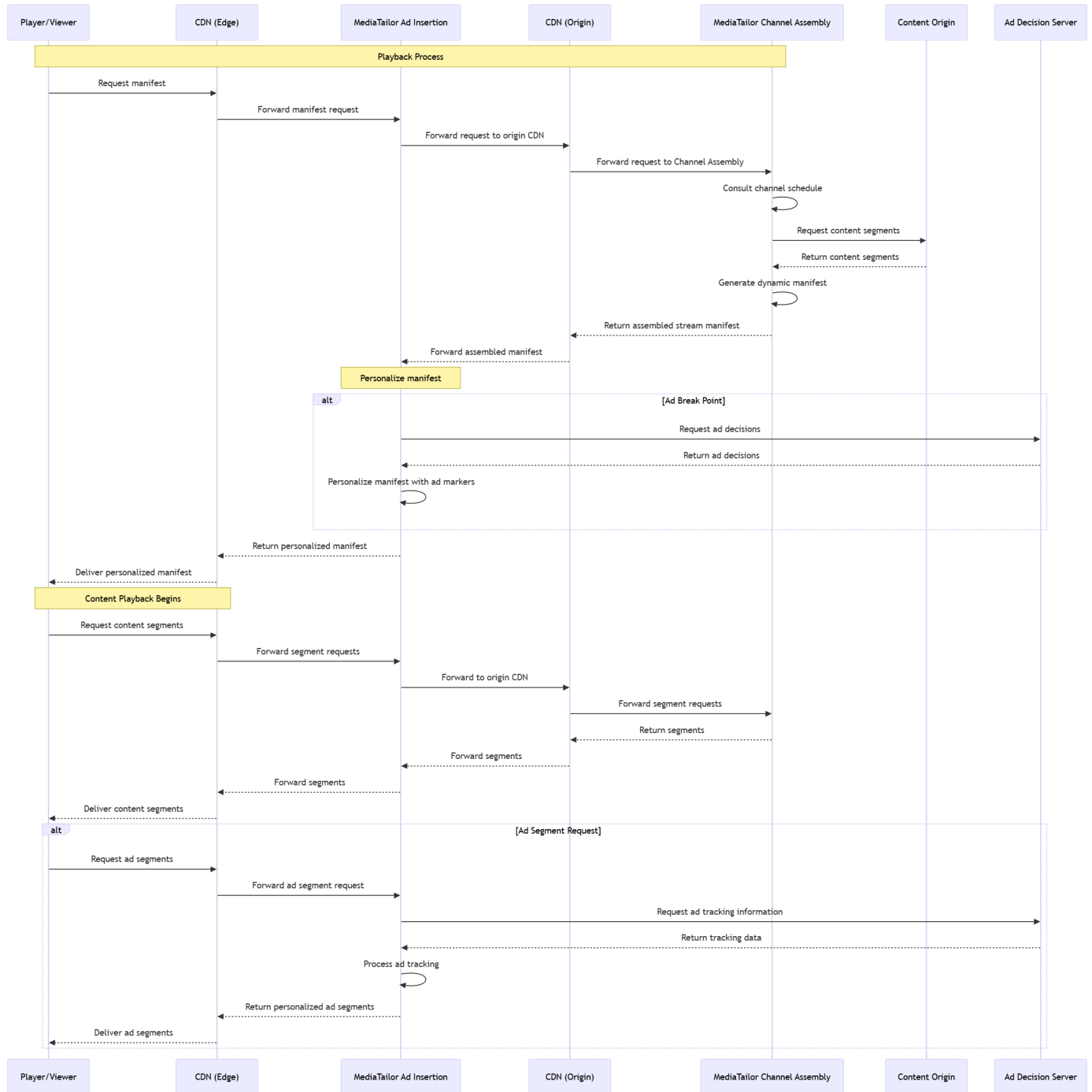
SSAI とチャネルアセンブリを組み合わせるアーキテクチャには通常、次のコンポーネントが含まれます。

- チャネルアセンブリ: VOD とライブコンテンツからリニアチャネルを作成し、生成されたマニフェストに広告マーカーを作成するスレートコンテンツを挿入します
- 広告挿入: 広告ブレイクポイントを認識し、マニフェスト内のパーソナライズされた広告セグメントを指す URLs を挿入します
- 広告決定サーバー (ADS): 各ビューワーに挿入する広告を決定します
- コンテンツ配信ネットワーク (CDN): アセンブルされたコンテンツと広告セグメントをビューワーに配信します
- オリジンサーバー: VOD とライブコンテンツセグメントを保存します

このアーキテクチャの詳細は以下のとおりです。

1. チャネルアセンブリは、VOD とライブコンテンツからリニアチャネルを作成し、生成されたマニフェストに広告マーカーを作成するスレートコンテンツを挿入します。
2. ビューワーがチャネルをリクエストすると、広告挿入はリニアチャネルに挿入された広告ブレイクを認識します。
3. 広告挿入は、ADS を呼び出して広告のリストを受信し、トランスコードして、トランスコードされた広告セグメントを指す URLs をパーソナライズされたマニフェストに挿入します。
4. CDN はパーソナライズされたストリームをビューワーに配信します。

次の図は、このワークフローを示しています。



統合のセットアップ

チャネルアセンブリで SSAI を設定するには:

1. ビューワーからのマニフェストリクエストを受け入れ、AWS Elemental MediaTailor 広告挿入に転送するようにエッジ CDN を設定します。
2. MediaTailor 広告挿入を設定して、リクエストをオリジン CDN に転送します。
3. MediaTailor チャンネルアセンブリにリクエストを転送するようにオリジン CDN を設定します。
4. MediaTailor チャンネルアセンブリを設定して、現在のスケジュールに基づいて動的マニフェストを生成します。
5. アセンブルされたマニフェストを MediaTailor 広告挿入に転送するようにオリジン CDN を設定します。
6. MediaTailor 広告挿入を設定して、広告ブレイクポイントで広告決定サーバーに広告決定をリクエストします。
7. 広告マーカー (チャンネルアセンブリから) をターゲット広告セグメント (ADS から) を指す URLs に置き換えて、マニフェストをパーソナライズするように MediaTailor 広告挿入を設定します。
8. パーソナライズされたマニフェストをビューワーに配信するようにエッジ CDN を設定します。
9. コンテンツと広告セグメントの両方のリクエストを効率的に処理するように CDN アーキテクチャを設定します。

チャンネルアセンブリでの広告ブレイクの定義

チャンネルアセンブリでプログラムを作成するときは、いくつかの方法で広告ブレイクを定義できます。

プログラム移行

チャンネルスケジュールのプログラム間に広告を挿入します。これは最も簡単なアプローチであり、広告がプログラムのコンテンツを中断しないようにします。

SCTE-35 マーカー

VOD コンテンツに SCTE-35 マーカーが含まれている場合、チャンネルアセンブリはこれらのマーカーを保持し、広告挿入はそれらを広告ブレイクポイントとして使用できます。

時間ベースの挿入

プログラム内の特定の時点で広告時間枠を定義します。これにより、コンテンツ内の自然なブレイクポイントに広告を挿入できます。

広告時間枠でプログラムを作成する方法の詳細については、[「プログラムの使用」](#)を参照してください。

CDN キャッシュに関する考慮事項

チャンネルアセンブリと SSAI を CDN と組み合わせるときに最適なパフォーマンスを得るには:

- チャンネルアセンブリと SSAI リクエストを区別するキャッシュ動作を設定する
- で推奨されているマニフェストとセグメントに適切な TTL 値を設定する [ステップ 1: 最適な広告配信のために CDN キャッシュを設定する](#)
- チャンネルアセンブリ、広告挿入、CDN オリジン間の適切なルーティングを確保する
- チャンネルアセンブリと広告挿入コンポーネントの両方のパフォーマンスメトリクスをモニタリングする

複合実装に推奨されるキャッシュ設定

コンテンツタイプ	TTL	キャッシュキー要素
チャンネルアセンブリマニフェスト	0 秒	URL パス + クエリパラメータ
SSAI パーソナライズされたマニフェスト	0 秒	URL パス + すべてのクエリパラメータ
コンテンツセグメント	24 時間以上	URL パスのみ
広告セグメント	24 時間以上	URL パスのみ

統合ソリューションのモニタリング

統合ソリューションが最適に動作していることを確認するには、次の主要なメトリクスをモニタリングします。

チャンネルアセンブリメトリクス

マニフェスト生成時間、プログラム移行、チャンネルアセンブリプロセスのエラーをモニタリングします。

広告挿入メトリクス

広告フィルレート、広告決定サーバーの応答時間、広告挿入エラーを追跡します。

CDN メトリクス

コンテンツセグメントと広告セグメントの両方で、キャッシュヒット率、オリジンリクエストボリューム、レスポンスレイテンシーをモニタリングします。

ビューワーエクスペリエンスメトリクス

特に広告の移行中に、再バッファリングイベント、起動時間、視聴者のエンゲージメントを追跡します。

モニタリングの詳細については、[CDN と MediaTailor の統合のオペレーションをモニタリングする](#)「」および「」を参照してください[MediaTailor チャンネルアセンブリ CDN オペレーションのモニタリング](#)。

一般的な問題のトラブルシューティング

統合ソリューションに関する問題をトラブルシューティングするときは、以下の一般的な問題を考慮してください。

広告ブレイク同期の問題

広告が予想されるブレイクポイントに表示されない場合は、チャンネルアセンブリプログラムの広告ブレイク定義が正しく設定されており、広告挿入によってこれらのブレイクポイントが適切に識別されていることを確認します。

マニフェスト配信エラー

視聴者に再生の問題が発生した場合は、CDN がチャンネルアセンブリと広告挿入の間でマニフェストリクエストを正しく転送していること、およびキャッシュ設定がこれらのマニフェストの動的な性質に適していることを確認します。

セグメントルーティングの問題

コンテンツまたは広告セグメントがロードされていない場合は、CDN がセグメントリクエストを適切なオリジンに正しくルーティングしていること、およびマニフェスト内のセグメント URLs が正しくフォーマットされていることを確認します。

パフォーマンスの低下

ビューワーにバッファリングまたは高レイテンシーが発生している場合は、CDN キャッシュヒット率とオリジンリクエストボリュームをチェックして、配信パイプラインの潜在的なボトルネックを特定します。

トラブルシューティングの詳しいガイダンスについては、「[中断のない広告配信のための CDN を使用した MediaTailor SSAI のトラブルシューティング](#)」を参照してください。

ベストプラクティス

SSAI とチャンネルアセンブリの統合を成功させるには、次のベストプラクティスに従います。

- 徹底的にテストする: 本稼働環境にデプロイする前に、さまざまなコンテンツタイプ、広告シナリオ、視聴者条件で統合ソリューションをテストします。
- 継続的なモニタリング: 包括的なモニタリングとアラートを設定して、発生した問題をすばやく特定して対処します。
- キャッシュの最適化: 実際の使用パターンとパフォーマンスメトリクスに基づいて、CDN キャッシュ設定を定期的に確認および調整します。
- スケールを計画する: 特に一般的なチャンネルやイベントでピークトラフィック負荷を処理するようにアーキテクチャを設計します。
- 冗長性を検討する: 重要なコンポーネントに冗長性を実装して、リニアチャンネルの高可用性を確保します。
- 広告遷移の最適化: 一貫したエンコーディングプロファイルとセグメント期間を使用して、コンテンツと広告間のスムーズな遷移を確保します。

関連情報

SSAI とチャンネルアセンブリの統合の詳細については、以下を参照してください。

チャンネルアセンブリドキュメント

[AWS Elemental MediaTailor を使用して線形アセンブルストリームを作成する](#) - チャンネルアセンブリの概念について説明します。

[CDN を使用したチャンネルアセンブリ](#) - CDN を使用してチャンネルアセンブリを設定する

SSAI ドキュメント

[CDN を使用した広告挿入](#) - CDN を使用して広告挿入を設定する

[CDN と MediaTailor の統合のための広告挿入アーキテクチャを理解する](#) - 広告挿入 CDN アーキテクチャを理解する

CDN 設定

[CDN 統合をセットアップする](#) - 一般的な CDN 設定ガイダンス

[CloudFront 統合](#) - CloudFront 固有の設定

CDN と MediaTailor の統合のパフォーマンスを最適化する

コンテンツ配信ネットワーク (CDN) 設定を最適化することで、AWS Elemental MediaTailor 広告挿入実装のパフォーマンスを最大化します。これらの設定により、効率的なコンテンツ配信と最適な視聴体験が保証されます。

SSAI 実装に固有のキャッシュとルーティング最適化の詳細なガイダンスについては、「」を参照してください[CDN パフォーマンスの最適化](#)。統合最適化ガイドには、すべての MediaTailor CDN 統合に適用される包括的なキャッシュ設定、リクエストルーティング設定、パフォーマンスベンチマークが記載されています。

一般的なパフォーマンスの課題

CDNs を使用した SSAI 実装は、いくつかのパフォーマンスの課題に直面する可能性があります。

マニフェスト操作のオーバーヘッド

MediaTailor はリアルタイムのマニフェスト操作を実行するため、適切に最適化しないとレイテンシーが発生する可能性があります。以下ではレイテンシーが発生する可能性があります。

- 広告決定サーバー (ADS) リクエストの処理時間
- 広告セグメント参照を使用してマニフェストを変更するのに必要な時間
- パーソナライゼーションの追加処理

キャッシュ効率の問題

パーソナライズされたマニフェストは、次の理由で CDN キャッシュ効率を低下させる可能性があります。

- 各ビューワーは一意的なマニフェストを受け取る場合があります
- セッションパラメータはキャッシュをフラグメント化できます
- 動的コンテンツには慎重なキャッシュ設定が必要です

オリジン負荷スパイク

キャッシュが正しくないと、次の場合にオリジン負荷が急増する可能性があります。

- トラフィックの多いイベント
- キャッシュの更新

- CDN 設定の変更

広告関連の再生の問題

広告挿入は、次のような再生中断を引き起こす可能性があります。

- 広告遷移中のバッファリング
- コンテンツと広告の品質の違い
- 広告を取得できない場合の再生失敗

キャッシュ戦略、リクエストルーティング、パフォーマンスベンチマーク、高度な最適化手法を含む包括的なパフォーマンス最適化ガイダンスについては、「」を参照してください[CDN パフォーマンスの最適化](#)。統合最適化ガイドには、SSAI 実装を含むすべての MediaTailor CDN 統合に適用される詳細な設定とベンチマークが記載されています。

ベストプラクティスの概要

CDNs で最適な SSAI パフォーマンスを確保するには、次のベストプラクティスに従います。

アーキテクチャのベストプラクティス

- スケールと要件に適したアーキテクチャパターンを選択する
- レイテンシーを最小限に抑えるために、サービスを近接してデプロイする
- 重要なコンポーネントの冗長性を実装する

キャッシュのベストプラクティス

- コンテンツタイプごとに異なるキャッシュ戦略を使用する
- キャッシュキーを最適化してパーソナライゼーションと効率のバランスを取る
- コンテンツタイプと更新頻度に基づいて適切な TTLsを設定します。TTL の詳細な推奨事項については、「」を参照してください[CDN と MediaTailor の統合のキャッシュ最適化](#)。

広告配信のベストプラクティス

- タイムアウトとフォールバックによる ADS インタラクションの最適化
- コンテンツ仕様に合わせて広告を準備する
- 効率的な広告セグメント配信を実装する

モニタリングのベストプラクティス

- SSAI 実装のすべてのコンポーネントをモニタリングする
- パフォーマンス低下のアラートを設定する

- 設定を定期的に確認して最適化する

最適化ガイダンスを完了する

詳細なキャッシュ戦略、リクエストルーティング設定、パフォーマンスベンチマーク、高度な最適化手法を含む包括的な CDN 最適化ガイダンスについては、「」を参照してください[CDN パフォーマンスの最適化](#)。統合最適化ガイドには、SSAI 実装を含むすべての MediaTailor CDN 統合に適用される完全な設定とベンチマークが記載されています。

CDN と MediaTailor の統合のオペレーションをモニタリングする

AWS Elemental MediaTailor は、コンテンツ配信ネットワーク (CDN) メトリクスと組み合わせることで、SSAI 実装に関する包括的なインサイトを提供する堅牢な分析機能を提供します。このトピックでは、次の内容について説明します。

必須メトリクス、モニタリングツールのセットアップ、アラート設定、すべての MediaTailor 実装に適用されるトラブルシューティング戦略を含む包括的な CDN モニタリングガイダンスについては、「」を参照してください[CDN モニタリング](#)。このトピックでは、SSAI 固有のモニタリング要件と広告挿入分析に焦点を当てます。

- SSAI と CDN 統合のモニタリング戦略
- 分析ツールとデータ収集方法
- データ駆動型の最適化手法

CDNs を使用した SSAI の主要なメトリクス

CDNs を使用して SSAI 実装を効果的にモニタリングするには、以下の重要なメトリクスを追跡します。

広告挿入メトリクス

広告フィルレート: 広告で正常に埋められた広告機会の割合。

広告エラー率: エラーが発生した広告リクエストの割合。

広告応答時間: 広告決定サーバーが広告リクエストに応答するまでにかかる時間。

広告期間の精度: 挿入された広告の実際の期間が、予想される期間とどの程度一致しているか。

ビューワーエクスペリエンスメトリクス

再バッファリング率: バッファリングに費やされた表示時間の割合。

起動時間: 動画再生の開始にかかる時間。

広告遷移の滑らかさ: プレイヤーがコンテンツと広告の間でどの程度シームレスに遷移するか。

セッション期間: ストリームを中止する前に視聴する時間。

分析ツールと統合

これらのツールを組み合わせ、SSAI 実装用の包括的な分析ソリューションを作成します。

AWS Elemental MediaTailor サーバー側のメトリクス

MediaTailor は、広告リクエスト、レスポンス、エラーを追跡する Amazon CloudWatch を通じて組み込みメトリクスを提供します。これらのメトリクスは、CloudWatch コンソールで表示することも、カスタムダッシュボードに統合することもできます。

MediaTailor の主要なメトリクスは次のとおりです。

- `AdDecisionServer.Ads`: 広告決定サーバーによって返された広告の数。
- `AdDecisionServer.Duration`: 広告決定サーバーによって返される広告の合計期間。
- `AdDecisionServer.Errors`: 広告決定サーバーによって返されたエラーの数。
- `AdDecisionServer.Latency`: 広告決定サーバーの応答時間。

MediaTailor メトリクスの完全なリストについては、[Amazon CloudWatch による MediaTailor のモニタリング](#)」を参照してください。

CDN 分析

CDN プロバイダーは、コンテンツ配信パフォーマンスの詳細な分析を提供します。Amazon CloudFront の場合、CloudWatch メトリクスと Amazon CloudFront アクセスログを使用して配信パターンを分析します。

モニタリングすべき重要な CDN メトリクス:

- コンテンツタイプ別のリクエスト数 (マニフェストとセグメント)
- さまざまなコンテンツタイプのキャッシュヒット率

- ビューワーの地理的分布
- エラーコード別のエラー率

クライアント側の追跡

クライアント側の追跡を実装して、サーバー側で利用できないビューワーエクスペリエンスメトリクスを収集します。

- プレイヤーイベント (再生、一時停止、シーク、バッファ)
- 広告ビューの完了率
- サービス品質メトリクス (解決の変更、ビットレート)
- 視聴者のエンゲージメントパターン

MediaTailor クライアント側の追跡を使用して、これらのメトリクスを収集および報告することを検討してください。

統合されたダッシュボード

複数のソースからのメトリクスを組み合わせた包括的なダッシュボードを作成します。

- CloudWatch ダッシュボードを使用して MediaTailor メトリクスと CloudFront メトリクスを組み合わせる
- より高度な視覚化のためにサードパーティーの分析プラットフォームを検討する
- クロスサービス相関を設定してメトリクス間の関係を特定する

モニタリング戦略の実装

CDN 実装で SSAI の包括的なモニタリング戦略を実装するには、次の手順に従います。

1. 基本モニタリングを設定する

- MediaTailor の CloudWatch メトリクスを有効にする
- CDN ログ記録とメトリクス収集を設定する
- ビデオプレーヤーにクライアント側の追跡を実装する

2. カスタムダッシュボードを作成する

- 主要なメトリクスを組み合わせた CloudWatch ダッシュボードを構築する
- 広告フィルレート、CDN パフォーマンス、ビューワーエクスペリエンスの視覚化を含める
- 重要なイベント (設定変更、メジャーブロードキャスト) の注釈を追加する

3. アラートを設定する

- 重要なメトリクスの CloudWatch アラームを設定する
- 複数の関連条件でトリガーする複合アラームを作成する
- 通知チャンネル (E メール、SMS、Amazon SNS) を設定する

4. 自動レスポンスを実装する

- CloudWatch Events を使用して一般的な問題への自動応答をトリガーする
- 必要に応じて手動介入用のランブックを作成する
- さまざまなアラートシナリオのトラブルシューティング手順を文書化する

Example 包括的な SSAI モニタリングダッシュボードの作成

この例では、MediaTailor メトリクスと CloudFront メトリクスを組み合わせた CloudWatch ダッシュボードを作成する方法を示します。CloudFront

```
{
  "widgets": [
    {
      "type": "metric",
      "properties": {
        "metrics": [
          [ "AWS/MediaTailor", "AdDecisionServer.Ads", "Configuration", "your-config-
name" ],
          [ ".", "AdDecisionServer.Errors", ".", "." ]
        ],
        "period": 300,
        "stat": "Sum",
        "region": "us-west-2",
        "title": "Ad Decision Server Performance"
      }
    },
    {
      "type": "metric",
      "properties": {
        "metrics": [
          [ "AWS/CloudFront", "Requests", "DistributionId", "your-distribution-id" ],
          [ ".", "4xxErrorRate", ".", "." ],
          [ ".", "5xxErrorRate", ".", "." ]
        ],
        "period": 300,
        "stat": "Average",
        "region": "us-east-1",
```

```
        "title": "CDN Performance"
    }
}
]
```

データ駆動型の最適化

収集した分析データを使用して SSAI 実装を最適化します。

CDN キャッシュの最適化

キャッシュヒット率を分析して改善の機会を特定します。

- コンテンツタイプと更新頻度に基づいて TTL 設定を調整します。TTL の詳細な推奨事項については、「」を参照してください [ステップ 1: 最適な広告配信のために CDN キャッシュを設定する](#)。
- キャッシュキー設定を最適化してキャッシュ効率を向上させる
- 多層キャッシュにオリジンシールドを実装することを検討する

広告配信の最適化

広告パフォーマンスメトリクスを使用して広告配信を改善します。

- 一般的な広告挿入エラーを特定して対処する
- 広告決定サーバーの応答時間を最適化する
- フィルレート分析に基づいて広告ターゲティングパラメータを調整する

ビューワーエクスペリエンスの最適化

クライアント側のメトリクスに基づいてビューワーエクスペリエンスを向上させます。

- 広告時間枠中のドロップオフパターンを分析する
- 広告遷移ポイントを最適化して再生をスムーズにする
- 視聴者のエンゲージメントデータに基づいて広告の頻度と期間を調整する

コスト最適化

パフォーマンスとコストの考慮事項のバランスを取ります。

- 帯域幅の使用状況パターンを分析して CDN コストを最適化する
- CloudFront デイストリビューションの料金クラス調整を検討する
- さまざまなキャッシュ戦略のコストメリットを評価する

ベストプラクティス

効果的な SSAI のモニタリングと分析を行うには、次のベストプラクティスに従ってください。

- ベースラインを確立する: 通常のオペレーション中にメトリクスを収集し、トラブルシューティング中の比較に使用できるパフォーマンスベースラインを確立します。
- マルチレベルモニタリングを実装する: アーキテクチャのさまざまなレベル (オリジン、CDN、プレイヤー) でモニタリングし、パフォーマンスの全体像を把握します。
- サービス間でメトリクスを関連付ける: さまざまなサービスのメトリクス間の関係を探して、問題の根本原因を特定します。
- 異常検出を使用する: CloudWatch 異常検出を実装して、メトリクスの異常なパターンを自動的に識別します。
- 定期的なレビューと改善: モニタリング戦略の定期的なレビューをスケジュールし、要件の変化と新しいインサイトに基づいて調整します。
- 結果とアクションを文書化する: 最適化の取り組みとその結果の記録を維持し、組織の知識を構築します。

関連情報

CDNs、以下を参照してください。

- [CDN と MediaTailor の統合のパフォーマンスを最適化する](#) パフォーマンス最適化手法
- [中断のない広告配信のための CDNs を使用した MediaTailor SSAI のトラブルシューティング](#) 一般的な問題のトラブルシューティング
- [MediaTailor メトリクスの詳細については、Amazon CloudWatch で MediaTailor をモニタリングする MediaTailor](#)
- [CloudFront メトリクスに関する情報の CloudFront およびエッジ関数メトリクスの表示](#) CloudFront

中断のない広告配信のための CDNs を使用した MediaTailor SSAI のトラブルシューティング

このセクションでは、コンテンツ配信ネットワーク (CDN) で AWS Elemental MediaTailor 動的広告挿入を使用する場合の一般的な問題の解決策を示します。これらのソリューションは、パーソナライズされた広告を通じて動画収益化に関する問題のトラブルシューティングに役立ちます。

キャッシュパフォーマンスの問題、HTTP エラー解決、テスト手順、すべての MediaTailor 実装に適用される診断手法を含む包括的な CDN トラブルシューティングガイドンスについては、「」を参照してください[CDN 統合のトラブルシューティング](#)。このセクションでは、SSAI 固有のトラブルシューティング要件と広告挿入の問題に焦点を当てます。

CDN と SSAI の設定で問題が発生した場合は、次の一般的な問題を確認してください。

パーソナライズされた広告がストリームに表示されない

ADS が正しく応答し、ADS と AWS Elemental MediaTailor 通信することを確認します。以下の潜在的な問題を確認してください。

- CDN 経由で正しく転送されないクエリパラメータをターゲットとする広告
- コンテンツで誤って定義された広告ブレイクポイント
- ADS 接続またはレスポンスの問題

広告ブレイクポイントでの再生エラー

広告セグメントがコンテンツのビットレートと解像度に合わせて適切にトランスコードされていることを確認します。以下の一般的な問題を確認してください。

- CDN が広告セグメントのリクエストを誤ってルーティングする
- 移行ポイントでのマニフェスト操作エラー
- コンテンツと広告間のエンコーディングプロファイルの不一致

古いマニフェスト

ライブコンテンツの場合は、CDN キャッシュ TTL 設定が適切であることを確認します。パーソナライズされたマニフェストの場合、TTL は 0 秒を使用します。急速に変化するマニフェストに対してキャッシュ無効化を実装することを検討してください。包括的な TTL ガイダンスについては、「」を参照してください[CDN と MediaTailor の統合のキャッシュ最適化](#)。

高レイテンシー

CDN 設定で最適なルーティングを確認します。最高のパフォーマンスを得るには、CDN にビューワの近くにエッジロケーションがあることを確認します。

高度なトラブルシューティング

より複雑な問題については、次の高度なトラブルシューティング手法を試してください。

一貫性のない広告ターゲティング動作

プレイヤーリクエストと ADS リクエスト間のクエリパラメータの不整合を確認します。必要なすべてのターゲティングパラメータが CDN に適切に渡されていることを確認します。

CDN キャッシュの不整合

キャッシュキー設定を検証して、コンテンツの適切な区別を確認します。重要なマニフェスト更新にキャッシュパージを実装することを検討してください。

広告追跡の失敗

ビーコン URLs が正しく転送され、CDN によってブロックされていないことを確認します。クライアントプレイヤーが追跡エンドポイントに到達できることを確認します。

パフォーマンスの最適化

動的広告挿入と動画収益化ワークフローのパフォーマンスを最適化するには:

- コンテンツタイプとビューワーパターンに基づいて TTL 設定を微調整します。TTL の詳細な推奨事項については、「」を参照してください[CDN と MediaTailor の統合のキャッシュ最適化](#)。
- 地理的ルーティングを実装して、世界中の視聴者のレイテンシーを最小限に抑える
- 冗長性と最適なパフォーマンスのために複数の CDN を使用することを検討する
- キャッシュヒット率をモニタリングし、それに応じて設定を調整する
- マニフェスト操作プロセスを最適化して、広告ブレイクポイントの処理時間を短縮する
- 一般的なコンテンツプロファイルに合わせて広告をトランスコードし、シームレスにパーソナライズされた広告挿入を可能にする

パフォーマンスベンチマーク

AWS Elemental MediaTailor 広告挿入 CDN 統合を最適化するときは、次のパフォーマンスベンチマークを目指します。

キャッシュヒット率のターゲット

コンテンツセグメント: 95% を超えるキャッシュヒット率

広告セグメント: 90% を超えるキャッシュヒット率

マニフェスト: 該当なし (パーソナライズされた広告挿入用にキャッシュしないでください)

レイテンシーベンチマーク

マニフェストリクエストのレイテンシー: 100 ミリ秒未満 (P95)

コンテンツセグメント配信: 50 ミリ秒未満 (P95)

広告セグメント配信: 75 ミリ秒未満 (P95)

End-to-endの起動時間: 2 秒未満

オリジンロードメトリクス

ビューワーあたりのオリジンリクエスト: ビューワーあたり 1 分あたり 0.1 リクエスト未満

ビューワーあたりのオリジン帯域幅: ビューワー帯域幅全体の 5% 未満

エラーレートのターゲット

マニフェストエラー: 0.1% 未満

セグメントエラー: 0.01% 未満

プレイヤーが報告したリバッファリング: 1% 未満

スケーラビリティベンチマーク

ピークイベント中にパフォーマンスを低下させることなく通常のトラフィックの 10 倍をサポート

チャンネルあたり 1 秒あたり 1000 件を超えるリクエストを処理する機能

Amazon CloudWatch メトリクスを使用して、これらのパフォーマンス指標を追跡します。詳細なモニタリング手順については、「」を参照してください[Amazon CloudWatch メトリクス AWS Elemental MediaTailor によるモニタリング](#)。

関連情報

CDNs、以下を参照してください。

広告挿入ドキュメント

[MediaTailor 広告挿入の使用開始](#) - 広告挿入の概念について説明します。

[設定](#) - 広告挿入の使用を開始する

CDN 統合

[CDN 統合をセットアップする](#) - 一般的な CDN 設定ガイダンス

[CloudFront 統合](#) - CloudFront 固有の設定

チャンネルアセンブリの統合

[CDN を使用したチャンネルアセンブリ](#) - CDN

[広告挿入の実装](#) - チャンネルアセンブリを使用して広告挿入を実装する

モニタリングと最適化

[CDN と MediaTailor の統合のオペレーションをモニタリングする](#) - 包括的なモニタリングと分析

[CDN と MediaTailor の統合のパフォーマンスを最適化する](#) - パフォーマンス最適化ガイド

[Amazon CloudWatch メトリクス AWS Elemental MediaTailor によるモニタリング](#) - MediaTailor の CloudWatch メトリクス

チャンネルアセンブリと CDN を使用して MediaTailor リニアチャンネルを構築する

このセクションでは、AWS Elemental MediaTailor チャンネルアセンブリをコンテンツ配信ネットワーク (CDN) と統合するための包括的なガイダンスを提供します。チャンネルアセンブリ CDN 統合をセットアップ、設定、最適化するには、次の手順に従います。

また、チャンネルアセンブリとサーバー側の広告挿入 (SSAI) を組み合わせて、パーソナライズされた広告で収益化されたリニアチャンネルを作成することもできます。この強力な統合により、同じチャンネルコンテンツを見ているさまざまな視聴者にターゲットを絞った広告を配信できるため、ブロードキャスト品質の視聴エクスペリエンスを維持しながら収益機会を増やすことができます。CDNs「」を参照してください[CDN を使用した広告挿入](#)。

このトピックでは、マニフェストという用語を使用して、マルチバリエーションプレイリスト、メディアプレイリスト、MPDs。

MediaTailor チャンネルアセンブリの詳細については、「」を参照してください[AWS Elemental MediaTailor を使用して線形アセンブルストリームを作成する](#)。

必要なもの

CDN で MediaTailor チャンネルアセンブリを設定する前に、以下の必要なリソースを収集します。

AWS アカウントとアクセス許可

MediaTailor リソースを作成および管理するための適切なアクセス許可を持つ AWS アカウント

MediaTailor、CloudFront (使用している場合)、および関連サービスの IAM アクセス許可

アクセス許可の要件の詳細については、「」を参照してください[のセキュリティ AWS Elemental MediaTailor](#)。

必要なサービス

実行中の MediaTailor チャンネルアセンブリチャンネル (SSAI 設定だけでなく)

コンテンツ配信ネットワーク (CDN) アカウント (Amazon CloudFront またはサードパーティー CDN)

VOD コンテンツのオリジンストレージ (Amazon S3、MediaPackage、またはその他のオリジンサーバー)

コンテンツ要件

VOD ソースは適切にエンコードされ、HLS または DASH 形式でパッケージ化されます。ソースの場所と VOD ソースの操作については、「」を参照してください[ソースロケーションの使用](#)。

一貫したセグメント期間を持つコンテンツ (推奨最小: 1 秒)

広告時間枠の広告スレートコンテンツ (広告挿入を実装する場合)。スレートの設定については、「」を参照してください[MediaTailor スレート広告挿入](#)。

[開始する前に]

Important

このワークフローには、実行中の MediaTailor チャンネルアセンブリチャンネルが必要です。MediaTailor SSAI 設定のみを持つだけでは、この統合には不十分です。CDN 統合に進む前に、アクティブなチャンネルアセンブリチャンネルが設定され、動作している必要があります。

CDN で MediaTailor チャンネルアセンブリを実装する前に、以下のセットアップタスクを完了します。

1. CDN、MediaTailor、オリジンサーバー間のネットワーク接続を設定する
2. 安全なコンテンツ配信のために HTTPS を設定する
3. CDN ドメインの DNS 設定を構成する

ナレッジの前提条件

このソリューションを正常に実装するには、以下が必要です。

- ストリーミングプロトコル (HLS/DASH) の理解
- CDN 設定原則の基本知識
- MediaTailor チャンネルアセンブリの概念に精通していること

MediaTailor の基本的なセットアップについては、[設定](#)「」および「」を参照してください。[MediaTailor チャンネルアセンブリの使用開始](#)。ソースの場所と VOD ソースの操作については、「」を参照してください。[ソースロケーションの使用](#)。スレートの設定については、「」を参照してください。[MediaTailor スレート広告挿入](#)。

CDN 統合の利点

チャンネルアセンブリを CDN と統合すると、これらの主な利点が得られます。

ビューワーエクスペリエンスの向上

CDNs を適切に設定することで、リニアチャンネルのバッファリング、起動時間、再生エラーを削減できます。これにより、視聴者のエンゲージメントと満足度が向上します。

コスト削減

効率的なキャッシュ戦略は、オリジンリクエストを最小限に抑えます。これにより、データ転送コストとオリジンサーバーの負荷が軽減され、特に大量のリニアチャンネルにとって重要です。

スケーラビリティ

最適化された CDN 設定は、パフォーマンスを低下させることなく、一般的なイベント中のトラフィックの急増を処理し、ピーク視聴時間でもリニアチャンネルを利用できるようにします。

グローバルリーチ

適切に設定された CDN、場所に関係なく、世界中の視聴者に低レイテンシーでコンテンツを配信し、潜在的な視聴者を拡大します。

シームレスなプログラム移行

最適化された CDN 設定により、リニアチャンネル内のプログラム間のスムーズな移行が可能になり、ブロードキャスト品質の視聴体験が得られます。

トピック

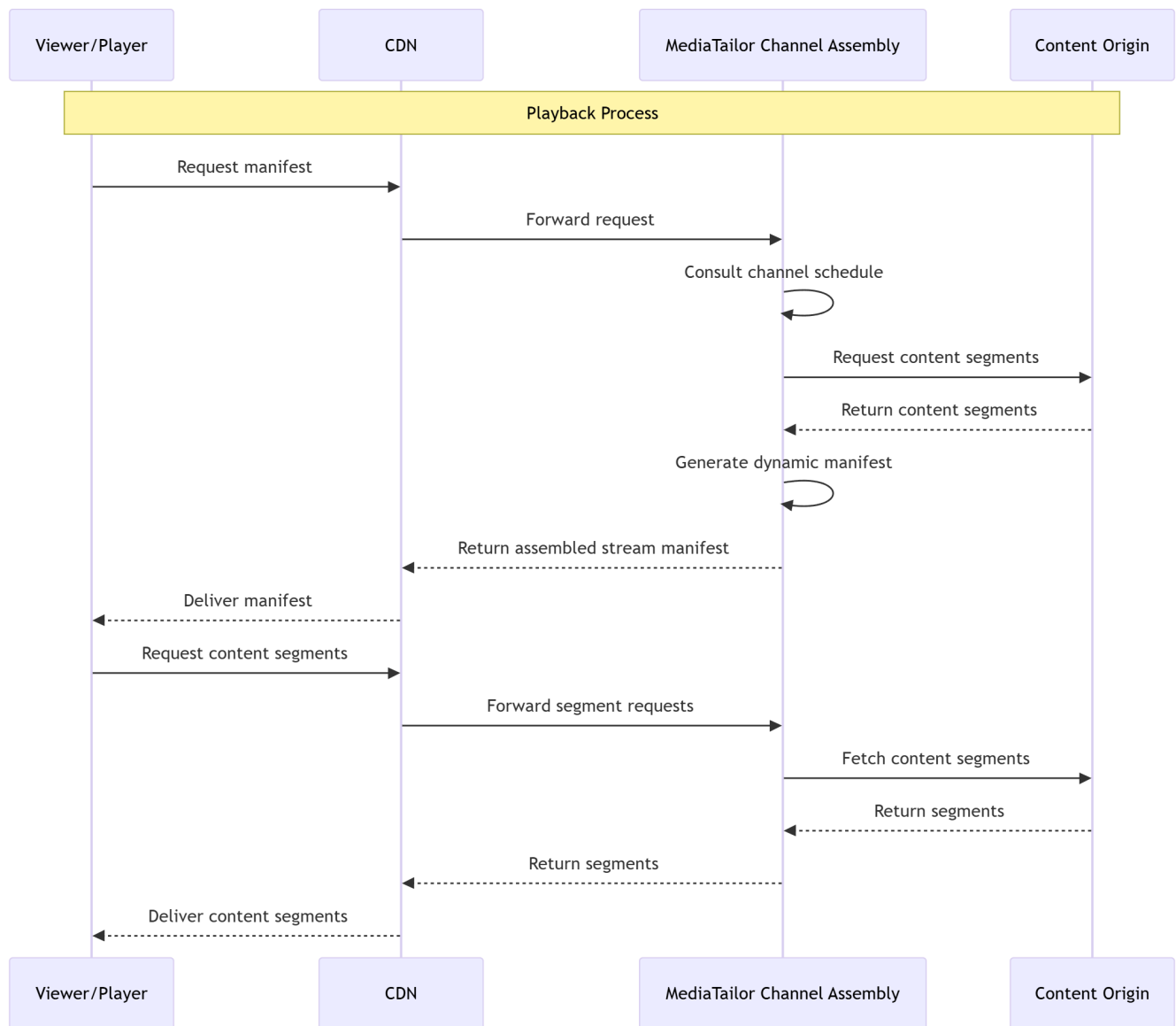
- [MediaTailor チャンネルアセンブリ CDN アーキテクチャを理解する](#)
- [CDN を使用して基本的な MediaTailor チャンネルアセンブリを設定する](#)
- [MediaTailor チャンネルアセンブリ CDN の基本 URLs を設定する](#)
- [チャンネルアセンブリを使用して MediaTailor 広告挿入を実装する](#)
- [MediaTailor チャンネルアセンブリのタイムシフト表示を設定する](#)
- [MediaTailor チャンネルアセンブリ CDN オペレーションのモニタリング](#)
- [完全なチャンネルアセンブリ CDN 最適化](#)

MediaTailor チャンネルアセンブリ CDN アーキテクチャを理解する

AWS Elemental MediaTailor チャンネルアセンブリは、コンテンツ配信ネットワーク (CDNs と統合され、パフォーマンスとグローバルリーチが向上したリニアストリーミングチャンネルを提供します。推奨されるアーキテクチャでは、ビューワーとチャンネルアセンブリの間に CDN を配置し、チャンネルアセンブリはオリジンから直接コンテンツにアクセスします。このトピックでは、コアアーキテクチャコンポーネントと、それらが連携してコンテンツを配信する方法について説明します。

1. CDN からのビューワーリクエストマニフェスト
2. CDN がチャンネルアセンブリにリクエストを転送する
3. チャンネルアセンブリは VOD ソースからマニフェストをアセンブルします
4. チャンネルアセンブリはマニフェストを CDN に返し、CDN はマニフェストをビューワーに転送します。
5. ビューワーが CDN を介してセグメントをリクエストする
6. CDN はセグメントリクエストを適切なオリジンにルーティングします

このアーキテクチャは、CDN を使用するセキュリティと柔軟性の利点を維持しながら、最適なパフォーマンスを保証します。



チャネルアセンブリの CDN 用語

これらの重要な用語を理解することは、チャネルアセンブリ CDN 統合の実装とトラブルシューティングに役立ちます。

オリジン CDN とエッジ CDN

オリジン CDN: MediaTailor とコンテンツオリジンの間に配置される CDN。コンテンツセグメントをキャッシュして、オリジンサーバーの負荷を軽減します。マルチ CDN アーキテクチャでは、これはオリジンと直接インターフェイスする最初の CDN レイヤーです。

Edge CDN: ビューワーと MediaTailor の間に配置される CDN。パーソナライズされたマニフェストとコンテンツをビューワーに配信します。マルチ CDN アーキテクチャでは、これはビューワーと直接インターフェイスする最も外側の CDN レイヤーです。

CDN 設定条件

キャッシュ動作: CDN がキャッシュ期間やオリジンルーティングなど、さまざまなタイプのリクエストを処理する方法を決定するルール。

TTL (有効期限): オリジンから更新する必要があるコンテンツが CDN キャッシュで有効である期間。TTL の詳細な推奨事項については、「」を参照してください[CDN と MediaTailor の統合のキャッシュ最適化](#)。

キャッシュキー: CDN がキャッシュされたコンテンツを保存および取得するために使用する一意の識別子。多くの場合、URL パス、クエリパラメータ、ヘッダーが含まれます。

オリジンシールド: CDN エッジロケーションとオリジンサーバー間の中間キャッシュレイヤーで、オリジンへのリクエストの数を減らします。

リクエストの折りたたみ: 同じコンテンツに対する複数の同時リクエストを 1 つのオリジンリクエストに結合する CDN 機能。

MediaTailor 固有の CDN 用語

CDN コンテンツセグメントプレフィックス: MediaTailor がマニフェストでコンテンツセグメント URLs を生成するときに使用する CDN ドメイン名。

CDN 広告セグメントプレフィックス: MediaTailor がマニフェストで広告セグメント URLs を生成するときに使用する CDN ドメイン名。

MediaTailor での CDN 設定の詳細については、「」を参照してください[CDN 統合をセットアップする](#)。

CDN を使用して基本的な MediaTailor チャネルアセンブリを設定する

AWS Elemental MediaTailor チャネルアセンブリを使用すると、コンテンツ配信ネットワーク (CDN) との基本的な統合を設定して、ビューワーにリニアストリーミングチャネルを効率的に配信できます。チャネルアセンブリと CDN の統合を設定するには、次の手順に従います。

1. ビューワーからのマニフェストを受け入れて MediaTailor チャネルアセンブリに転送するように CDN を設定します。
2. MediaTailor チャネルアセンブリを設定してチャネルスケジュールにアクセスし、現在のプログラミングを決定します。
3. スケジュールに基づいてオリジンサーバーからコンテンツセグメントをリクエストするように MediaTailor チャネルアセンブリを設定します。
4. コンテンツオリジンがリクエストされたセグメントを MediaTailor チャネルアセンブリに配信できることを確認します。
5. MediaTailor チャネルアセンブリを設定して、現在のスケジュールに基づいて動的マニフェストを生成します。
6. アセンブルされた多変量プレイリスト、メディアプレイリスト、MPDs をビューワーに配信するように CDN を設定します。
7. 適切なキャッシュ設定を使用して、ビューワーからのセグメントリクエストを処理するように CDN を設定します。
8. キャッシュミス を MediaTailor チャネルアセンブリに転送するように CDN を設定します。
9. MediaTailor チャネルアセンブリを設定して、コンテンツオリジンからリクエストされたセグメントを取得します。
10. 再生のためにコンテンツセグメントをビューワーに配信するように CDN を設定します。

MediaTailor チャネルアセンブリ CDN の基本 URLs を設定する

AWS Elemental MediaTailor チャネルアセンブリでは、コンテンツ配信ネットワーク (CDN) を介したコンテンツルーティングが正しく機能するように、適切なベース URL 設定が必要です。ビューワーへのコンテンツ配信を成功させるために、チャネルアセンブリの基本 URL 設定を構成します。

コンテンツセグメント URL 設定

チャネルアセンブリチャネル設定で、ベース URL を CDN ドメインに設定します。これにより、アセンブルされたマニフェスト内のすべてのセグメント URLs が、オリジンサーバーに直接ではなく CDN を指すようになります。

たとえば、オリジンコンテンツが `http://origin.example.com/content/` にあり、CDN ドメインがである場合 `https://cdn.example.com/`、基本 URL を に設定します `https://cdn.example.com/content/`。

アクセス制限の設定

セキュリティを強化するには、オリジンサーバーへの直接アクセスを制限するように CDN を設定します。

1. CDN でオリジンアクセスコントロールを設定します。
2. CDN からのリクエストのみを受け入れるようにオリジンサーバーを設定します。
3. 必要に応じて、ビューワー認証用の署名URLs または Cookie を使用します。

Amazon CloudFront では、オリジンアクセスコントロール (OAC) を使用してオリジンへのアクセスを保護できます。CloudFront 統合の保護の詳細については、「」を参照してください [CloudFront 統合](#)。

チャネルアセンブリを使用して MediaTailor 広告挿入を実装する

のチャネルアセンブリは、サーバー側の広告挿入 (SSAI) やコンテンツ配信ネットワーク (CDNs) とシームレスに AWS Elemental MediaTailor 統合され、パーソナライズされた広告で収益化されたリニアチャネルを作成します。

チャネルアセンブリを SSAI と組み合わせると、ブロードキャスト品質のエクスペリエンスを維持しながら、パーソナライズされた広告を視聴者に配信するリニアチャネルを構築できます。このインテグレーションでは、以下の操作を行うことができます。

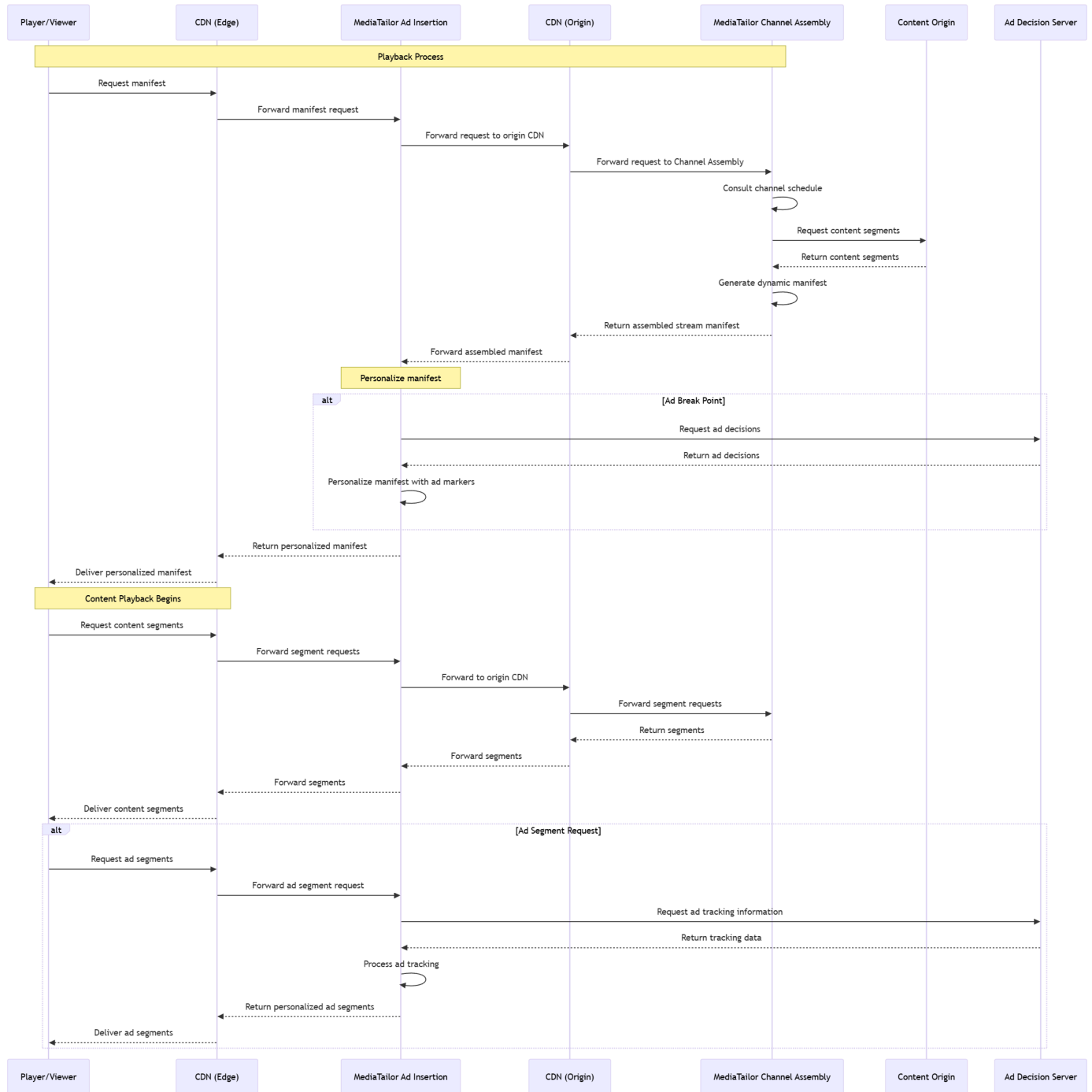
- コンテンツの収益化 - リニアチャネルでのターゲットを絞った広告による収益の生成
- エクスペリエンスをパーソナライズする - プロファイルに基づいて同じチャネルを見ている視聴者にさまざまな広告を配信する
- 品質を維持する - ブロードキャスト品質の視聴のためにコンテンツと広告をシームレスに移行する
- 効率的なスケーリング - CDN 配信を通じて数百万の同時視聴者をサポート

CDNs 「」を参照してください [CDN を使用した広告挿入](#)。

1. ビューワーからのマニフェストリクエストを受け入れ、MediaTailor 広告挿入に転送するように エッジ CDN を設定します。

2. MediaTailor 広告挿入を設定して、リクエストをオリジン CDN に転送します。
3. MediaTailor チャネルアセンブリにリクエストを転送するようにオリジン CDN を設定します。
4. MediaTailor チャネルアセンブリを設定して、現在のスケジュールに基づいて動的マニフェストを生成します。
5. アセンブルされたマニフェストを MediaTailor 広告挿入に転送するようにオリジン CDN を設定します。
6. MediaTailor 広告挿入を設定して、広告ブレイクポイントで広告決定サーバーに広告決定をリクエストします。
7. MediaTailor 広告挿入を設定して、マニフェストを広告マーカでパーソナライズします。
8. パーソナライズされたマニフェストをビューワーに配信するようにエッジ CDN を設定します。
9. コンテンツと広告セグメントの両方のリクエストを効率的に処理するように CDN アーキテクチャを設定します。

次の図は、この組み合わせワークフローを示しています。



チャンネルアセンブリと SSAI を組み合わせるときに最適なパフォーマンスを得るには:

- チャンネルアセンブリと SSAI リクエストを区別するキャッシュ動作を設定する

- で推奨されているマニフェストとセグメントに適切な TTL 値を設定する [CDN と MediaTailor の統合のキャッシュ最適化](#)
- チャンネルアセンブリ、広告挿入、CDN オリジン間の適切なルーティングを確保する
- チャンネルアセンブリと広告挿入コンポーネントの両方のパフォーマンスメトリクスをモニタリングする

CDNs、以下を参照してください。

- [CDN と MediaTailor の統合のための広告挿入アーキテクチャを理解する](#) - SSAI アーキテクチャと概念について説明します。
- [最適な広告配信のために CDN を使用して基本的な MediaTailor SSAI を設定する](#) - SSAI Step-by-step 設定手順
- [中断のない広告配信のための CDNs を使用した MediaTailor SSAI のトラブルシューティング](#) - SSAI 統合に関する一般的な問題のトラブルシューティング

MediaTailor チャンネルアセンブリのタイムシフト表示を設定する

AWS Elemental MediaTailor チャンネルアセンブリは、リニアチャンネルの一時停止、巻き戻し、スタートオーバーなどの DVR のような機能を有効にするタイムシフト表示機能をサポートしています。タイムシフト表示をサポートするようにコンテンツ配信ネットワーク (CDN) を設定することで、これらの機能を有効にします。これにより、ビューワーは表示エクスペリエンスを制御できます。

タイムシフト表示について

タイムシフト表示は、次のようなリニアチャンネルの DVR のような機能を有効にします。

- スタートオーバー: ビューワーは、プログラムの開始後に参加した場合でも、最初からプログラムの視聴を開始できます。
- 一時停止と再開: 視聴者はコンテンツを一時停止し、後で視聴を再開できます
- 巻き戻しと早送り: 閲覧者は利用可能なコンテンツを前後に移動できます
- 表示の遅延: 視聴者は、チャンネルのスケジュールの前半で放送されたコンテンツを視聴できます。

タイムシフト表示は、チャンネルの再生 URL に start パラメータを追加することで機能します。パラメータは、現在の時刻を基準にしたオフセットを秒単位で指定します。

- 負の値は過去の時間を示します (「1 時間前から開始start=-3600」など)
- 正の値は、将来の時間を示します (「1 時間から開始するstart=3600」など)。

time-shift パラメータを含む URL の例:

```
https://example-cdn.com/out/v1/channel-name/index.m3u8?start=-3600
```

時間遅延の解決

CDN によるタイムシフト表示をサポートするには:

1. start クエリパラメータをチャンネルアセンブリに転送するように CDN を設定します。
2. キャッシュキーに startパラメータを含むキャッシュ動作を設定します。
3. タイムシフトパラメータを持つマニフェストの場合は、短い TTL を使用するか、キャッシュしません。

これにより、各ビューワーは要求された時間位置に対して正しいマニフェストを受信できます。

タイムシフトの CDN 要件

チャンネルアセンブリによるタイムシフト表示をサポートするには、CDN が以下の要件を満たしている必要があります。

- すべてのクエリパラメータをチャンネルアセンブリに転送します。
- キャッシュキーに startパラメータを含めます。
- タイムシフトされたマニフェストの適切なキャッシュ無効化をサポートします。
- クエリパラメータに基づいてさまざまなマニフェストレスポンスを処理します。

MediaTailor チャンネルアセンブリ CDN オペレーションのモニタリング

AWS Elemental MediaTailor チャンネルアセンブリでは、信頼性の高いコンテンツ配信を確保するために、コンテンツ配信ネットワーク (CDN) と統合するときに効果的なモニタリングが必要です。チャンネルアセンブリと CDN 統合のモニタリング戦略を実装して、信頼性の高いコンテンツ配信と迅速な問題解決を実現します。

必須メトリクス、モニタリングツールのセットアップ、アラート設定、すべての MediaTailor 実装に適用されるトラブルシューティング戦略を含む包括的な CDN モニタリングガイドンスについては、

「」を参照してください[CDN モニタリング](#)。このセクションでは、チャンネルアセンブリ固有のモニタリング要件に焦点を当てます。

チャンネルアセンブリと CDN 統合に特定のモニタリングを実装します。

- チャンネルアセンブリでマニフェスト生成メトリクスを追跡します。
- タイムシフトされた表示リクエストとその CDN キャッシュヒット率への影響をモニタリングします。
- マニフェストリクエストの異常なパターンのアラートを設定します。
- コンテンツソース全体でセグメントの可用性の追跡を実装します。

Amazon CloudFront では、CDN メトリクスを MediaTailor メトリクスと統合して、[CloudWatch ダッシュボードを使用して配信パイプライン全体を視覚化するダッシュボード](#)を作成します。

チャンネルアセンブリで SSAI も使用している場合は、広告挿入に固有の追加のモニタリング推奨事項[CDN オペレーションのモニタリング](#)については、「」を参照してください。

完全なチャンネルアセンブリ CDN 最適化

キャッシュ戦略、リクエストルーティング設定、パフォーマンスベンチマーク、高度な最適化手法など、チャンネルアセンブリの実装に固有の詳細な CDN 最適化ガイダンスについては、「」を参照してください[CDN パフォーマンスの最適化](#)。

統合最適化ガイドには、以下が用意されています。

- 推奨 TTL 値を使用したチャンネルアセンブリ固有のキャッシュ設定
- マニフェストおよびセグメント配信のリクエストルーティングの最適化
- すべての MediaTailor 実装のパフォーマンスベンチマークとターゲット
- Origin Shield や圧縮などの高度な最適化手法
- SSAI を使用したチャンネルアセンブリのワークフローガイダンスの組み合わせ

MediaTailor を MediaPackage および CDN と統合する

AWS Elemental MediaTailor は と統合 AWS Elemental MediaPackage して、コンテンツ配信ネットワーク (CDN) を介してパーソナライズされた動画広告を配信します。MediaPackage は、インターネット経由で配信するための動画コンテンツを準備して保護するjust-in-timeの動画パッケージ化およ

び配信サービスです。ライブまたはオンデマンドの動画コンテンツを取得し、HLS や DASH などのストリーミング形式にパッケージ化することで、さまざまなデバイスで視聴者の準備が整います。

MediaPackage を MediaTailor と CDN と組み合わせると、パーソナライズされた広告を大規模に配信する完全なストリーミングワークフローが作成されます。CDN はコンテンツをグローバルに配信し、レイテンシーを減らして視聴者エクスペリエンスを向上させ、MediaTailor はターゲットを絞った広告をストリームに挿入します。

このトピックでは、MediaTailor、MediaPackage、および CDN を連携させるための重要な統合ステップに焦点を当てます。高度な設定オプション、トラブルシューティング、モニタリングガイダンスについては、「」を参照してください[次の手順](#)。

MediaPackage と CDN ワークフローについて

統合を設定する前に、MediaPackage、MediaTailor、および CDN がどのように連携するかを理解することが重要です。

1. コンテンツの準備: MediaPackage はライブまたはオンデマンドのビデオコンテンツを受信し、ストリーミング形式 (HLS または DASH マニフェストとセグメント) にパッケージ化します。
2. 広告挿入: MediaTailor は MediaPackage からマニフェストをリクエストし、パーソナライズされた広告を挿入し、変更されたマニフェストをビューワーに提供します。
3. グローバルディストリビューション: CDN は、コンテンツセグメント (MediaPackage から) と広告セグメント (MediaTailor から) の両方をキャッシュして世界中の視聴者に配信します。
4. 視聴者の再生: ビデオプレイヤーは CDN を介してマニフェストをリクエストします。CDN は MediaTailor (マニフェストの場合) と MediaPackage (コンテンツセグメントの場合) の間でリクエストを適切にルーティングします。

このアーキテクチャにはいくつかの利点があります。

- スケーラビリティ: CDN は、オリジンサーバーに影響を与えずに高いビューワー負荷を処理します。
- パフォーマンス: コンテンツはビューワーに最も近いエッジロケーションから配信されます。
- コスト効率: キャッシュによる帯域幅コストの削減
- 信頼性: 複数のエッジロケーションで冗長性を実現

前提条件

開始する前に、次のコンポーネントが設定されていることを確認してください。

1. MediaPackage エンドポイント: 動画コンテンツを受信してパッケージ化する設定済み MediaPackage エンドポイント。セットアップ手順については、[MediaPackage ユーザーガイドの「MediaPackage の開始方法」](#)を参照してください。MediaPackage
2. MediaTailor 設定: MediaPackage エンドポイントをコンテンツオリジンとして使用する MediaTailor 設定。MediaPackage セットアップ手順については、「」を参照してください[MediaTailor 広告挿入用のコンテンツソースの統合](#)。
3. CDN ディストリビューション: ストリーミングメディアで動作するように設定された CDN ディストリビューション (CloudFront など)。セットアップ手順については、CloudFront [デベロッパーガイドの「ディストリビューションの作成」](#)を参照してください。
4. 広告決定サーバー: 広告挿入のために VAST または VMAP レスポンスを返す設定済み広告決定サーバー。

ステップ 1: 重要な CDN 設定を構成する

MediaPackage の統合を成功させるには、適切な CDN 設定が不可欠です。設定が正しくないと、再生が失敗し、キャッシュパフォーマンスが低下し、コストが増加する可能性があります。適切なキャッシュポリシーとクエリパラメータの転送がないと、CDN がマニフェストを正しく配信できないか、キャッシュを完全にバイパスして、オリジンサーバーの負荷が高くなり、ビューワーエクスペリエンスが低下する可能性があります。

基本的なキャッシュ設定を構成する

MediaPackage は特定のキャッシュコントロールヘッダーを使用してコンテンツ配信を最適化するため、基本的なキャッシュの設定は不可欠です。適切なキャッシュ設定がないと、CDN がこれらのヘッダーを無視し、不要なオリジンリクエストが発生し、レイテンシーが増加する可能性があります。最適なキャッシュ動作を確保するには、次の手順に従います。

MediaPackage で動作する基本的なキャッシュを設定するには:

1. CloudFront コンソールで CloudFront ディストリビューション設定を開きます。
2. MediaPackage オリジンのキャッシュポリシーを選択または作成します。
3. 「オリジンキャッシュコントロールヘッダー」オプションを有効にします。

- MediaPackage がキャッシュ制御ヘッダーを使用してキャッシュ動作を制御できるようにします。

この基本設定により、MediaPackage はさまざまなコンテンツタイプに適切なキャッシュ期間を自動的に設定できます。特定の TTL 値とパフォーマンスチューニングを使用して高度なキャッシュ最適化を実装するには、まずこの基本的なセットアップを完了してから、「」に進みます[MediaTailor および MediaPackage コンテンツ配信の CDN キャッシュを最適化する](#)。

必須クエリパラメータを設定する

MediaPackage の機能には、クエリパラメータの設定が不可欠です。CDN は特定のクエリパラメータを転送して、タイムシフト表示や低レイテンシーストリーミングなどの機能を有効にする必要があります。クエリパラメータの設定が正しくないと、これらの機能が機能しなくなり、キャッシュ効率が低下することが懸念されます。クエリパラメータの転送を設定するには、次の手順に従います。

CDN が必要なクエリパラメータを MediaPackage に転送するようにするには:

- CloudFront ディストリビューション設定で、マニフェストリクエストのキャッシュポリシーを選択または作成します。
- 「キャッシュキー設定」で、「指定されたクエリ文字列を含める」を選択します。
- 次の必須クエリパラメータを追加します。
 - start および end - タイムシフト表示機能用。これらのパラメータは MediaPackage に渡され、スタートオーバーとキャッチアップ表示の特定のコンテンツウィンドウを定義します。
 - _HLS_msn および _HLS_part - LL-HLS 再生リクエストのサポート用
 - m - エンドポイントの変更された ti/compame をキャプチャする場合。MediaPackage レスponseには、エンドポイントの変更時刻をキャプチャするための ?m=### タグが常に含まれます。コンテンツがこのタグの別の値で既にキャッシュされている場合、CloudFront はキャッシュされたバージョンを提供する代わりに新しいマニフェストをリクエストします。
 - aws.manifestfilter - マニフェストフィルタリング機能用。マニフェストフィルタリングを使用している場合は、このパラメータを含めて、aws.manifestfilterクエリ文字列を MediaPackage オリジンに転送するようにディストリビューションを設定する必要があります。これは、マニフェストフィルタリング機能が機能するために必要です。
- MediaPackage が使用するクエリ文字列のみを含めます。不要なクエリ文字列を含めると、同じコンテンツに対して複数のキャッシュバリエーションを作成できるため、キャッシュ効率が低下します。

これらのパラメータは、CDN で基本的な MediaPackage 機能を有効にします。さまざまなサブスクリプション層またはデバイスタイプに対してコンテンツフィルタリングを実装する必要がある場合は、まずこの基本的なクエリパラメータの設定を完了してから、「」に進みます [MediaTailor、MediaPackage、CDN でマニフェストフィルタリングを設定する](#)。

MediaTailor がタイムシフト表示のために start や end などのクエリパラメータを MediaPackage に渡す方法については、[オリジンの MediaTailor クエリパラメータ処理](#)「」の「」を参照してください [MediaTailor マニフェストクエリパラメータ](#)。

LL-HLS の応答タイムアウトを設定する

LL-HLS は CDN が新しいコンテンツセグメントを待機する「ブロックリクエスト」メカニズムを使用するため、低レイテンシーの HLS ではタイムアウト設定が重要です。タイムアウトが短すぎると、MediaPackage が新しいセグメントで応答できるようになる前にリクエストが失敗し、再生が中断され、ビューワーエクスペリエンスが低下します。LL-HLS の再生がスムーズに行われるように、適切なタイムアウトを設定します。

低レイテンシーの HLS を使用している場合は、CDN タイムアウト設定を構成します。

1. CDN 設定で、オリジンタイムアウト設定を見つけます。
2. 応答タイムアウト値をパート期間の少なくとも 3 倍に設定します。
3. たとえば、パートの所要時間が 0.3 秒の場合、タイムアウトを少なくとも 0.9 秒に設定します。

これにより、CDN は、ブロックリクエストメカニズムを使用するときに MediaPackage が応答するまで十分に待機します。

ステップ 2: 統合を検証する

ビューワーに問題が発生する前にすべてのコンポーネントが正しく連携するようにするには、統合をテストすることが重要です。統合に失敗すると、再生が中断されたり、広告が欠落したり、パフォーマンスが低下する可能性があります。この検証プロセスは、制御された環境の問題を特定して解決するのに役立ちます。

CDN 設定を設定したら、コンテンツリクエストから広告挿入までのワークフロー全体をテストして、統合が正しく機能していることを確認します。

ステップ 2.1: 基本的な再生をテストする

基本的な再生テストでは、CDN がマニフェストリクエストを正しく処理し、MediaTailor に転送することを確認します。このテストは、キャッシュポリシー、クエリパラメータの転送、マニフェスト処

理の設定問題を特定するのに役立ちます。基本的なマニフェスト配信をテストするには、次の手順に従います。

CDN を介してマニフェストをリクエストして、基本的な統合が機能していることをテストします。

1. ウェブブラウザまたは curl を使用して、CDN を介してマニフェスト URL をリクエストします。
2. マニフェストが正常にロードされ、コンテンツセグメントと広告セグメントの両方が含まれていることを確認します。
3. マニフェスト内のコンテンツセグメント URLs が CDN ドメインを指していることを確認します。
4. 広告セグメント URLs CDN ドメインも指していることを確認します。

マニフェストが正しくロードされ、予想される URLs が含まれている場合、基本的な統合は機能しています。包括的なテスト方法と高度な検証手順については、「」を参照してください[CDN と MediaTailor の統合のテストと検証](#)。統合のパフォーマンスとヘルスの包括的なモニタリングを設定するには、「」を参照してください[MediaPackage、CDN、MediaTailor 統合のパフォーマンスをモニタリングする](#)。

ステップ 2.2: ビデオ再生をテストする

動画再生テストにより、広告挿入や CDN 経由のコンテンツ配信など、完全な統合が end-to-end で機能します。このテストでは、コンテンツセグメントと広告セグメントの両方が適切にキャッシュおよび配信され、ビューワーエクスペリエンスが品質基準を満たしていることを確認します。完全な再生機能をテストするには、次の手順に従います。

挿入された広告でビデオ再生が正しく機能することをテストします。

1. CDN を介してコンテンツを再生するには、ビデオプレーヤー (Video.js や HLS.js など) を使用します。
2. バッファリングの問題なしでビデオがスムーズに再生されることを確認します。
3. 再生中に広告が予定時刻に挿入されていることを確認します。
4. コンテンツセグメントと広告セグメントの両方が CDN (オリジンから直接ではなく) からロードされていることを確認します。

広告で再生がスムーズに機能する場合、統合は正常に機能しています。包括的なテスト方法と高度な検証手順については、「」を参照してください[CDN と MediaTailor の統合のテストと検証](#)。再生の問題、バッファリング、または広告挿入の問題が発生した場合は、「」を参照してください[MediaPackage、CDN、MediaTailor の統合のトラブルシューティング](#)。

次の手順

基本的な統合が完了したら、高度な機能と最適化を実装できます。

高度な CDN 最適化

詳細なキャッシュ最適化、TTL 設定、パフォーマンスチューニングについては、「」を参照してください [MediaTailor および MediaPackage コンテンツ配信の CDN キャッシュを最適化する](#)。

マニフェストでのフィルタリング

階層型サービス、デバイス最適化、またはアクセスコントロールのコンテンツフィルタリングを実装するには、「」を参照してください [MediaTailor、MediaPackage、CDN でマニフェストフィルタリングを設定する](#)。

トラブルシューティング

統合で問題が発生した場合は、「」を参照してください [MediaPackage、CDN、MediaTailor の統合のトラブルシューティング](#)。

パフォーマンスのモニタリング

包括的なモニタリングを設定し、主要なパフォーマンスメトリクスを理解するには、「」を参照してください [MediaPackage、CDN、MediaTailor 統合のパフォーマンスをモニタリングする](#)。

MediaTailor、MediaPackage、CDN でマニフェストフィルタリングを設定する

AWS Elemental MediaTailor は、でマニフェストフィルタリング AWS Elemental MediaPackage を使用して、コンテンツ配信ネットワーク (CDN) を介して異なるビューワーに配信されるマニフェストに含まれるオーディオストリームとビデオストリームをカスタマイズします。これは、階層型サービス、デバイス固有の最適化、またはコンテンツアクセスコントロールの実装に特に役立ちます。

このトピックでは、特にマニフェストフィルタリング機能の実装に焦点を当てます。マニフェストフィルタリングを実装する前に、基本的なコンテンツ配信ネットワーク統合設定を完了する必要があります。基本的な MediaPackage とコンテンツ配信ネットワーク統合をまだ設定していない場合は、から始めます [MediaTailor を MediaPackage および CDN と統合する](#)。

マニフェストフィルタリング機能

マニフェストフィルタリングを実装する前に、この機能で何ができるかを理解してください。

コアフィルタリング機能

マニフェストフィルタリングは、コンテンツ配信の制御に役立ついくつかの主要な機能を提供します。

- プレミアムコンテンツ (4K HEVC など) へのビューワーアクセスを制限する
- 適切なストリームを使用して特定のデバイスタイプをターゲットにする
- オーディオサンプルレート、言語、またはビデオコーデックに基づいてコンテンツをフィルタリングする
- 異なるサブスクライバーに異なる品質階層を配信する

一般的なユースケース

これらのユースケースは、マニフェストフィルタリングが特定のビジネス要件にどのように対処できるかを示しています。

サブスクリプション階層

基本的なサブスクライバーに低解像度ストリームを提供し、プレミアムサブスクライバーに 4K コンテンツへのアクセスを提供する

例: 基本階層は 720p に制限され、プレミアム階層は最大 4K になります

デバイスの最適化

デバイス機能に基づいて適切なストリームを自動的に提供する

例: モバイルデバイスはビットレートが低くなり、スマートTVs高品質になります

帯域幅管理

ピーク使用期間中のストリーム品質を制限してネットワークコストを管理する

例: トラフィックの多いイベント中の最大ビットレートを下げる

リージョンコンテンツ

ビューワーの場所に基づいて異なるオーディオ言語またはコンテンツバリエーションを提供する

例: ローカル言語オーディオトラックを自動的にフィルタリングする

マニフェストフィルタリングの概念の詳細については、AWS Elemental MediaPackage ユーザーガイドの「[マニフェストフィルタリング](#)」を参照してください。

マニフェストフィルタリング用に CDN を設定する

CDN はフィルタリングを機能させるために `aws.manifestfilter` クエリパラメータを MediaPackage に転送する必要があるため、マニフェストフィルタリングの CDN 設定は不可欠です。適切なクエリ文字列転送がないと、フィルターパラメータは CDN によって削除され、すべてのビューワーはサブスクリプション階層やデバイス機能に関係なくフィルタリングされていないマニフェストを受け取ります。この設定により、フィルタリングロジックが MediaPackage に到達し、意図したとおりに機能します。

CDN を介してマニフェストフィルタリングを有効にするには、クエリ文字列の転送を設定する必要があります。

1. CloudFront デイストリビューションで、マニフェストリクエストのキャッシュ動作を作成または編集します。
2. キャッシュポリシーの場合は、新しいポリシーを作成するか、既存のポリシーを編集します。
3. キャッシュキー設定で、「指定されたクエリ文字列を含める」を選択します。
4. 許可されたクエリ文字列のリスト `aws.manifestfilter` に を追加します。
5. 他の MediaPackage 機能も使用している場合は、それらのクエリパラメータを追加します。
 - `start` および `end` - タイムシフト表示の場合
 - `time_delay` - 時間遅延機能の場合
 - `_HLS_msn` および `_HLS_part` - LL-HLS の場合

デイストリビューションの作成の詳細については、Amazon CloudFront [デベロッパーガイド](#) の「[デイストリビューションの作成](#)」を参照してください。

クライアント側のフィルタリングを実装する

クライアント側の実装では、動画プレーヤーとアプリケーションがフィルタリングされたコンテンツをリクエストする方法を定義します。この設定は、サブスクリプションレベル、デバイス機能、またはその他の基準に基づいて、各ビューワーが受け取るコンテンツを決定します。適切な実装により、ビューワーは最適な CDN キャッシュ効率を維持しながら、アクセスする必要があるコンテンツのみを受信できます。

ビデオプレーヤーとアプリケーションにマニフェストフィルタリングを実装するには:

フィルタリングの仕組み

フィルタリングプロセスは次のように機能します。

1. 動画プレーヤーまたはアプリケーションがフィルターパラメータを含むマニフェスト URL をリクエストする
2. CDN はリクエスト (クエリパラメータを含む) を MediaTailor に転送します
3. MediaTailor は、オリジンマニフェストをリクエストするときにフィルターパラメータを MediaPackage に渡します。
4. MediaPackage はフィルターを適用し、条件に一致するバリエーションのみを含むカスタマイズされたマニフェストを返します。
5. MediaTailor は、広告挿入用にフィルタリングされたマニフェストを処理し、プレーヤーに返します。

フィルタリング用の URL 形式

フィルタリングの実装を成功させるには、正しい URL 形式を理解することが不可欠です。URL 形式が正しくない場合、フィルタリングパラメータは無視されるか、HTTP エラーが発生します。URL 構造には、CDN が MediaPackage に転送するクエリ文字列としてフィルターパラメータを含める必要があります。適切な URL 形式を実装するには、次の手順に従います。

ビデオプレーヤーにマニフェストフィルタリングを実装するには:

1. プレーヤーのマニフェストリクエスト URLs を変更して、適切なフィルターパラメータを含めます。
2. クエリパラメータで次の URL 形式を使用します。

```
https://CloudFront-Domain/v1/master/MediaTailor-Config/index.m3u8?  
aws.manifestfilter=video_codec:h264;audio_language:en-US
```

3. プレーヤーがこの URL をリクエストすると、MediaTailor はこれらのパラメータを MediaPackage に渡し、フィルタリングされたマニフェストになります。

一般的なフィルタリングシナリオ

一般的なフィルタリングシナリオを実装するには、次の例を使用します。

デバイス固有のコンテンツ配信

デバイス機能に基づいてフィルタリングするには、マニフェストリクエストに次のパラメータを追加します。

```
aws.manifestfilter=video_codec:h264;audio_sample_rate:0-44100
```

この例では、モバイルデバイスに適した最大 44.1 kHz のサンプルレートで、コンテンツを H.264 ビデオとオーディオに制限します。

プレミアムコンテンツの制限

高ビットレートコンテンツへのアクセスを制限するには、このパラメータをマニフェストリクエストに追加します。

```
aws.manifestfilter=video_bitrate:0-9000000
```

この例では、基本的なサブスクリプション層に適したビデオビットレートを 9 Mbps 以下に制限します。

言語の選択

特定のオーディオ言語をフィルタリングするには、このパラメータをマニフェストリクエストに追加します。

```
aws.manifestfilter=audio_language:fr,en-US,de
```

この例では、フランス語、米国英語、ドイツ語のオーディオトラックのみが含まれます。

解決策のターゲティング

特定のビデオ解像度をフィルタリングするには、このパラメータをマニフェストリクエストに追加します。

```
aws.manifestfilter=video_height:240-360,720-1080
```

この例では、中間解像度を除く、高さが 240～360 ピクセルから 720-1080 ピクセルのビデオストリームが含まれています。

コーデックベースのフィルタリング

特定のビデオコーデックをフィルタリングするには、このパラメータをマニフェストリクエストに追加します。

```
aws.manifestfilter=video_codec:h264,h265
```


この例では、他のコーデックを除き、H.264 および H.265 ビデオストリームのみが含まれています。

特別な考慮事項と制限事項

マニフェストフィルタリングを実装する際の一般的な問題を回避するには:

技術的な制限事項

- TS マニフェストの場合、オーディオレンディショングループを使用して、フィルタリングされたオーディオストリームで多重化されたビデオストリームを削除しないようにします。
- TS マニフェストと CMAF マニフェストでは、検証のためにオーディオサンプルレートとビデオビットレートがマニフェストに簡単に表示されません。
- メディアプレイリストまたはセグメントに追加されたリクエストパラメータは、HTTP 400 エラーになります。

エラー状態

- フィルタリングの結果が空のマニフェストである場合 (フィルター条件を満たすストリームがない場合)、MediaPackage は HTTP 400 エラーを返します。
- フィルター設定 (エンドポイントフィルター + クエリパラメータ) が競合すると、HTTP 404 エラーが発生します
- 無効なフィルター構文またはサポートされていないフィルタータイプでは、HTTP 400 エラーが発生します。

パフォーマンスに関する考慮事項

- 一意のフィルターの組み合わせごとに個別のキャッシュエントリが作成され、キャッシュ効率が低下する可能性があります
- 多くの条件を持つ複雑なフィルターは、マニフェスト生成のパフォーマンスに影響を与える可能性があります
- キャッシュパフォーマンスを向上させるために、静的フィルタリングシナリオにエンドポイントレベルのフィルターを使用することを検討する

フィルタリング実装をテストする

マニフェストフィルタリングの実装をテストすることは、ビューワーがアクセスレベルとデバイス機能に基づいて正しいコンテンツを受信できるようにするために不可欠です。フィルタリングに失敗すると、ビューワーが誤った品質レベル、サポートされていない形式、またはアクセスできないコンテンツを受け取る可能性があります。包括的なテストは、これらの問題がビューワーに影響を与える前に特定して解決するのに役立ちます。

マニフェストフィルタリングが正しく機能していることを確認するには:

1. 異なるフィルターパラメータを使用してマニフェストをリクエストし、結果を検証する
2. フィルタリングされたマニフェストに予想されるストリームのみが含まれていることを確認する
3. エッジケース (空の結果、無効なフィルター) をテストして、適切なエラー処理を行う
4. CDN がフィルターパラメータを適切に転送していることを確認します。
5. さまざまなデバイスやプレーヤーでテストして互換性を確保する

フィルタリング問題のトラブルシューティングについては、MediaPackage CDN 統合問題のトラブルシューティング」を参照してください。

HTTP 400 エラー、空のマニフェスト、またはフィルタリングパラメータが期待どおりに動作しない場合は、特定のマニフェストフィルタリングのトラブルシューティングガイド [スMediaPackage、CDN、MediaTailor の統合のトラブルシューティング](#)については、「」を参照してください。

MediaTailor および MediaPackage コンテンツ配信の CDN キャッシュを最適化する

AWS Elemental MediaTailor では、コンテンツオリジン AWS Elemental MediaPackage として使用する場合、最適なパフォーマンスを得るために、適切なコンテンツ配信ネットワーク (CDN) キャッシュ設定が必要です。MediaPackage には、さまざまなタイプのコンテンツをキャッシュする期間をコンテンツ配信ネットワークに伝える特定のキャッシュコントロールヘッダーが用意されています。これらの推奨事項に従うことで、スムーズな再生と効率的なコンテンツ配信が可能になります。

このトピックでは、パフォーマンスを最大化し、コストを最小限に抑えるためのキャッシュ動作の最適化に特に重点を置いています。高度なキャッシュ最適化を実装する前に、基本的なコンテンツ配信ネットワーク統合の設定が完了していることを確認してください。基本統合をまだ設定していない場合は、[から始めますMediaTailor を MediaPackage および CDN と統合する](#)。

MediaPackage キャッシュコントロールヘッダー

MediaPackage は、キャッシュ動作を最適化するために、さまざまなコンテンツタイプに特定の TTL 値を設定します。

多変量プレイリスト (HLS および LL-HLS)

TTL: メディアセグメントの期間の半分

理由: これらのプレイリストは新しいセグメントが利用可能になると変更されるため、頻繁に更新する必要があります

メディアプレイリスト (通常の HLS)

TTL: メディアセグメントの期間の半分

理由: 多変量プレイリストと同様に、コンテンツの進行に伴うこれらの更新

メディアプレイリスト (LL-HLS)

TTL: 1 秒

理由: 低レイテンシーストリーミングでは、非常に頻繁な更新が必要です

TS メディアセグメントと init セグメント

TTL: 1209600 秒 (14 日間)

理由: メディアセグメントは作成後に変更されないため、長期間キャッシュできます

CMAF メディアセグメントと初期化セグメント

TTL: 1209600 秒 (14 日間)

理由: TS セグメントと同様に、作成後は変更できません

すべての MediaTailor ワークフローと追加のキャッシュ最適化戦略に関する包括的な TTL 推奨事項については、「」を参照してください[CDN と MediaTailor の統合のキャッシュ最適化](#)。

CDN キャッシュポリシーを設定する

最適なパフォーマンスとコスト効率を得るには、適切なキャッシュポリシー設定が不可欠です。異なるタイプのコンテンツ (マニフェスト、セグメント、初期化ファイル) には、異なるキャッシュ要件

があります。個別のキャッシュ動作を使用すると、コンテンツタイプごとにキャッシュを最適化し、キャッシュヒット率を向上させ、オリジンの負荷を軽減できます。適切なキャッシュポリシーがないと、不要なオリジンリクエスト、コストの増加、再生パフォーマンスの低下が発生する可能性があります。

MediaPackage キャッシュコントロールヘッダーを適切に尊重し、キャッシュを最適化するには:

1. CloudFront コンソールで CloudFront デイストリビューション設定を開きます。
2. コンテンツタイプごとに個別のキャッシュ動作を作成します。
 - マニフェストリクエスト (*.m3u8、*.mpd)
 - メディアセグメント (*.ts、*.mp4、*.m4s)
 - 初期化セグメント
3. キャッシュ動作ごとに、以下の設定でキャッシュポリシーを作成または選択します。
 - 「オリジンキャッシュコントロールヘッダー」オプションを有効にする
 - 「オリジンリクエストポリシー」を設定して必要なヘッダーを転送する
 - コンテンツタイプに基づいてクエリ文字列転送を設定する

マニフェストキャッシュの動作

マニフェストリクエスト (*.m3u8、*.mpd) の場合:

- パスパターン: *.m3u8 および *.mpd
- キャッシュポリシー: Honor オリジンキャッシュコントロールヘッダー
- クエリ文字列: 特定のパラメータを転送する (「」を参照[クエリ文字列転送の最適化](#))
- ヘッダー: すべてのヘッダーを転送する (最小要件については、「」を参照してください[MediaTailor CDN 統合に必要なヘッダー](#))

メディアセグメントのキャッシュ動作

メディアセグメント (*.ts、*.mp4、*.m4s) の場合:

- パスパターン: *.ts、*.mp4、*.m4s
- キャッシュポリシー: Honor オリジンキャッシュコントロールヘッダー (14 日間の TTL)
- クエリ文字列: なし (セグメントはクエリパラメータを使用しません)
- 圧縮: を有効にして配信パフォーマンスを向上させる

クエリ文字列転送の最適化

クエリ文字列の最適化はキャッシュ効率にとって重要です。これは、不要なクエリパラメータが同じコンテンツに対して複数のキャッシュバリエーションを作成するためです。一意のクエリパラメータの組み合わせごとに個別のキャッシュエントリが作成され、キャッシュヒット率が低下し、オリジンリクエストが増加します。MediaPackage が実際に使用するクエリ文字列のみを転送することで、完全な機能を維持しながらキャッシュ効率を最大化できます。

MediaPackage が使用するクエリ文字列のみを転送するように CDN を設定し、キャッシュ効率を向上させます。

重要なクエリ文字列

start および end - タイムシフト表示ウィンドウの場合

time_delay - マニフェストコンテンツに遅延時間を適用する場合

_HLS_msn、_HLS_m、および _HLS_part - LL-HLS 再生リクエストの場合

機能固有のクエリ文字列

aws.manifestfilter - [マニフェストフィルタリング](#) の場合

Important

キャッシュキーに他のクエリ文字列を含めないでください。MediaPackage は認識されないパラメータを無視し、含めることで、不要なキャッシュバリエーションを作成することでキャッシュ効率が低下します。

パフォーマンス最適化手法

これらの最適化は、MediaPackage や MediaTailor ではなく CDN (CloudFront など) で設定されます。キャッシュパフォーマンスを最大化するには、以下の追加の最適化を実装します。

オリジンシールド

オリジンシールドは、CDN エッジロケーションと MediaPackage エンドポイント間に追加のキャッシュレイヤーを提供します。これにより、MediaPackage エンドポイントに到達するリクエストの数が減少し、特にトラフィックの急増時やキャッシュヒット率が最適より低い場合に、パフォーマンス

が向上し、コストを削減できます。オリジンシールドは、複数のエッジロケーションが同じコンテンツを同時にリクエストするライブストリーミングに特に役立ちます。

オリジンシールドを有効にして MediaPackage エンドポイントの負荷を軽減します。

1. CloudFront デイストリビューションで、MediaPackage オリジンの Origin Shield を有効にします。
2. MediaPackage エンドポイントに近いオリジンシールドリージョンを選択します。
3. これにより、MediaPackage へのリクエストを減らす追加のキャッシュレイヤーが作成されます。

圧縮設定

テキストベースのレスポンスの圧縮を有効にします。

- マニフェストファイルの圧縮を有効にする (*.m3u8、*.mpd)
- メディアセグメントを圧縮しない (既に圧縮されている)
- すべてのヘッダーが MediaPackage に転送されていることを確認します (最小要件については、「」を参照してください[MediaTailor CDN 統合に必要なヘッダー](#))。

キャッシュパフォーマンスのモニタリング

これらの主要なメトリクスを追跡して、最適なキャッシュパフォーマンスを確保します。

キャッシュヒット率

ターゲット: メディアセグメントで 90% 以上、マニフェストで 70% 以上

比率が低い場合は、TTL 設定が正しくないか、不要なクエリパラメータを示している可能性があります。

オリジンリクエストボリューム

MediaPackage エンドポイントに到達するリクエストをモニタリングする

大量のはキャッシュの問題を示している可能性があります

キャッシュキーのバリエーション

キャッシュキーパターンを確認して不要なバリエーションを特定する

バリエーションが多すぎるとキャッシュ効率が低下する

これらのキャッシュ最適化を実装したら、モニタリングを設定してその有効性を追跡します。キャッシュヒット率、オリジンリクエストパターン、およびその他の主要なパフォーマンスメトリクスのモニタリングに関するガイダンスについては、「」を参照してください[MediaPackage、CDN、MediaTailor 統合のパフォーマンスをモニタリングする](#)。キャッシュパフォーマンスの低下や予期しないオリジンリクエストが発生した場合は、トラブルシューティングの手順[MediaPackage、CDN、MediaTailor の統合のトラブルシューティング](#)について「」を参照してください。

MediaPackage、CDN、MediaTailor 統合のパフォーマンスをモニタリングする

AWS Elemental MediaTailor では、AWS Elemental MediaPackage とコンテンツ配信ネットワーク (CDN) の統合の最適なパフォーマンスを維持するために、効果的なモニタリングが必要です。このトピックでは、追跡する主要なメトリクス、使用するモニタリングツール、プロアクティブな問題検出のためのアラートを設定する方法に関するガイダンスを提供します。

モニタリングを設定する前に、基本的な統合が正しく機能していることを確認してください。基本的なコンテンツ配信ネットワーク統合のセットアップを完了していない場合は、から始めます[MediaTailor を MediaPackage および CDN と統合する](#)。モニタリングによって特定された問題をトラブルシューティングする必要がある場合は、「」を参照してください[CDN 統合のトラブルシューティング](#)。

主要なパフォーマンスメトリクス

MediaPackage と CDN の統合の最適なパフォーマンスを確保するために、これらの重要なメトリクスをモニタリングします。

CDN パフォーマンスメトリクス

キャッシュヒット率ターゲット、オリジンリクエストボリュームモニタリング、応答時間ベンチマークなどの包括的な CDN パフォーマンスメトリクスについては、CDN 最適化ガイド[CDN と MediaTailor の統合のパフォーマンスベンチマーク](#)の「」を参照してください。

CDN メトリクスに関する EMP 固有の主な考慮事項:

EMP キャッシュコントロールヘッダー

検証対象: CDN が最適な TTL 動作のために EMP のキャッシュコントロールヘッダーを尊重していることを確認する

予想される動作: コンテンツタイプごとに、EMP のヘッダーに基づいて異なるキャッシュ期間を設定する必要があります

EMP キャッシュの最適化に関する詳細なガイダンスについては、「」を参照してください [MediaTailor および MediaPackage コンテンツ配信の CDN キャッシュを最適化する](#)。

クエリパラメータの影響

モニタリング対象: EMP 固有のクエリパラメータがキャッシュ効率にどのように影響するかを追跡する

最適化ターゲット: 必要な EMP クエリパラメータのみがキャッシュキーに含まれていることを確認します。

CDN 応答時間

モニタリング対象: さまざまなコンテンツタイプ (マニフェストとセグメント) の応答時間を追跡します。

ターゲット値:

- キャッシュされたコンテンツ: 100 ミリ秒未満
- オリジンリクエスト: 500 ミリ秒未満

MediaPackage パフォーマンスメトリクス

エラー率

モニタリング対象: CDN エンドポイントと MediaPackage エンドポイントの両方から HTTP エラー率をモニタリングします。設定の問題を示している可能性がある 4xx エラーには特に注意してください。

主なエラーコード:

- 400 エラー: マニフェストフィルタリングの問題に関連していることが多い
- 404 エラー: ルーティングまたは設定の問題を示している可能性があります
- 504 エラー: タイムアウトの問題、特に LL-HLS の場合

リクエストボリュームとパターン

モニタリング対象: MediaPackage エンドポイントへのリクエストパターンを追跡して、使用状況の傾向と容量のニーズを特定します。

監視するパターン:

- ピーク使用時間
- リクエストの地理的分布
- コンテンツタイプのディストリビューション (ライブとオンデマンド)

レイテンシーメトリクス

End-to-endレイテンシー

モニタリング対象: LL-HLS 実装では、コンテンツの取り込みから視聴者の再生までのend-to-endのレイテンシーをモニタリングします。レイテンシーが高い場合は、CDN 設定の問題を示している可能性があります。

ターゲット値:

- LL-HLS: 3 秒未満のglass-to-glassレイテンシー
- 通常の HLS: 30 秒未満

マニフェスト生成時間

モニタリング対象: MediaPackage がマニフェストを生成するのにかかる時間。特にフィルタリングを適用した場合。

ターゲット値: マニフェスト生成で 200 ミリ秒未満

モニタリングツールとセットアップ

包括的なモニタリングツールを設定することは、最適なパフォーマンスを維持し、ビューワーに影響を与える前に問題をすばやく特定するために不可欠です。適切なモニタリングを行わないと、視聴者の再生品質が低下するまで、パフォーマンスの低下、キャッシュの非効率性、統合の問題に気付かないことがあります。適切なモニタリング設定により、MediaPackage と CDN の統合のあらゆる側面を可視化できます。

MediaPackage と CDN の統合をモニタリングするには、以下の AWS のサービスとツールを使用します。

Amazon CloudWatch

Amazon CloudWatch は、両方のサービスからメトリクスを収集して保存することで、MediaPackage と CDN の統合をモニタリングするための基盤を提供します。適切な CloudWatch 設定により、パフォーマンスの傾向の特定、問題のトラブルシューティング、統合の最

適化に必要なデータを確保できます。CloudWatch メトリクスがないと、システムパフォーマンスの可視性が低下し、問題が重大になるまで検出されない可能性があります。

包括的なメトリクス収集のために CloudWatch モニタリングを設定します。

1. MediaPackage メトリクス: MediaPackage エンドポイントの CloudWatch メトリクスを有効にして、リクエストボリューム、エラー率、応答時間を追跡します。
2. CDN メトリクス: キャッシュヒット率、オリジンリクエスト数、エラー率などの CloudFront メトリクスを収集するように CloudWatch を設定します。
3. カスタムメトリクス: ビューワーのエンゲージメントやコンテンツの人気など、ビジネス固有の KPIs のカスタムメトリクスを作成します。

CloudWatch ダッシュボード

メトリクスを視覚化する包括的なダッシュボードを作成します。

1. 概要ダッシュボード: 全体的なシステムの状態を示す高レベルのメトリクス
2. CDN パフォーマンスダッシュボード: キャッシュパフォーマンスと地理的分散を含む詳細な CDN メトリクス
3. MediaPackage パフォーマンスダッシュボード: リクエストパターンやエラー率を含む MediaPackage 固有のメトリクス
4. レイテンシーダッシュボード: End-to-end さまざまなコンテンツタイプとリージョンのエンドツーエンドのレイテンシーメトリクス

ログ分析

詳細なトラブルシューティングのためにログ分析を設定します。

1. CDN アクセスログ: CDN アクセスログを有効にして分析し、リクエストパターンを理解し、問題を特定する
2. MediaPackage CloudWatch ログ: MediaPackage ログのエラーとパフォーマンスの問題を監視する
3. ログ集約: Amazon CloudWatch Logs Insights またはサードパーティーツールを使用してログパターンを分析する

アラートと通知を設定する

アラート設定は、プロアクティブな問題の検出と解決に不可欠です。適切なアラートがないと、問題がビューワーエクスペリエンスに大きな影響を与えたり、サービスの中断を引き起こすまで、問題が気付かなくなる可能性があります。適切に設定されたアラートは、ビューワーに影響を与える前に問題を特定して対処し、すぐに対処する必要がある重大な問題がチームに通知されるようにするのに役立ちます。

プロアクティブアラートを設定して、ビューワーに影響を与える前に問題を特定します。

重要なアラート

重大な問題に対する即時アラートを設定します。

- 高エラー率: 4xx または 5xx エラー率が 5 分間で 5% を超えた場合に警告
- キャッシュヒット率の低下: キャッシュヒット率がマニフェストの場合は 70%、セグメントの場合は 85% を下回ると警告
- 高レイテンシー: end-to-endのレイテンシーがターゲットしきい値を超えた場合に警告する
- オリジンリクエストのスパイク: オリジンリクエストがベースラインと比較して 50% 以上増加した場合のアラート

警告アラート

問題の発生を示す傾向の警告アラートを設定します。

- パフォーマンスの段階的な低下: 応答時間が 30 分間で 20% 増加した場合のアラート
- キャッシュ効率の傾向: キャッシュヒット率が時間の経過とともに減少傾向を示す場合に警告する
- 異常なトラフィックパターン: リクエスト量または地理的分布の予期しない変更を警告する

モニタリングデータを使用して最適化する

モニタリングデータを活用してパフォーマンスを継続的に向上させます。

定期的なパフォーマンスレビュー

1. 毎週のレビュー: 毎週のパフォーマンス傾向を分析し、最適化の機会を特定する
2. 毎月のキャパシティプランニング: トラフィックパターンを使用してキャパシティのニーズと CDN 最適化を計画する

3. 四半期ごとのアーキテクチャレビュー: 全体的なアーキテクチャ効率を評価し、改善すべき分野を特定する

一般的な最適化アクション

モニタリングデータに基づいて、次の最適化アクションを検討してください。

- キャッシュポリシーの調整: 実際のコンテンツ更新パターンに基づいて TTL 値を変更します。TTL 最適化の詳細なガイダンスについては、「」を参照してください[CDN と MediaTailor の統合のキャッシュ最適化](#)。
- 地理的最適化: トラフィックの多いリージョンに CDN エッジロケーションを追加する
- クエリパラメータの最適化: キャッシュをフラグメント化する不要なクエリパラメータを削除する
- オリジンシールド設定: オリジンリクエストボリュームが多いリージョンにオリジンシールドを実装する

MediaPackage に固有の詳細なモニタリングガイダンスについては、[MediaPackage ユーザーガイドの「MediaPackage のモニタリング」](#)を参照してください。MediaPackage

MediaPackage、CDN、MediaTailor の統合のトラブルシューティング

AWS Elemental MediaTailor AWS Elemental MediaPackage およびコンテンツ配信ネットワーク (CDN) との統合では、再生、キャッシュ、またはその他の統合機能に影響する一般的な問題が発生する可能性があります。このガイドは、再生の問題、キャッシュの問題、またはその他の統合関連のエラーが発生した場合に使用します。

ユニバーサルキャッシュパフォーマンスの問題、HTTP エラー解決、テスト手順、すべての MediaTailor 実装に適用される診断手法など、包括的な CDN トラブルシューティングガイダンスについては、「」を参照してください[CDN 統合のトラブルシューティング](#)。このセクションでは、MediaPackage 固有のトラブルシューティング要件に焦点を当てます。

トラブルシューティングを行う前に、基本的な統合設定が正しく完了していることを確認してください。統合をまだセットアップしていない場合、またはセットアップ手順を確認する必要がある場合は、「」を参照してください[MediaTailor を MediaPackage および CDN と統合する](#)。問題を解決した後のキャッシュパフォーマンスの最適化に関するガイダンスについては、「」を参照してください[CDN キャッシュ](#)。

マニフェストフィルタリングエラー

文書化されたエラー条件に基づく MediaPackage マニフェストフィルタリング機能の問題:

マニフェストフィルタリングによる HTTP 400 エラー

症状: `aws.manifestfilter` パラメータを含むリクエストは HTTP 400 Bad Request を返します

検証済みの原因 (ドキュメントから AWS):

- フィルター条件の結果が空のマニフェストになる (フィルター条件に一致するストリームがない)
- 無効なフィルターパラメータ名または値
- 不正な形式のクエリ文字列構文
- 重複または繰り返されるフィルターパラメータ
- フィルターパラメータ文字列が 1024 文字を超えています
- メディアプレイリストまたはセグメントに適用されるクエリパラメータ (サポートされていません)

解決方法:

1. フィルターパラメータを確認して、使用可能なコンテンツストリームと一致することを確認します。フィルタリングの結果に一致するストリームがない場合、MediaPackage は HTTP 400 を返します。
2. サポートされているパラメータ名と値形式に対してフィルター構文を検証します。
3. クエリ文字列に重複するパラメータがないか確認します。
4. フィルターパラメータは、メディアプレイリストやセグメントではなく、多変量プレイリストにのみ適用されます。
5. 合計パラメータ文字列が 1024 文字未満であることを確認します。

リファレンス: [AWS Elemental MediaPackage マニフェストフィルタリングエラー条件](#)

マニフェストフィルタリングが機能しない (HTTP 200 ではあるが、フィルタリングは適用されない)

症状: リクエストは HTTP 200 を返しますが、マニフェストにはフィルタリングされたサブセットではなくすべてのストリームが含まれます

考えられる原因:

- CDN が `aws.manifestfilter` クエリパラメータを MediaPackage に転送しない

- 使用可能なストリームにフィルターパラメータが見つかりません (HTTP 200 でフィルタリングされていないマニフェストを返します)

解決方法:

1. CDN キャッシュポリシーが転送されたクエリ文字列のリスト `aws.manifestfilter` に含まれていることを確認します。
2. MediaPackage エンドポイントに対してフィルターパラメータを直接テストし (CDN を超過)、期待どおりに動作することを確認します。
3. フィルター値がコンテンツストリームの実際の特性と一致することを確認します。

リファレンス: [AWS Elemental MediaPackage マニフェストフィルタリングエラー条件](#)

診断手順

体系的な診断手順は、統合問題の根本原因を迅速かつ効率的に特定するのに役立ちます。構造化されたアプローチに従うことで、誤った仮定に浪費される時間を防ぎ、症状ではなく実際の問題に対処できます。これらの証拠ベースの診断手順は、問題を分離し、適切なソリューションに導くように設計されています。

以下の証拠ベースの診断手順に従って問題を特定します。

キャッシュのパフォーマンスの分析

EMP 統合ではキャッシュパフォーマンス分析が不可欠です。キャッシュ効率が低いと、オリジンの負荷が増加し、コストが増加し、再生の問題が発生する可能性があるためです。

キャッシュヒット率分析、キャッシュキーの最適化、体系的な診断手順を含む包括的なキャッシュパフォーマンスのトラブルシューティングについては、メイン CDN トラブルシューティングガイド [CDN キャッシュのパフォーマンスの問題](#) の「」を参照してください。

EMP 固有のキャッシュに関する考慮事項:

- EMP キャッシュコントロールヘッダー: CDN が EMP のキャッシュコントロールヘッダーを上書きするのではなく尊重していることを確認します。
- EMP クエリパラメータ: 必要な EMP クエリパラメータのみがキャッシュキーに含まれていることを確認します
- EMP TTL の動作: さまざまな EMP コンテンツタイプに適切なキャッシュ期間があることを確認する

EMP キャッシュポリシーと TTL 設定の最適化に関する詳細なガイダンスについては、「」を参照してください[MediaTailor および MediaPackage コンテンツ配信の CDN キャッシュを最適化する](#)。

マニフェストフィルタリング設定を検証する

マニフェストフィルタリングの検証は不可欠です。フィルタリングの問題により、ビューワーが誤ったコンテンツ、サポートされていない形式、またはアクセスすべきでないコンテンツを受信する可能性があります。体系的なテストは、問題が CDN 設定、フィルターパラメータ構文、またはコンテンツの可用性に関連しているかどうかを特定するのに役立ちます。

マニフェストフィルタリング機能を体系的にテストします。

1. MediaPackage エンドポイントに対して (CDN をバイパスして) フィルターパラメータを直接テストし、正しく動作することを確認します。
2. フィルタリングされたマニフェストとフィルタリングされていないマニフェストを比較して、予想されるストリームが含まれているか除外されているかを確認します。
3. CDN キャッシュポリシーがaws.manifestfilterクエリパラメータを転送することを確認します。
4. HTTP 400 エラーをチェックし、文書化されたエラー条件と照合します。

問題を解決した後にマニフェストフィルタリングを実装または変更する必要がある場合は、セットアップガイダンスの詳細については、[MediaTailor、MediaPackage、CDN でマニフェストフィルタリングを設定する](#)「」を参照してください。

クエリパラメータ設定を検証する

CDN が必要なクエリパラメータのみを転送していることを確認します。

1. CDN キャッシュポリシーを確認して、AWS 推奨パラメータのみが含まれていることを確認します。
 - aws.manifestfilter - マニフェストフィルタリング用
 - aws.manifestsettings - タイムシフト表示用
 - _HLS_msn LL-HLS サポート用の _HLS_part および -
2. MediaPackage は他のクエリパラメータを無視し、キャッシュ効率を低下させるため、キャッシュキーから他のクエリパラメータを削除します。

リファレンス: [AWS Elemental MediaPackage および CDN の使用](#)

エラーコードのリファレンス

文書化されたエラー条件とその原因に関するリファレンス:

HTTP 400 不正なリクエスト (マニフェストフィルタリング)

文書化された原因:

- フィルターを適用すると、空のマニフェストになります。
- 無効なパラメータ名または値
- 不正な形式のクエリ文字列構文
- フィルターパラメータの複製
- パラメータ文字列が 1024 文字を超えています
- メディアプレイリストまたはセグメントのクエリパラメータ

リファレンス: [MediaPackage マニフェストフィルタリングエラー条件](#)

HTTP 200 OK (フィルタリングは適用されません)

文書化された原因:

- 使用可能なストリームにフィルターパラメータが見つかりません (フィルタリングされていないマニフェストを返します)
- フィルタリング後に存在するサブタイトルストリームのみ (フィルタリングされていないマニフェストを返します)

リファレンス: [MediaPackage マニフェストフィルタリングエラー条件](#)

その他のトラブルシューティングリソース

このトピックで説明されていない問題については、以下の公式 AWS リソースを参照してください。

- [からマニフェストをプレビューする AWS Elemental MediaPackage](#) - マニフェストプレビューを使用してコンテンツパッケージングの問題をトラブルシューティングする
- [CloudFront キャッシュヒット率の向上](#) - CDN キャッシュ最適化の包括的なガイド
- [マニフェストフィルタリング](#) - MediaPackage フィルタリング機能の完全なガイド

Amazon CloudFront AWS Elemental MediaTailor との統合

AWS Elemental MediaTailor は Amazon CloudFront と統合して、コンテンツ配信のパフォーマンスと信頼性を向上させます。CloudFront は、エッジロケーションと呼ばれるデータセンターの世界的ネットワークを通じてコンテンツを配信するコンテンツ配信ネットワーク (CDN) です。ビューワーが MediaTailor にコンテンツをリクエストすると、CloudFront はリクエストを最も近いエッジロケーションにルーティングします。このアプローチにより、レイテンシーが短縮され、ビューワーのパフォーマンスが向上します。

MediaTailor と CloudFront の統合には、いくつかの主な利点があります。

- パーソナライズされたコンテンツにアクセスするビューワーのレイテンシーの短縮
- 大規模なオーディエンスサイズを処理するためのスケーラビリティの向上
- 冗長な配信パスによる信頼性の向上
- 効率的なキャッシュ戦略によるコスト最適化
- Media Quality-Aware Resiliency (MQAR) によるマルチリージョンフェイルオーバーなどの高度な機能

CloudFront の機能に関する包括的な情報については、[CloudFront デベロッパーガイド](#) を参照してください。CloudFront の料金については、[CloudFront の料金](#) を参照してください。

簡潔にするために、「マニフェスト」を使用して、マルチバリエーションプレイリスト、メディアプレイリスト、MPDs。

トピック

- [MediaTailor との基本的な CloudFront 統合を設定する](#)
- [CloudFront 機能を使用した MediaTailor のパフォーマンスの最適化](#)
- [MQAR を使用して MediaTailor のマルチリージョンレジリエンスを実装する](#)
- [CloudFront と MediaTailor の統合のモニタリングとトラブルシューティング](#)

MediaTailor との基本的な CloudFront 統合を設定する

AWS Elemental MediaTailor Amazon CloudFront との統合により、ビューワーのコンテンツ配信パフォーマンスが向上します。このトピックでは、MediaTailor の基本的な CloudFront ディストリビューションを設定する方法について説明します。この統合により、ビューワーは CloudFront ネットワークを通じてコンテンツにアクセスします。

トワークを介してパーソナライズされたコンテンツにアクセスできます。また、さまざまなコンテンツタイプに適切なキャッシュを設定する方法についても説明します。

認可とルーティングのために CloudFront を介してクエリパラメータを渡す方法については、「」を参照してください[MediaTailor マニフェストクエリパラメータ](#)。動的変数を使用した高度なルーティングについては、「」を参照してください[複数のコンテンツソースの MediaTailor ドメイン変数](#)。

前提条件

MediaTailor で CloudFront を設定する前に、以下があることを確認してください。

- CloudFront デイストリビューションを作成および管理するためのアクセス許可を持つアクティブな AWS アカウント
- 設定された MediaTailor 再生設定 (「」を参照[AWS Elemental MediaTailor を使用して広告を挿入する](#))
- コンテンツオリジンサーバーが適切にセットアップされ、アクセス可能である
- ビデオストリーミングの概念に関する基本的な理解 (HLS/DASH)

CloudFront デイストリビューションの設定

MediaTailor の CloudFront デイストリビューションを作成および設定するには、次の手順に従います。

MediaTailor の CloudFront デイストリビューションを作成するには

1. AWS マネジメントコンソールにサインインし、<https://console.aws.amazon.com/cloudfront/v3/home> で CloudFront コンソールを開きます。
2. [デイストリビューションを作成] を選択します。
3. オリジンドメインには、MediaTailor エンドポイント URL (例:) を入力します `a1b2c3d4.mediataylor.us-west-2.amazonaws.com`。
4. Protocol では、HTTPS のみを選択します。
5. Name には、このオリジンを識別するのに役立つ名前を入力します (例: `mediatailor-origin`)。
6. デフォルトのキャッシュ動作設定を構成します。
 - a. パスパターンには、デフォルト値 () を使用します*。

- b. オブジェクトを自動的に圧縮するには、「はい」を選択します。
- c. ビューワープロトコルポリシーで、HTTP を HTTPS にリダイレクトを選択します。
- d. 許可された HTTP メソッドでは、GET、HEAD を選択します。
- e. キャッシュポリシーで、CachingDisabled を選択します。
- f. オリジンリクエストポリシーで、AllViewer を選択して、デフォルトの動作のすべてのヘッダーを転送します。

 Note

デフォルトの動作では、AllViewer を使用して、特定のパスパターンに一致しないコンテンツを安全に処理します。マニフェストとセグメントの特定のキャッシュ動作は、適切なポリシーで個別に設定されます。

- 7. ディストリビューション設定を構成します。
 - a. Price クラスで、対象者のロケーションに最も適したオプションを選択します。
 - b. AWS WAF ウェブ ACL の場合は、既存のウェブ ACL を選択するか、セキュリティ保護を有効にしないのままにします。
 - c. デフォルトルートオブジェクトの場合は、空のままにします。
 - d. 標準ログ記録では、オンを選択してログ記録を有効にします。
- 8. [ディストリビューションを作成] を選択します。

キャッシュ動作の設定

ディストリビューションを作成したら、さまざまなタイプのコンテンツを適切に処理するように追加のキャッシュ動作を設定する必要があります。このセクションでは、CloudFront の基本的なキャッシュ動作の設定について説明します。

高度な TTL 設定、キャッシュキー設定、パフォーマンスチューニングなどの包括的なキャッシュ最適化については、CDN 最適化ガイド[CDN と MediaTailor の統合のキャッシュ最適化](#)の「」を参照してください。

マニフェストキャッシュ動作の設定

マニフェストにはパーソナライズされたコンテンツが含まれているため、キャッシュしないでください。キャッシュ動作を設定するには、次の手順に従います。

マニフェストキャッシュの動作を設定するには

1. CloudFront コンソールで、ディストリビューションを選択します。
2. [Behaviors] タブを選択します。
3. [Create behavior] (動作の作成) を選択します。
4. パスパターンには、HLS マルチバリエーションとメディアプレイリスト*.m3u8と一致するようにと入力します。
5. オリジンで、MediaTailor オリジンを選択します。
6. キャッシュポリシーで、CachingDisabled を選択します。
7. オリジンリクエストポリシーで、AllViewer を選択して、動的コンテンツに必要なすべてのヘッダーを転送します。
8. [作成] を選択します。
9. MPDs に一致するパスパターンを使用して*.mpd、DASH マニフェストに対してこれらのステップを繰り返します。

この設定により、各ビューワーは特定の広告コンテンツを含むパーソナライズされたマニフェストを受信できます。CDN はこれらのマニフェストをキャッシュしないため、各リクエストは MediaTailor に直接送信されます。

セグメントキャッシュ動作の設定

広告セグメントとコンテンツセグメントに個別のキャッシュ動作を設定して、パフォーマンスを最適化し、CORS の適切な処理を確保します。

広告セグメントのキャッシュ動作の設定

/tm/* パスパターンを通じて提供される広告セグメントには、CORS を適切に処理するための特定の設定が必要です。以下の手順に従ってください。

広告セグメントのキャッシュ動作を設定するには

1. CloudFront コンソールで、ディストリビューションを選択します。
2. [Behaviors] タブを選択します。
3. [Create behavior] (動作の作成) を選択します。
4. パスパターンには、MediaTailor が提供する広告セグメントと一致する/tm/*ようにと入力します。

5. オリジンで、MediaTailor セグメントのオリジン (segments.mediatailor.region.amazonaws.com ホスト名を使用) を選択します。
6. キャッシュポリシーで、CachingOptimized を選択します。
7. オリジンリクエストポリシーで、なしを選択します。
8. レスponseヘッダーポリシーで、CORS-with-preflight-and-SecurityHeadersPolicy を選択して、適切な CORS ヘッダーがレスポンスに含まれていることを確認します。
9. [作成] を選択します。

コンテンツセグメントのキャッシュ動作の設定

コンテンツセグメントは、パフォーマンスを最適化するために標準のキャッシュポリシーを使用できます。セグメント形式ごとに個別の動作を設定します。

コンテンツセグメントのキャッシュ動作を設定するには

1. CloudFront コンソールで、ディストリビューションを選択します。
2. [Behaviors] タブを選択します。
3. [Create behavior] (動作の作成) を選択します。
4. パスパターンには、HLS コンテンツセグメント*.tsと一致するように と入力します。
5. オリジン で、コンテンツオリジンを選択します。
6. キャッシュポリシーで、CachingOptimized を選択します。
7. オリジンリクエストポリシーで、なしを選択します。
8. レスponseヘッダーポリシーで、CORS-with-preflight-and-SecurityHeadersPolicy を選択して、すべてのコンテンツタイプで一貫した CORS 処理を確保します。
9. [作成] を選択します。
10. 適切なパスパターンを使用して、他のコンテンツセグメント形式に対してこれらのステップを繰り返します。
 - *.mp4 MP4 セグメントの
 - *.m4s DASH セグメントの
 - *.cmfv CMAF セグメント*.cmfaの および

この設定により、広告セグメントとコンテンツセグメントが適切な CORS 処理で適切にキャッシュされます。広告セグメントは CORS 保護で MediaTailor セグメントオリジンを使用し、コンテンツセグメントは最適化されたキャッシュポリシーでコンテンツオリジンを使用します。

MediaTailor 設定の更新

CloudFront ディストリビューションを設定したら、CloudFront ドメインを使用するように MediaTailor 設定を更新します。

MediaTailor 設定を更新するには

1. [MediaTailor コンソール](#)を開きます。
2. 更新する設定を選択します。
3. CDN 設定セクションで、CDN コンテンツセグメントプレフィックスフィールドに CloudFront ディストリビューションドメイン名 (など `d1234abcdef.cloudfront.net`) を入力します。
4. 変更内容を保存します。

この設定では、MediaTailor はオリジンに直接ではなく CloudFront ディストリビューションを指す URLs を持つマニフェストを生成します。

統合のテスト

CloudFront ディストリビューションを設定し、MediaTailor 設定を更新したら、統合をテストします。

CloudFront と MediaTailor の統合をテストするには

1. CloudFront ディストリビューション (など `https://d1234abcdef.cloudfront.net/v1/master/12345/my-config/index.m3u8`) を使用してマニフェストをリクエストします。
2. マニフェストに CloudFront ドメインを指すセグメント URLsが含まれていることを確認します。
3. 動画プレーヤーを介してコンテンツを再生し、コンテンツと広告の両方が正しく再生されていることを確認します。
4. CloudFront ログをチェックして、リクエストが正しくルーティングされていることを確認します。

設定例

以下は、適切なキャッシュ動作を持つ MediaTailor の CloudFront デистриビューション設定の例です。

Example CloudFront デистриビューション設定の例

```
{
  "DefaultCacheBehavior": {
    "TargetOriginId": "mediatailor-origin",
    "ViewerProtocolPolicy": "redirect-to-https",
    "AllowedMethods": {
      "Quantity": 2,
      "Items": ["GET", "HEAD"]
    },
    "CachePolicyId": "4135ea2d-6df8-44a3-9df3-4b5a84be39ad",
    "OriginRequestPolicyId": "59781a5b-3903-41f3-afcb-af62929ccde1",
    "Comment": "Default behavior with CachingDisabled and AllViewer"
  },
  "CacheBehaviors": [
    {
      "PathPattern": "/*.m3u8",
      "TargetOriginId": "mediatailor-origin",
      "ViewerProtocolPolicy": "redirect-to-https",
      "CachePolicyId": "4135ea2d-6df8-44a3-9df3-4b5a84be39ad",
      "OriginRequestPolicyId": "59781a5b-3903-41f3-afcb-af62929ccde1",
      "Comment": "Manifest behavior with CachingDisabled and AllViewer"
    },
    {
      "PathPattern": "/*.ts",
      "TargetOriginId": "mediatailor-origin",
      "ViewerProtocolPolicy": "redirect-to-https",
      "CachePolicyId": "658327ea-f89d-4fab-a63d-7e88639e58f6",
      "OriginRequestPolicyId": "88a5eaf4-2fd4-4709-b370-b4c650ea3fcf",
      "Comment": "Segment behavior with CachingOptimized and HostHeaderOnly"
    }
  ]
}
```

この例は、以下を示しています。

- デフォルトの動作: CachingDisabledと AllViewerを使用して、特定のパスパターンに一致しないコンテンツを安全に処理します。
- マニフェスト動作 (*.m3u8): 動的コンテンツAllViewerに CachingDisabledと を使用します
- セグメント動作 (*.ts): CachingOptimizedと を使用します CORS-with-preflight-and-SecurityHeadersPolicy

次の手順

MediaTailor との基本的な CloudFront 統合を設定したら、次のステップを検討してください。

- 追加の CloudFront 機能でパフォーマンスを最適化する (「」を参照[CloudFront 機能を使用した MediaTailor のパフォーマンスの最適化](#))
- MQAR を使用してマルチリージョンレジリエンスを実装する (「」を参照[MQAR を使用して MediaTailor のマルチリージョンレジリエンスを実装する](#))
- モニタリングとトラブルシューティングを設定する (「」を参照[CloudFront と MediaTailor の統合のモニタリングとトラブルシューティング](#))

CloudFront 機能を使用した MediaTailor のパフォーマンスの最適化

AWS Elemental MediaTailor Amazon CloudFront のパフォーマンスは、基本設定以外の追加機能によって強化できます。MediaTailor で基本的な CloudFront 設定をセットアップしたら、パフォーマンス、信頼性、カスタマイズオプションを強化するための追加機能を実装できます。これらの最適化により、表示エクスペリエンスが向上します。また、コストとオリジン負荷も削減されます。

動的ルーティングとパラメータ処理を使用したパフォーマンスの最適化については、「」を参照してください[ADS リクエストの MediaTailor 動的広告変数](#)。クエリパラメータの最適化戦略については、「」を参照してください[MediaTailor マニフェストクエリパラメータ](#)。

Origin Shield によるオリジン負荷の低減

Origin Shield は、CloudFront エッジロケーションとオリジンサーバーの間にキャッシュレイヤーを追加します。この機能は、ライブストリーミングや一般的な VOD コンテンツに役立ちます。これは、多くの視聴者が同時に同じコンテンツをリクエストする場合に役立ちます。

複数のエッジロケーションからのリクエストを統合することで、Origin Shield は MediaTailor とコンテンツオリジンの負荷を軽減します。

MediaTailor オリジンの Origin Shield を有効にするには

1. CloudFront コンソールを開き、ディストリビューションに移動します。
2. MediaTailor 再生設定を指すオリジンを選択します。
3. Origin Shield で、はいを選択します。
4. ドロップダウンメニューから、MediaTailor エンドポイントに最も近い AWS リージョンを選択します。
5. 変更内容を保存します。

トラフィックの多いイベントの場合、Origin Shield はオリジンへのリクエスト負荷を大幅に削減し、信頼性を向上させます。詳細な手順については、CloudFront デベロッパーガイドの「[Origin Shield の有効化](#)」を参照してください。

CloudFront Functions を使用したコンテンツ配信のカスタマイズ

CloudFront Functions を使用すると、エッジで軽量の JavaScript コードを実行して、ビューワーのリクエストとレスポンスを変更できます。これらの関数は、URL の変更、ヘッダー操作、基本認証などの簡単なカスタマイズに使用できます。MediaTailor ワークフローの場合、関数は複雑な処理を必要としないタスクに役立ちます。

CloudFront Functions は、エッジでのコンテンツ配信を軽量にカスタマイズする方法を提供します。MediaTailor 統合用にこれらを実装する方法は次のとおりです。

MediaTailor の CloudFront Functions を実装するには

1. CloudFront コンソールで、関数に移動します。
2. 新しい関数を作成し、適切な目的を選択します。
 - URL 操作 - MediaTailor に到達する前に、多変量プレイリスト、メディアプレイリスト、MPD リクエスト URLs を変更するには
 - ヘッダー操作 - リクエストヘッダーを追加または変更するには
 - 簡易認証 - トークンまたはクエリパラメータを検証するには
3. JavaScript 関数コードを記述します。
4. サンプル MediaTailor リクエストで関数をテストします。
5. 関数を発行し、ディストリビューションのキャッシュ動作に関連付けます。

Example URL 正規化のサンプル CloudFront 関数

```
function handler(event) {
    var request = event.request;
    var uri = request.uri;

    // Normalize URLs to lowercase to improve cache hit ratio
    if (uri.includes('.m3u8') || uri.includes('.mpd')) {
        request.uri = uri.toLowerCase();
    }

    return request;
}
```

詳細とコード例については、[CloudFront デベロッパーガイドの「CloudFront Functions を使用してエッジでカスタマイズする」](#)を参照してください。 CloudFront

Lambda@Edge を使用した高度なカスタマイズの実装

CloudFront Functions が提供するよりも複雑な処理機能が必要な場合は、Lambda@Edge を使用します。このサービスでは、CloudFront エッジロケーションで Node.js または Python 関数を実行できます。

Lambda@Edge 関数は、複雑な認証、大規模なレスポンス変更、サードパーティー API 統合などの高度なオペレーションを実行できます。

より複雑なカスタマイズを行うには、MediaTailor と CloudFront の統合で Lambda@Edge 関数を使用します。

MediaTailor で Lambda@Edge を実装するには

1. 米国東部 (バージニア北部) リージョンで Lambda 関数を作成します。
2. 次のいずれかのユースケースの関数コードを記述します。
 - URL 操作 - MediaTailor に到達する前に、多変量プレイリスト、メディアプレイリスト、MPD リクエスト URLs を変更するには
 - A/B テスト - ユーザーを異なる広告決定サーバーにルーティングするには
 - 認証のリクエスト - 認証ヘッダーを追加するには

- レスポンスヘッダーの変更 - CORS ヘッダーを追加するには
3. 関数のバージョンを発行し、関数エイリアスを作成します。
 4. 関数を適切なトリガーポイント (ビューワーリクエストまたはビューワーレスポンス) で CloudFront ディストリビューションに関連付けます。

Note

MediaTailor で Lambda@Edge を使用する場合は、Media Quality-Aware Resiliency (MQAR) 機能を使用する予定がある場合は、オリジン向けトリガー (オリジンリクエストとオリジンレスポンス) を使用しないでください。これらは現在互換性がないためです。

詳細とコード例については、CloudFront デベロッパーガイドの「[Lambda@Edge を使用してエッジでカスタマイズ](#)する」を参照してください。

パフォーマンス最適化に関するその他のヒント

パフォーマンスをさらに向上させるには、以下の追加の最適化を検討してください。

キャッシュヒット率の最適化

CloudFront メトリクスでキャッシュヒット率をモニタリングし、改善の機会を探します。

- URL パターンを標準化してキャッシュキーの一貫性を向上させる
- クエリ文字列のホワイトリストを使用して、キャッシュキーに必要なパラメータのみを含める
- CloudFront Functions を使用した URL 正規化の実装を検討する

レイテンシーの短縮

レイテンシーを最小限に抑えるには、次の手法を実装します。

- テキストベースのレスポンスの Brotli 圧縮を有効にする
- HTTP/2 または HTTP/3 を使用して接続効率を向上させる
- 最新のネットワークで IPv6 サポートを有効にすることを検討してください

コスト最適化

パフォーマンスとコスト効率のバランスを取ります。

- Origin Shield を使用して冗長なオリジンリクエストを減らす
- 対象者の地域に基づいて価格クラスを選択を検討する

- セグメントに積極的なキャッシュを実装してオリジントラフィックを削減する

次の手順

CloudFront 機能でパフォーマンスを最適化したら、次のステップを検討してください。

- MQAR を使用してマルチリージョンレジリエンスを実装する (「」を参照[MQAR を使用して MediaTailor のマルチリージョンレジリエンスを実装する](#))
- モニタリングとトラブルシューティングを設定する (「」を参照[CloudFront と MediaTailor の統合のモニタリングとトラブルシューティング](#))

MQAR を使用して MediaTailor のマルチリージョンレジリエンスを実装する

AWS Elemental MediaTailor マルチリージョンレジリエンスは、MediaTailor が可能な限り最高のストリーミングエクスペリエンスを提供するのに役立つ高度な Amazon CloudFront 機能である Media Quality-Aware Resiliency (MQAR) によって強化されています。MediaTailor 異なる AWS リージョンに複数のオリジンがある場合、最高品質のスコアを持つオリジンが自動的に選択されます。この機能は、中断のないサービスが必要なライブストリーミングに特に役立ちます。

MQAR の仕組み

MQAR を使用すると、CloudFront は最高品質のスコアを持つオリジンを自動的に選択できます。これにより、ビューワーは可能な限り最高のストリーミングエクスペリエンスを得ることができます。適切に設定すると、MQAR には次のような利点があります。

- 最高品質のオリジンの自動選択
- 停止中の AWS リージョン間のシームレスなフェイルオーバー
- 中断を最小限に抑えてビューワーエクスペリエンスを改善
- リアルタイムの品質モニタリングと適応

ステップ 1: MQAR 要件を検証する

MQAR を実装する前に、インフラストラクチャがこれらの要件を満たしていることを確認します。MQAR は複数のオリジンの品質スコアを比較することで機能するため、複数の AWS リージョンで利用可能な同一のコンテンツが必要です。

- エポックロックされた CMAF 取り込みストリームセットを使用して、調整された取り込みストリームをすべての MediaPackage チャンネルに送信するエンコーダー
- 異なる AWS リージョンの 2 つの同一の MediaPackage チャンネルと同一のオリジンエンドポイント
- MediaPackage チャンネルの CMAF 取り込み (デフォルトで有効)
- MQAR をサポートするように設定された CloudFront デистриビューション
- 各 MediaPackage エンドポイントの MediaTailor 設定 MediaPackage

ステップ 2: MQAR のエンコーダーを設定する

MQAR を効果的に機能させるには、エンコーダーがすべてのリージョンで一貫した同期された出力を生成する必要があります。この一貫性により、CloudFront はオリジン間で正確な品質比較を行うことができます。

以下の設定で MediaLive 出力を設定します。

- CMAF 出力グループ内のすべてのビデオフレームレートが一貫していることを確認します (すべての小数フレームレートまたはすべての整数フレームレート)。
- エンコーディングセッション間での小数フレームレートと整数フレームレート間の遷移を回避します。
- エンコードセッション間で後退しないように、出力セグメントのシーケンス番号を設定します。
- すべてのリージョンで同一のエンコーダー出力名を使用します。

MediaLive for MQAR の設定の詳細については、MediaLive ユーザーガイドの [「MQCS の使用」](#) を参照してください。

ステップ 3: MQAR の MediaPackage を設定する

以下の設定で MediaPackage チャンネルとエンドポイントを設定します。

MQAR の MediaPackage を設定するには

1. 各 AWS リージョンに同一のチャンネルとエンドポイントの設定を作成します。
2. CMAF をチャンネル入力タイプとして使用します。
3. プライマリ MediaPackage オリジンで、次の設定で Force エンドポイントエラー設定を有効にします。

- 古い多変量プレイリスト、メディアプレイリスト、または MPDs
 - 不完全な多変量プレイリスト、メディアプレイリスト、または MPD
 - スレート入力
4. バックアップ MediaPackage オリジンの場合、フェイルオーバーが成功する可能性を最大化するために、これらのエラー設定を有効にしないでください。

MediaPackage for MQAR の設定の詳細については、MediaPackage ユーザーガイドの「[を使用したメディア品質スコアの活用 AWS Elemental MediaPackage](#)」を参照してください。MediaPackage

ステップ 4: MQAR 用に CloudFront を設定する

CloudFront 設定では、MQAR を有効にし、オリジン間での選択方法を定義します。メディア品質スコアオプションを有効にしてオリジングループを作成します。

MQAR 用に CloudFront を設定するには

1. MediaTailor エンドポイントを指すオリジンを使用して CloudFront ディストリビューションを作成します。
2. これらのオリジンを含むオリジングループを作成します。
3. オリジングループ設定で、メディア品質スコアオプションを有効にします。
4. 404 Not Found レスポンスコードを含めるようにフェイルオーバー条件を設定します。オプションで、他の 4xx/5xx コードを含めることができます。
5. チャンネルのパスパターンごとに個別のキャッシュ動作を作成します。これにより、同じオリジングループが複数のチャンネルを処理するときにスコアが混在するのを防ぐことができます。

Note

MQAR は、ディストリビューションのキャッシュ動作に関連付けられたオリジン向けトリガー (オリジンリクエストとオリジンレスポンス) で Lambda@Edge 関数を使用する場合は使用できません。

MQAR 用に CloudFront を設定する方法の詳細については、CloudFront デベロッパーガイドの「[メディア品質対応の耐障害性](#)」を参照してください。

ステップ 5: MQAR の MediaTailor を設定する

MQAR のセットアップを完了するには、マルチリージョンアーキテクチャで動作するように各リージョンで MediaTailor を設定します。これにより、CloudFront が選択するオリジンに関係なく、一貫した広告挿入が保証されます。

MediaTailor for MQAR を設定するには

1. 対応する MediaPackage エンドポイントを参照しながら、各リージョンに同一の MediaTailor 設定を作成します。MediaPackage
2. CloudFront ディストリビューションドメインを使用するように CDN コンテンツセグメントプレフィックスを設定します。
3. 広告決定サーバーの設定がリージョン間で同一であることを確認します。

この設定により、CloudFront が品質スコアに基づいて選択するオリジンに関係なく、MediaTailor は広告を一貫してパーソナライズし続けることができます。

ステップ 6: MQAR 設定をテストする

MQAR を設定したら、設定をテストして期待どおりに動作することを確認します。

MQAR 設定をテストするには

1. CloudFront ディストリビューションを通じてコンテンツをリクエストします。
2. リアルタイムログをモニタリングして、CloudFront が品質スコアに基づいてオリジンを選択していることを確認します。
3. プライマリ AWS リージョンの障害をシミュレートしてフェイルオーバー動作をテストします。
4. フェイルオーバー中に広告挿入が正しく動作し続けます。

次の手順

MQAR を実装したら、次のステップを検討してください。

- MQAR 設定のモニタリングとトラブルシューティングを設定する (「」を参照 [CloudFront と MediaTailor の統合のモニタリングとトラブルシューティング](#))
- を使用した自動デプロイの実装 AWS CloudFormation (「」を参照 [MediaTailor と CDN を自動化する CloudFormation](#))

CloudFront と MediaTailor の統合のモニタリングとトラブルシューティング

AWS Elemental MediaTailor Amazon CloudFront との統合には、最適なパフォーマンスを維持するために継続的なモニタリングとトラブルシューティングが必要です。CloudFront と MediaTailor の統合を実装したら、パフォーマンスをモニタリングし、問題のトラブルシューティングに備えます。コンテンツ配信ネットワーク (CDN) には、ディストリビューションのパフォーマンスを理解し、潜在的な問題を特定するのに役立つツールが用意されています。

統合のモニタリングの設定

CloudFront と MediaTailor の統合のパフォーマンスを追跡するには、以下のモニタリング戦略を実装します。

CloudFront と MediaTailor の統合のモニタリングを設定するには

1. CloudFront 標準ログ記録を有効にします。
 - a. CloudFront コンソールで、ディストリビューションを選択します。
 - b. [ログ] タブを選択します。
 - c. 標準ログで、編集 を選択します。
 - d. On を選択し、ログストレージ用に Amazon S3 バケットを設定します。
 - e. [Save changes] (変更の保存) をクリックします。
2. CloudFront リアルタイムログを設定します。
 - a. CloudFront コンソールで、ディストリビューションを選択します。
 - b. [ログ] タブを選択します。
 - c. リアルタイムログで、編集 を選択します。
 - d. On を選択し、Amazon Kinesis Data Streams または Amazon Data Firehose 配信ストリームを設定します。
 - e. ログ設定に以下のフィールドを含めます。
 - time-to-first-byte - 応答時間
 - sc-status - HTTP ステータスコード
 - c-ip - ビューワー IP アドレス
 - cs-uri-stem - リクエスト URI パス

- cs-user-agent - ユーザーエージェント
 - x-edge-result-type - 結果タイプ (ヒット、ミス、エラー)
- f. [Save changes] (変更の保存) をクリックします。
3. CloudWatch ダッシュボードを作成する:
- a. CloudWatch コンソールで、ダッシュボードを選択します。
 - b. [ダッシュボードを作成] を選択します。
 - c. これらの CloudFront メトリクスのウィジェットを追加します。
 - リクエスト
 - BytesDownloaded
 - 4xxErrorRate
 - 5xxErrorRate
 - TotalErrorRate
 - CacheHitRate
4. CloudWatch アラームを設定します。
- a. CloudWatch コンソールで、アラームを選択します。
 - b. [アラームの作成] を選択します。
 - c. これらの条件のアラームを作成します。
 - 5xxErrorRate > 1% を 5 分間
 - 4xxErrorRate が 5% を 5 分間超える
 - CacheHitRate が 30 分間 80% 未満

MQAR パフォーマンスのモニタリング

MQAR を使用する場合、モニタリングは CloudFront がオリジンを選択する方法と、品質スコアが期待を満たしているかどうかを理解するのに役立ちます。リアルタイムログは、これらの決定を発生時に表示します。

MQAR のパフォーマンスをモニタリングするには

1. CloudFront デイストリビューションのリアルタイムログを有効にします。
2. ログ設定に以下のフィールドを含めます。

- `r-host` - 選択したオリジンのホスト名
 - `sr-reason` - オリジン選択の理由
 - `x-edge-mqcs` - メディア品質信頼スコア
3. Amazon Kinesis Data Streams または Amazon Data Firehose でログ送信先を設定します。
 4. これらのメトリクスに基づいてダッシュボードまたはアラートを作成し、品質スコアとオリジン選択の決定をモニタリングします。

Example MQAR モニタリング用の CloudWatch ダッシュボードのサンプル

次のメトリクスを使用して CloudWatch ダッシュボードを作成します。

- リージョン別のオリジン選択数
- 経時的な平均品質スコア
- フェイルオーバーイベント
- オリジン別の 4xx および 5xx エラー率

リアルタイムログの設定の詳細については、CloudFront デベロッパーガイドの [「リアルタイムログ」](#) を参照してください。

一般的な問題のトラブルシューティング

CloudFront と MediaTailor の統合では、慎重な計画を立てていても問題が発生する可能性があります。一般的な問題と解決策を理解することで、問題をすばやく解決し、視聴者への影響を最小限に抑えることができます。

CloudFront と MediaTailor の統合で問題が発生した場合は、以下の一般的な問題と解決策を確認してください。

マニフェストキャッシュの問題

症状: 古いマニフェストまたは広告が更新されない

解決策: マルチバリエーションプレイリスト、メディアプレイリスト、MPD パスに `CachingDisabled` キャッシュポリシーを使用していることを確認します。設定でクエリパラメータが正しく転送されていることを確認します。

CORS エラー

症状: コンテンツにアクセスするときにブラウザコンソールに CORS エラーが表示される

解決策: レスポンスヘッダーポリシーを適切な CORS ヘッダーで設定し、キャッシュ動作に関連付けます。

MQAR が機能しない

症状: オリジン選択が品質スコアを無視する

解決策: オリジングループ設定でメディア品質スコアオプションが有効になっていることを確認します。Lambda@Edge オリジン向けトリガーを使用していないことを確認します。

広告挿入の失敗

症状: 広告が正しく挿入されない

解決策: MediaTailor 設定がコンテンツセグメントプレフィックスの CloudFront デイストリビューションを指していることを確認します。設定で必要なすべてのヘッダーが転送されていることを確認します。

より複雑な問題には、次のトラブルシューティングアプローチを使用できます。

1. CloudFront デイストリビューションログでエラーパターンを確認する
2. ブラウザ開発者ツールを使用してネットワークリクエストを検査する
3. MediaTailor のマニフェストコンテンツと CloudFront 経由で配信されるコンテンツを直接比較する
4. 詳細なログ記録をサポートするシンプルなプレイヤーでテストする

トラブルシューティングの詳細については、MediaTailor ユーザーガイドの [「トラブルシューティング」](#) セクションを参照してください。

ワークフローのトラブルシューティング

CloudFront と MediaTailor の統合に関する問題のトラブルシューティングを行うには、次の体系的なアプローチに従います。

CloudFront と MediaTailor の統合の問題をトラブルシューティングするには

1. 特定の問題を特定します。

- a. 問題がすべてのビューワーに影響するか、一部のみに影響するかを判断する
 - b. 影響を受けるコンテンツタイプ (マニフェスト、セグメント、またはその両方) を特定する
 - c. エラーメッセージや症状を書き留める
2. CloudFront ログを確認します。
 - a. エラーステータスコードを探す (4xx または 5xx)
 - b. キャッシュヒット/ミスパターンを確認する
 - c. リクエストが正しいオリジンにルーティングされていることを確認します。
3. 設定を確認します。
 - a. キャッシュ動作で正しいパスパターンを確認する
 - b. キャッシュポリシーが正しく適用されていることを確認する
 - c. オリジンリクエストポリシーが必要なヘッダーを転送していることを確認します。
4. 直接アクセスをテストする:
 - a. MediaTailor から直接コンテンツにアクセスしてみてください (CloudFront をバイパス)
 - b. 直接アクセスと CloudFront アクセスのレスポンスを比較する
5. ソリューションを実装します。
 - a. 検出結果に基づいて適切な修正を適用する
 - b. テストして問題が解決されたことを確認する
 - c. 今後の参照用に問題とソリューションを文書化する

次の手順

CloudFront と MediaTailor の統合のモニタリングとトラブルシューティングを設定したら、次のステップを検討してください。

- 使用した自動デプロイの実装 AWS CloudFormation (「」を参照 [で MediaTailor と CDN を自動化する CloudFormation](#))
- 一般的な運用シナリオとトラブルシューティング手順のランブックを作成する
- 一般的な問題の自動修復を設定する

MediaTailor 広告配信のサードパーティー CDNs を設定する

Akamai や"などのサードパーティー CDNs は、帯域幅コストを削減しながら、AWS Elemental MediaTailor 広告配信のパフォーマンスとスケーラビリティを大幅に向上させることができます。ただし、パーソナライズされた広告の CDN 設定には、標準のビデオ配信とは異なる特定の設定が必要です。

サードパーティーの CDNs 「」を参照してください[MediaTailor マニフェストクエリパラメータ](#)。動変数を使用した高度なルーティング設定については、「」を参照してください[ADS リクエストの MediaTailor 動的广告変数](#)。

このガイドでは、MediaTailor と最適に連携するようにサードパーティー CDN を設定する完全なプロセスについて説明します。2 つの重要な動作を設定する方法について説明します。

- マニフェストバイパス: マニフェストキャッシュを防止することで、各ビューワーがパーソナライズされた広告挿入を受信できるようにします
- セグメントキャッシュ: ビデオコンテンツを効率的にキャッシュすることで、パフォーマンスを最大化し、コストを削減

設定プロセスには通常 30～60 分かかり、プロバイダーの選択、セットアップ、検証、最適化が含まれます。完了すると、最適な視聴エクスペリエンスを維持しながら、パーソナライズされた広告を効率的に配信する CDN 設定が完成します。

Note

このガイドでは、Akamai と"の設定に焦点を当てています。CloudFront のセットアップ手順については、「」を参照してください[CloudFront 統合](#)。

前提条件

MediaTailor でサードパーティー CDN を設定する前に、以下があることを確認してください。

- コンテンツオリジンと広告決定サーバーを含むアクティブな MediaTailor 設定
- CDN の設定インターフェイスへのアクセス
- コンテンツで使用するファイル拡張子のリスト (.m3u8、.mpd、.ts など)
- リファレンス用の CDN プロバイダーのドキュメント

サードパーティー CDN ではなく CloudFront のセットアップ手順については、「」を参照してください [CloudFront 統合](#)。CDNs

用語

CDN 設定要件を理解するには、次のマニフェストタイプを知っておく必要があります。

- HLS マニフェスト:
 - 多変量プレイリスト: メディアプレイリストへのリンクを含む最上位マニフェスト
 - メディアプレイリスト: コンテンツセグメントへのリンクを含む第 2 レベルのマニフェスト
- DASH マニフェスト:
 - MPD (メディアプレゼンテーションの説明): DASH マニフェストの標準用語

このガイドでは、一般的な設定要件について説明するときに、すべてのマニフェストファイル (多変量プレイリスト、メディアプレイリスト、MPDs) をマニフェストとしてまとめて参照します。

すべてのプロバイダーに適用される一般的な CDN 設定原則については、「」を参照してください [MediaTailor との CDN 統合を設定する](#)。

CDN 最適化ガイダンスについては、「」を参照してください [CDN と MediaTailor の統合のパフォーマンス最適化ガイド](#)。

CDN キャッシュルールを設定する

MediaTailor 広告配信では、コンテンツがビューワーに到達する方法を決定するため、CDN キャッシュ設定が重要です。適切な設定により、セグメントを効率的にキャッシュしながら、各ビューワーのマニフェストをパーソナライズしたままにして、オリジンの負荷を軽減し、パフォーマンスを向上させます。

この設定は通常、CDN プロバイダーごとに 15~30 分かかり、2 つの異なる動作が必要です。

- マニフェスト処理: キャッシュを防止して、各ビューワーがパーソナライズされた広告挿入を受信できるようにします
- セグメントキャッシュ: ビデオコンテンツのキャッシュ効率を最大化して配信パフォーマンスを向上させます

最適な広告配信のために CDN のキャッシュルールを設定するには、次の手順に従います。

特定の手順については、以下のタブから CDN プロバイダーを選択してください。

Akamai

Akamai プロパティで次の 2 つの動作を設定します。

- キャッシュを防ぐためのマニフェスト処理
- 最適なパフォーマンスを実現するセグメントキャッシュ

マニフェスト配信を設定する

各ビューワーがパーソナライズされた広告を受信できるように、マニフェストがキャッシュされないように Akamai CDN を設定します。

マニフェストファイルには、MediaTailor がビューワーごとに生成するパーソナライズされた広告挿入ポイントが含まれています。これらのファイルをキャッシュすると、すべての視聴者に同一の広告が表示され、パーソナライズされた広告の目的が損なわれます。

マニフェストリクエスト (.m3u8、.mpd、または .smil で終わるファイル) については、次の手順に従います。

1. マニフェストファイル拡張子 (.m3u8、.mpd、.smil) に一致する動作を作成する
2. キャッシュオプションをストアなしに設定する
3. すべてのクエリパラメータを含めるようにキャッシュキーを設定する
4. 適切なオリジンルーティングのために転送ホストヘッダーを有効にする
5. すべてのヘッダーのヘッダー転送を設定します。最小要件については、「」を参照してください [MediaTailor CDN 統合に必要なヘッダー](#)。

セグメント配信を設定する

動画セグメントをキャッシュするように Akamai CDN を設定して、CDN 効率を最大化し、オリジンの負荷を軽減します。

動画セグメントは、パーソナライズされた情報が含まれていないため、安全にキャッシュできる実際のコンテンツファイルです。適切なセグメントキャッシュにより、帯域幅のコストが削減され、視聴者の再生パフォーマンスが向上します。

セグメントリクエスト (.ts、.mp4、.m4s など) については、次の手順に従います。

1. セグメントファイル拡張子 (.ts、.mp4、.m4s) に一致する動作を作成する

2. Honor オリジンキャッシュコントロールを Yes に設定する
3. オリジンヘッダーがない場合のtime-to-live (TTL) 設定を構成します。
 - デフォルト TTL: 86400 秒 (24 時間)
 - 最大 TTL: 604800 秒 (7 日間)

 Note

これらの動作を設定したら、Akamai Control Center でプロパティの変更を有効にします。

変更はアクティベーション後に有効になります。

Fastly

次の 2 つの設定を、" サービスで作成します。

- キャッシュを防ぐためのマニフェスト処理
- 最適なパフォーマンスを実現するセグメントキャッシュ

マニフェスト配信を設定する

各ビューワーがパーソナライズされた広告コンテンツを受信できるように、マニフェストファイルのキャッシュをバイパスするように、" CDN を設定します。

マニフェストファイルは、適切な広告パーソナライゼーションを確保するために、リクエストごとに MediaTailor に到達する必要があります。これらのファイルのキャッシュをバイパスすると、各ビューワーの一意のターゲットパラメータが正しく処理されます。

マニフェストリクエストの場合は、次の手順に従います。

1. マニフェストパスを識別するリクエスト条件を作成する
2. キャッシュ条件を に設定する これらのリクエストに対してキャッシュしない
3. すべてのクエリパラメータを含めるように転送設定を構成する
4. ヘッダー転送設定User-Agentに を追加する

セグメント配信を設定する

配信パフォーマンスを向上させ、オリジントラフィックを減らすために、動画セグメントをキャッシュするように、" CDN を設定します。

セグメントキャッシュは、コスト効率の高い配信と最適なビューワーエクスペリエンスに不可欠です。これらのファイルはすべてのビューワーで同じであり、CDN キャッシュの大きな利点があります。

セグメントリクエストの場合は、次の手順に従います。

1. セグメントパスを識別するリクエスト条件を作成する
2. オリジンキャッシュヘッダーを優先するようにキャッシュ設定を設定する
3. オリジンヘッダーがない場合のデフォルトのtime-to-live (TTL) を 86400 秒 (24 時間) に設定する

Note

これらの変更を行った後、新しいバージョンの「" サービス」をアクティブ化します。設定はアクティベーション後に有効になります。

CDN 設定を確認する

検証により、本番トラフィックを経由する前に CDN 設定が正しく動作することを確認します。これらのテストでは、広告パーソナライゼーションが適切に機能し、キャッシュによって期待されるパフォーマンス上の利点が得られることを確認します。

完全な検証には通常 10 ~ 15 分かかり、可能であれば複数の地理的場所から実行する必要があります。

CDN をセットアップしたら、次のチェックを実行して正しく動作していることを確認します。

1. マニフェストのパーソナライゼーションをテストする:
 - a. 異なる広告パラメータを使用して同じコンテンツ URL をリクエストする
 - b. 各リクエストが異なる広告挿入を返すことを確認する
2. テストセグメントキャッシュ:
 - a. セグメントキャッシュヒット率の CDN メトリクスを確認する (90% 以上である必要があります)

- b. オリジントラフィックをモニタリングして、直接配信よりも低いことを確認する
3. 再生パフォーマンスをテストします。
- a. さまざまな場所から CDN 経由でコンテンツを再生する
 - b. バッファリングの問題なしでスムーズな再生を検証する

包括的なテスト方法と高度な検証手順については、「」を参照してください[CDN と MediaTailor の統合のテストと検証](#)。

CDN パフォーマンスの最適化

基本設定を確認したら、これらの最適化を実装してパフォーマンスを最大化し、コストを最小限に抑えます。

主要なパフォーマンスメトリクスのモニタリング

これらのメトリクスを追跡して、最適なパフォーマンスを確保します。

キャッシュヒット率

ターゲット: 動画セグメントで 90% を超える

影響: 比率が高いほどオリジンの負荷が軽減され、ビューワーエクスペリエンスが向上します

モニタリング: CDN プロバイダーの分析ダッシュボードを毎日確認する

オリジン応答時間

ターゲット: マニフェストリクエストの場合は 200 ミリ秒未満

影響: マニフェスト配信の高速化により、視聴者の起動時間を短縮

モニタリング: 500 ミリ秒を超える応答時間のアラートを設定する

エラー率

ターゲット: すべてのリクエストで 0.1% 未満

影響: エラー率が高い場合は、設定の問題またはオリジンの問題を示します。

モニタリング: 1% を超えるエラー率のアラートを設定する

キャッシュ動作を微調整する

コンテンツ特性とビューワーパターンに基づいて、これらの設定を調整します。

セグメント TTL 最適化

ライブコンテンツ: 短い TTL (1~4 時間) を使用してタイムリーな更新を確保する

VOD コンテンツ: キャッシュ効率を最大化するために長い TTL (24~48 時間) を使用する

広告セグメント: 頻繁に更新される広告コンテンツでは、TTL を短くすること (30 分から 2 時間) を検討してください。

すべての MediaTailor ワークフローにわたる包括的な TTL レコメンデーションとキャッシュ戦略については、「」を参照してください[CDN と MediaTailor の統合のキャッシュ最適化](#)。

地理的最適化

マルチリージョンオリジン: ビューワーの場所に基づいてオリジンの選択を設定する

エッジロケーション: ビューワーの集中度が高いリージョンで追加のエッジロケーションを有効にする

フェイルオーバー: 高可用性のためにバックアップオリジンを設定する

コストの最適化

パフォーマンスを維持しながら CDN コストを削減するには、以下の戦略を実装します。

- 圧縮: マニフェストファイルの gzip 圧縮を有効にして帯域幅の使用量を減らす
- ページ戦略: フルキャッシュクリアの代わりに選択的キャッシュページを実装する
- トラフィック分析: トラフィックパターンを毎月確認し、最適化の機会を特定する
- 階層の選択: パフォーマンス要件に基づいて適切な CDN サービス階層を使用する

サードパーティーの CDN 問題のトラブルシューティング

CDN 設定の問題は、通常、広告パーソナライゼーションの問題またはパフォーマンスの低下として現れます。この体系的なアプローチを使用して、MediaTailor 広告配信に影響する最も一般的な問題を特定して解決します。

ほとんどのトラブルシューティングは、特定の症状を確認し、対応する解決策を適用することで、15～30 分以内に完了できます。

視聴者が広告配信または再生品質に問題がある場合は、このガイドを使用して一般的な CDN 設定の問題を特定して解決します。

Akamai: キャッシュされたマニフェスト

症状: 異なるターゲティングパラメータを設定しても、ビューワーには同じ広告が表示されません。

解決策: マニフェストパスにストアなしキャッシュオプションを適用したことを確認します。

また、キャッシュキーにクエリパラメータが含まれていることも確認します。

「”: 不正なキャッシュキー」

症状: 視聴者の広告パーソナライゼーションに一貫性がない。

ビューワーには、他のビューワー向けの広告が表示される場合もあります。

解決策: キャッシュキーにすべてのクエリパラメータを含めるように転送設定を設定したことを確認します。

一般: 高オリジントラフィック

症状: オリジンサーバーのトラフィックが予想せず多い

解決策: セグメントキャッシュ設定とtime-to-live (TTL) 値を確認します。

CDN メトリクスのキャッシュヒット率を確認します。

全般: 再生エラー

症状: ビューワーにバッファリングまたは再生の失敗が発生する

解決策: CDN ルーティングルールとオリジンの状態を確認します。

必要なヘッダーがすべて正しく転送されていることを確認します。

Note

これらのソリューションで問題が解決しない場合は、CDN プロバイダーのドキュメントを確認してください。サポートチームに連絡して、追加のトラブルシューティング手順を依頼することもできます。

CDN の一般的なトラブルシューティングガイダンスについては、「」を参照してください [MediaTailor と CDN の統合に関する問題のトラブルシューティング](#)。

CDN と MediaTailor の統合のパフォーマンス最適化ガイド

AWS Elemental MediaTailor パフォーマンスは、体系的なコンテンツ配信ネットワーク (CDN) 最適化によって最大化できます。サーバー側の広告挿入 (SSAI)、チャネルアセンブリ、または複合ワークフローを実装する場合でも、最適化の原則とパフォーマンスターゲットは一貫しています。このガイドでは、すべての MediaTailor 実装に適用される包括的な最適化手法とベンチマークについて説明します。

動変数と設定エイリアスを使用した高度なルーティング最適化については、「」を参照してください [ADS リクエストの MediaTailor 動的広告変数](#)。クエリパラメータの最適化戦略については、「」を参照してください [MediaTailor マニフェストクエリパラメータ](#)。

最適化ワークフローの概要:

1. キャッシュの設定 - 適切な TTL 値とキャッシュ動作を設定します。
2. ルーティングの最適化 - リクエストルーティングとオリジンポリシーを設定する
3. パフォーマンスの測定 - 確立されたベンチマークに照らして追跡する
4. 高度な手法を適用する - 追加の最適化機能を実装する

トピック

- [CDN と MediaTailor の統合のキャッシュ最適化](#)
- [CDN と MediaTailor の統合のルーティング最適化をリクエストする](#)
- [CDN と MediaTailor の統合のパフォーマンスベンチマーク](#)
- [CDN と MediaTailor の統合のための高度な最適化手法](#)

CDN と MediaTailor の統合のキャッシュ最適化

AWS Elemental MediaTailor キャッシュ要件は、ワークフロータイプとコンテンツ形式によって異なります。最適なパフォーマンス、コスト効率、視聴者エクスペリエンスを得るには、適切なキャッシュ設定が不可欠です。以下のセクションでは、さまざまな MediaTailor 実装の詳細なキャッシュガイドランスを提供します。

サーバー側の広告挿入 (SSAI) キャッシュ

サーバー側の広告挿入ワークフローでは、パーソナライズされたマニフェストをキャッシュすることはできませんが、コンテンツと広告セグメントは積極的にキャッシュする必要があります。

SSAI CDN キャッシュ設定

コンテンツタイプ	TTL	パスパターン	キャッシュキー要素
多変量プレイリスト	0 秒	/v1/master/*	URL パス + すべてのクエリパラメータ
メディアプレイリスト	0 秒	/v1/マニフェスト/*	URL パス + すべてのクエリパラメータ
DASH MPDs	0 秒	/v1/dash/*	URL パス + すべてのクエリパラメータ
コンテンツセグメント	24 時間以上	コンテンツ固有のパス	URL パスのみ
広告セグメント	24 時間以上	/v1/segment/*	URL パスのみ

- パーソナライズされたマニフェストの TTL を 0 秒に設定して、視聴者が up-to-date 広告コンテンツを受信できるようにします。
- キャッシュ効率を最大化するために、コンテンツセグメントと広告セグメントの長い TTL 値を設定する
- ターゲット広告をサポートしている場合、キャッシュキーにパーソナライゼーションパラメータを含むキャッシュ動作を設定する
- CDN レベルでリクエストの折りたたみを実装して、同時リクエストを効率的に処理する

推奨される TTL 設定

SSAI のパフォーマンスを最適化するには、次の特定の TTL 設定で CDN キャッシュポリシーを設定します。

SSAI TTL 設定

コンテンツタイプ	TTL 設定	推奨値
広告セグメント	最小 TTL	1 秒
広告セグメント	最大 TTL	86400 秒 (24 時間)
広告セグメント	デフォルト TTL	86400 秒 (24 時間)
コンテンツセグメント	最小 TTL	1 秒
コンテンツセグメント	最大 TTL	86400 秒 (24 時間)
コンテンツセグメント	デフォルト TTL	86400 秒 (24 時間)

これらの設定により、以下が保証されます。

- 最小 TTL 1 秒: 過剰なオリジンリクエストを防止しながら、必要に応じてキャッシュを迅速に無効化できます
- 最大 TTL 24 時間: キャッシュ効率とコンテンツの鮮度要件のバランスを取ります
- デフォルトの TTL は 24 時間: 明示的なキャッシュ制御ヘッダーを持たないセグメントに最適なキャッシュを提供します

サーバーガイド広告挿入 (SGAI) キャッシュ

サーバーガイド広告挿入 (SGAI) は、予測可能な URL パターンを使用するキャッシュ可能なメディアマニフェストを通じて効率的な CDN キャッシュを可能にします。このセクションでは、最適な SGAI パフォーマンスのための CDN 固有の設定要件に焦点を当てます。

SGAI の CDN キャッシュ設定

SGAI 固有のキャッシュ動作を使用して CDN を設定します。

SGAI CDN キャッシュ設定

コンテンツタイプ	TTL	パスパターン	キャッシュキー要素
SGAI マルチバリエーション プレイリスト (キャッシュしない)	0 秒 (キャッシュしない)	/v1/master/*	URL パス + 選択したクエリパラメータ
SGAI メディアプレイリスト	1~4 秒 (セグメントの半分の長さ)	/v1/i-media/*	URL パス + 選択したクエリパラメータ
アセットリストレスポンス (キャッシュしない)	0 秒 (キャッシュしない)	/v1/interstitials/*	URL パス + すべてのクエリパラメータ
広告セグメント	24 時間以上	広告固有のパス	URL パスのみ

キャッシュ動作設定

SGAI コンテンツの専用キャッシュ動作を設定します。

- SGAi マニフェストの動作 - 1~4 秒の TTL を持つ /v1/i-media/* パスのキャッシュ動作を作成する
- アセットリストの動作 - 0 秒の TTL を持つ /v1/interstitials/* パスのキャッシュ動作を作成します。
- クエリパラメータ処理 - キャッシュ効率を最大化するために、キャッシュキーに必須のターゲットパラメータのみを含めます
- オリジンリクエストヘッダー - キャッシュ性を維持しながら、広告ターゲティングに必要なヘッダーを転送する

チャンネルアセンブリキャッシュ

チャンネルアセンブリワークフローでは、マニフェストを短期間キャッシュできますが、セグメントは積極的にキャッシュする必要があります。

チャンネルアセンブリ CDN キャッシュ設定

コンテンツタイプ	VOD TTL	ライブ TTL	パスパターン	キャッシュキー要素
多変量プレイリスト	5 ~ 30 分	5 ~ 10 秒	チャンネル固有のパス	URL パス + すべてのクエリパラメータ
メディアプレイリスト	5 ~ 30 分	2 ~ 5 秒	チャンネル固有のパス	URL パス + すべてのクエリパラメータ
DASH MPDs	5 ~ 30 分	5 ~ 10 秒	チャンネル固有のパス	URL パス + すべてのクエリパラメータ
コンテンツセグメント	24 時間以上	5 ~ 15 分	コンテンツ固有のパス	URL パスのみ
広告セグメント	24 時間以上	24 時間以上	広告固有のパス	URL パスのみ

- マニフェストの短い TTL 値を設定して、ビューワーがup-to-dateプログラミングを受け取れるようにします。
- キャッシュ効率を最大化するために、コンテンツセグメントの長い TTL 値を設定する
- タイムシフト表示をサポートしている場合、キャッシュキーにタイムシフトパラメータを含むキャッシュ動作を設定する
- キャッシュキーにクエリパラメータを含めて、タイムシフトされた表示リクエストを適切に処理する

詳細な TTL 設定とベストプラクティスについては、「」を参照してください[the section called “キャッシュの最適化”](#)。

SSAI とチャネルアセンブリのキャッシュの組み合わせ

チャネルアセンブリと SSAI の両方を実装する場合は、競合を回避し、パフォーマンスを最適化するために、両方のサービスでキャッシュ戦略が一貫していることを確認してください。

ワークフローキャッシュ設定の組み合わせの比較

コンテンツタイプ	チャネルアセンブリ	SSAI	複合レコメンデーション
VOD マニフェスト	5～30 分	0 秒	(別の設定を使用)
ライブマニフェスト	2～10 秒	0 秒	(別の設定を使用)
SSAI VOD マニフェスト	5～30 分	5～30 分	5～30 分 (キャッシュ可能なマニフェスト)
SSAI Live マニフェスト	2～4 秒	2～4 秒	2～4 秒 (キャッシュ可能なマニフェスト)
コンテンツセグメント	24 時間以上	24 時間以上	24 時間以上 (一定)
広告セグメント	24 時間以上	24 時間以上	24 時間以上 (一定)

この設定により、パーソナライズされた広告挿入のためにビューワーがup-to-dateマニフェストを受信できるようにしながら、キャッシュ効率を最大化できます。

CDN と MediaTailor の統合のルーティング最適化をリクエストする

すべての AWS Elemental MediaTailor CDN 統合に次のルーティング最適化を実装します。

- マニフェストリクエストとセグメントリクエストに個別のキャッシュ動作を作成する
- ヘッダー転送を制御するようにオリジンリクエストポリシーを設定する
- 適切なエラー処理とフェイルオーバーメカニズムを設定する
- CDN で使用可能な場合はオリジンシールドを実装して、オリジンの負荷を軽減します。
- CDN レベルでリクエストの折りたたみを実装して、同時リクエストを効率的に処理する

CDN と MediaTailor の統合のパフォーマンスベンチマーク

AWS Elemental MediaTailor CDN 統合を最適化するときには、これらのパフォーマンスベンチマークを目指します。これらのターゲットは、SSAI、チャネルアセンブリ、複合ワークフローを含むすべての MediaTailor 実装に適用されます。

キャッシュヒット率のターゲット

コンテンツセグメント: 95% を超えるキャッシュヒット率

広告セグメント: 90% を超えるキャッシュヒット率

マニフェスト: SSAI には適用されません (パーソナライズされた広告挿入用にキャッシュしないでください)。チャネルアセンブリの場合は 85% 以上

レイテンシーベンチマーク

マニフェストリクエストのレイテンシー: 100 ミリ秒未満 (P95)

コンテンツセグメント配信: 50 ミリ秒未満 (P95)

広告セグメント配信: 75 ミリ秒未満 (P95)

End-to-endの起動時間: 2 秒未満

オリジンロードメトリクス

ビューワーあたりのオリジンリクエスト: ビューワーあたり 1 分あたり 0.1 リクエスト未満

ビューワーあたりのオリジン帯域幅: ビューワー帯域幅全体の 5% 未満

エラーレートのターゲット

マニフェストエラー: 0.1% 未満

セグメントエラー: 0.01% 未満

プレイヤーが報告したリバッファリング: 1% 未満

スケーラビリティベンチマーク

ピークイベント中にパフォーマンスを低下させることなく通常のトラフィックの 10 倍をサポート

チャンネルあたり 1 秒あたり 1000 件を超えるリクエストを処理する機能

Amazon CloudWatch メトリクスを使用して、これらのパフォーマンス指標を追跡します。詳細なモニタリング手順については、「」を参照してください[モニタリングツールのセットアップ](#)。

CDN と MediaTailor の統合のための高度な最適化手法

基本的なキャッシュとルーティングの最適化を実装したら、パフォーマンスをさらに強化するために、以下の高度な手法を検討してください。

Origin Shield の実装

Origin Shield は、CDN エッジロケーションとオリジンサーバーの間にキャッシュレイヤーを追加し、オリジンの負荷を軽減してパフォーマンスを向上させます。

- トラフィックの多いコンテンツとライブストリーミングで Origin Shield を有効にする
- MediaTailor リージョンに近い Origin Shield の場所を選択する
- Origin Shield キャッシュヒット率をモニタリングし、必要に応じて調整する
- グローバルデプロイで複数の Origin Shield の場所を検討する

コンテンツ圧縮の最適化

コンテンツ圧縮を最適化して帯域幅を減らし、配信速度を向上させます。

- マニフェストファイルの gzip 圧縮を有効にする
- MediaTailor マニフェスト圧縮の Accept-Encoding ヘッダー転送を設定する
- サポートされる Brotli 圧縮を使用して帯域幅をさらに節約する
- 既に圧縮されたビデオセグメントを圧縮しない

リージョン最適化戦略

リージョン戦略を通じて、グローバルオーディエンスのパフォーマンスを最適化します。

- グローバルオーディエンス向けに複数のリージョンに MediaTailor 設定をデプロイする
- ジオルーティングを使用して、最も近い MediaTailor リージョンにビューワーを誘導する
- 高可用性のためにリージョンフェイルオーバーを設定する

- リージョン別パフォーマンスメトリクスを個別にモニタリングする

CDN と MediaTailor の統合に関するアーキテクチャ上の考慮事項

最適なパフォーマンスと信頼性を確保するために、コンテンツ配信ネットワーク (CDN) をアーキテクチャに正しく配置します AWS Elemental MediaTailor。推奨されるアーキテクチャでは、CDN は MediaTailor とオリジンではなく、ビューワーと MediaTailor の間に配置されます。

詳細なアーキテクチャ図とワークフローの説明については、以下のトピックを参照してください。

- [CDN を使用した広告挿入](#) 広告挿入アーキテクチャ図と詳細なワークフロー用
- [CDN アーキテクチャを理解する](#) チャンネルアセンブリのアーキテクチャ図とワークフロー

CDN をアーキテクチャに正しく配置します。

1. プレイヤーと MediaTailor の間に CDN を配置します (MediaTailor とオリジンの間ではありません)。

このアーキテクチャにより、CDN は広告セグメントとコンテンツセグメントをキャッシュできます。同時に、MediaTailor はビューワーごとにパーソナライズされたマニフェストを生成できます。

2. リクエストタイプごとに個別のキャッシュ動作を作成します。

- マニフェストリクエスト (キャッシュなし)
- コンテンツセグメント (長い TTL)
- 広告セグメント (長い TTL)

3. 適切なエラー処理を設定します。

- サービス中断中のリクエストの繰り返してオリジンに負担をかけないように、ネガティブキャッシュ (エラーレスポンスを一時的に保存) を設定します。キャッシュが負の場合、CDN はエラーレスポンス (404 や 500 エラーなど) を一時的に保存し、存在しないコンテンツや一時的に使用できないコンテンツのリクエストが繰り返されるのを防ぎます。
- 適切なエラーレスポンスコードと再試行動作を設定する

4. 中間キャッシュを実装する (オリジンシールド):

オリジンシールドは、CDN エッジロケーションとオリジンサーバーの間に追加のキャッシュレイヤーを作成する機能です。これにより、オリジンサーバーに到達する冗長リクエストの数が減少します。

- エッジロケーションとオリジンの間に中間キャッシュレイヤーを設定する
- キャッシュミス時のオリジンへの冗長リクエストの数を減らす
- CDN インフラストラクチャ全体でキャッシュヒット率を向上させる

MediaTailor の高度な CDN 機能

基本的なコンテンツ配信ネットワーク (CDN) 設定を実装したら、これらの高度な機能を調べて、AWS Elemental MediaTailor ストリーミングプラットフォームのパフォーマンスと信頼性をさらに強化します。

メディア品質対応ルーティング (MQAR)

MQAR は、リアルタイムのネットワークパフォーマンスメトリクスに基づいて最高品質のコンテンツソースを自動的に選択する Amazon CloudFront 機能です。固定オリジンサーバーを使用する代わりに、MQAR はレイテンシーやスループットなどの要因に基づいて、最もパフォーマンスの高いオリジンにリクエストを動的にルーティングします。これにより、ネットワークの変動中でも、ビューワーは可能な限り最高の品質のストリームを受信できます。

Amazon CloudFront を使用する場合は、MQAR を実装して、リアルタイムメトリクスに基づいて最高品質のコンテンツソースを自動的に選択します。詳細については、CloudFront 統合セクション [CloudFront 統合](#) の「」を参照してください。

マニフェストでのフィルタリング

マニフェストフィルタリングを使用すると、MediaTailor がビューワーに配信するマニフェストに含まれるレンディション (同じコンテンツのさまざまな品質バージョン) をカスタマイズできます。フィルタリングは、特定のデバイスやネットワーク条件に適さないレンディションを削除することで、帯域幅の使用を最適化するのに役立ちます。たとえば、モバイルデバイスや低帯域幅接続の 4K レンディションを削除できます。

を使用したマニフェストフィルタリングの実装の詳細については AWS Elemental MediaPackage、[「」](#)を参照してください [MediaPackage CDN 統合](#)。

マルチ CDN 戦略

マルチ CDN 戦略では、複数の CDN プロバイダーを同時に使用して信頼性とパフォーマンスを向上させます。1 つの CDN で問題が発生した場合、トラフィックは自動的に別のプロバイダーに移行できます。このアプローチは、信頼性が重要な注目度の高いライブイベントに特に役立ちます。

MediaTailor を使用したマルチ CDN 戦略の実装については、「」を参照してください[CDN 統合を計画する](#)。

MediaTailor CDN オペレーションとパフォーマンスのモニタリング

AWS Elemental MediaTailor とコンテンツ配信ネットワーク (CDN) の統合を効果的にモニタリングすることで、信頼性の高いコンテンツ配信、最適なパフォーマンス、迅速な問題検出が可能になります。このモニタリングアプローチは、サーバー側の広告挿入 (SSAI)、チャンネルアセンブリ、ワークフローの組み合わせなど、すべての MediaTailor 実装に適用されます。

CDN 統合をモニタリングすると、次のことが可能になります。

- ビューワーに影響を与える前に問題を検出して解決する
- 主要なパフォーマンス指標を追跡し、サービス品質を維持する
- 最適なキャッシュパフォーマンスを維持し、オリジンの負荷を軽減する
- 広告挿入の成功率がビジネス要件を満たしていることを確認する

モニタリングデータに表示される可能性のあるパラメータ関連の問題のトラブルシューティングについては、「」を参照してください[MediaTailor パラメータのトラブルシューティングガイド](#)。クエリパラメータの使用状況をモニタリングする方法については、「」を参照してください[MediaTailor マニフェストクエリパラメータ](#)。

重要な CDN パフォーマンスメトリクス

MediaTailor 実装で CDN の有効性を評価するには、これらのコアメトリクスを追跡します。

キャッシュヒット率

測定対象: CDN キャッシュから提供されたリクエストとオリジンから提供されたリクエストの割合。

ターゲット値:

- コンテンツセグメント: 95% 以上のキャッシュヒット率
- 広告セグメント: 90% 以上のキャッシュヒット率
- マニフェスト: 実装によって異なります (パーソナライズされたマニフェストはキャッシュしないでください)

重要な理由: キャッシュヒット率が高いほど、オリジンの負荷が軽減され、応答時間が短縮され、帯域幅コストが削減されます。

オリジンリクエストボリューム

測定対象: MediaTailor オリジンサーバーに到達するリクエストの数。

ターゲットパターン: キャッシュミスや新しいコンテンツが急増することがあるため、低く安定している必要があります。

重要な理由: オリジンリクエスト量が多いとキャッシュの非効率性が示され、MediaTailor のパフォーマンスに影響する可能性があります。

レスポンスレイテンシー

測定対象: CDN がビューワーリクエストに応答する速度。

ターゲット値:

- キャッシュされたコンテンツ: 100 ミリ秒未満 (P95)
- オリジンリクエスト: 500 ミリ秒未満 (P95)
- マニフェストリクエスト: 100 ミリ秒未満 (P95)
- セグメントリクエスト: 50 ミリ秒未満 (P95)

重要な理由: 低レイテンシーにより、スムーズな再生と良好な視聴体験が保証されます。

エラー率

測定対象: HTTP エラーが発生するリクエストの割合。

ターゲット値:

- 4xx エラー: 合計リクエストの 0.1% 未満
- 5xx エラー: 合計リクエストの 0.01% 未満
- オリジンエラー: オリジンリクエストの 0.05% 未満

重要な理由: エラー率が高い場合は、ビューワーエクスペリエンスに影響する設定の問題またはサービスの問題を示します。

MediaTailor 固有のメトリクス

これらの MediaTailor メトリクスを CDN メトリクスとともにモニタリングして、完全な可視性を実現します。

広告フィルレート

主要なメトリクス: `Avail.FillRate` および `AdDecisionServer.FillRate`

ターゲット値: 両方のメトリクスで 90% を超える

重要な理由: 広告収益と視聴者のエクスペリエンスに直接影響する

マニフェスト生成パフォーマンス

主要なメトリクス: `GetManifest.Latency` および `GetManifest.Errors`

ターゲット値: レイテンシーが 200 ミリ秒未満、エラー率が 1% 未満

重要な理由: 再生の起動時間と信頼性に影響します

広告決定サーバーのヘルス

主要なメトリクス: `AdDecisionServer.Latency`、`AdDecisionServer.Errors`、および `AdDecisionServer.Timeouts`

ターゲット値: レイテンシーが 1000 ミリ秒未満、エラー率が 5% 未満、最小タイムアウト

重要な理由: ADS のパフォーマンスが広告挿入の成功に直接影響する

モニタリングツールのセットアップ

MediaTailor と CDN の統合を効果的にモニタリングするように、これらのツールを設定します。

Amazon CloudWatch との統合

Amazon CloudWatch は、MediaTailor と CDN の統合をモニタリングするための基盤を提供します。

MediaTailor メトリクス

MediaTailor は、リクエスト、レスポンス、エラーを追跡するメトリクスを CloudWatch に自動的に発行します。主なメトリクスには次のものがあります。

- `RequestCount`: MediaTailor へのリクエストの合計数
- `ResponseTime`: MediaTailor レスポンスレイテンシー
- `4xxErrorCount` および `5xxErrorCount`: エラー追跡

MediaTailor メトリクスの完全なリストについては、「」を参照してください [Amazon CloudWatch メトリクス AWS Elemental MediaTailor によるモニタリング](#)。

CDN メトリクス

CloudWatch で以下を含む CDN メトリクス収集を有効にします。

- さまざまなコンテンツタイプのキャッシュヒット率
- オリジンリクエスト数と応答時間
- ステータスコード別のエラー率

CloudFront 固有のメトリクスについては、ディストリビューション設定で詳細モニタリングを有効にします。

ダッシュボード設定

MediaTailor と CDN のパフォーマンスを可視化するダッシュボードを作成します。

1. 統一されたダッシュボードを作成する: MediaTailor と CDN メトリクスを単一の CloudWatch ダッシュボードに結合して、完全な可視性を実現します。
2. ワークフロー別に整理: 実装タイプ (SSAI、チャネルアセンブリ、または結合ワークフロー) 別にメトリクスをグループ化します。
3. 主要なパフォーマンス指標を含めます。
 - キャッシュヒット率の経時的な傾向
 - レスポンスレイテンシーのパーセンタイル (P50、P95、P99)
 - エラー率の傾向とスパイク

ダッシュボード作成の詳細なガイダンスについては、[CloudWatch ダッシュボード](#)」を参照してください。

アラート設定

ビューワーに影響を与える前に問題を検出するようにアラートを設定します。

重要なアラート

重大な問題に対する即時アラートを設定します。

- 高エラー率: 5xx エラーが 5 分間にリクエストの 0.1% を超えた場合に警告
- オリジンサーバーの問題: オリジン応答時間が 2 秒を超えるとアラート
- キャッシュヒット率の低下: キャッシュヒット率がマニフェストの場合は 70%、セグメントの場合は 85% を下回ると警告

警告アラート

パフォーマンスの低下に関する早期警告アラートを設定します。

- レイテンシーの増加: P95 応答時間が 200 ミリ秒を超えるとアラート
- キャッシュ効率の低下: セグメントのキャッシュヒット率が 90% を下回った場合のアラート

実装チェックリスト

このチェックリストを使用して、包括的なモニタリングカバレッジを確保します。

1. メトリクスコレクション:

- ✓ CloudWatch で有効になっている MediaTailor メトリクス
- ✓ CDN 詳細モニタリングが有効

2. ダッシュボードの設定:

- ✓ Unified CloudWatch ダッシュボードが作成されました
- ✓ 適切な時間範囲で視覚化された主要なメトリクス
- ✓ 関連するチームと共有するダッシュボード

3. アラート設定:

- ✓ 即時通知で設定された重要なアラート
- ✓ 早期検出用に設定された警告アラート
- ✓ アラートエスカレーション手順の文書化

4. 運用手順:

- ✓ インシデント対応手順の文書化
- ✓ 定期的なレビュースケジュールの確立
- ✓ チームトレーニング完了

関連トピック

特定のモニタリングシナリオに関する追加のガイダンスについては、以下を参照してください。

- パフォーマンスの最適化: モニタリングデータに基づく詳細な最適化手法については、「」を参照してください [CDN パフォーマンスの最適化](#)。
- トラブルシューティング: モニタリングデータを使用した詳細なトラブルシューティング手順については、ワークフロー固有のトラブルシューティングドキュメントを参照してください。

- ログ分析: 包括的なログ分析とモニタリングについては、「」を参照してください[MediaTailor の CDN 統合ログ分析とエラーコードリファレンス](#)。

CDN と MediaTailor の統合のテストと検証

AWS Elemental MediaTailor コンテンツ配信ネットワーク (CDN) の統合には、信頼性の高い広告配信と最適なパフォーマンスを保証するための徹底的なテストが必要です。体系的なテストは、統合の問題がビューワーに影響を与える前に特定するのに役立ちます。この包括的なアプローチを使用して、さまざまなシナリオ、設定、ロード条件にわたってコンテンツ配信ネットワークと MediaTailor の統合を検証します。

関連するトピック:

- モニタリングやログ記録などの運用設定については、「」を参照してください。 [CDN モニタリング](#)
- 問題が発生した場合のトラブルシューティングについては、「」を参照してください。 [CDN 統合のトラブルシューティング](#)
- パフォーマンス最適化のガイダンスについては、「」を参照してください。 [CDN と MediaTailor の統合のパフォーマンス最適化ガイド](#)

トピック

- [CDN と MediaTailor の統合の前提条件とセットアップのテスト](#)
- [CDN と MediaTailor の統合の体系的なテスト方法](#)
- [CDN と MediaTailor の統合に関するデプロイ前テストチェックリスト](#)
- [CDN および MediaTailor 統合のテストツールとユーティリティのリファレンス](#)

CDN と MediaTailor の統合の前提条件とセットアップのテスト

AWS Elemental MediaTailor コンテンツ配信ネットワーク (CDN) 統合テストには、適切な準備とリソース割り当てが必要です。体系的なテストを開始する前に、必要なリソースとツールが整っていることを確認してください。

必要なリソース:

- 本番稼働設定をミラーリングするように設定された CDN ディストリビューションをテストする
- 既知のコンテンツと広告ソースを使用して MediaTailor 設定をテストする

- 予測可能な特性 (期間、形式、広告時間枠マーカー) でコンテンツをテストする
- 広告決定サーバーまたはモック ADS レスポンスをテストする
- 複数のテストデバイスとプレイヤータイプ

テストツール:

- curl HTTP リクエストテスト用
- ffprobe HLS マニフェスト検証用
- mp4box DASH マニフェスト検証用
- ネットワーク分析用のブラウザ開発者ツール
- end-to-endのテスト用のビデオプレイヤー

CDN と MediaTailor の統合の体系的なテスト方法

AWS Elemental MediaTailor コンテンツ配信ネットワーク (CDN) 統合テストは、包括的なカバレッジを確保するために、体系的で段階的なアプローチに従う必要があります。この構造化されたアプローチに従って、コンテンツ配信ネットワークと MediaTailor 統合を包括的にテストします。各フェーズは、潜在的な問題を分離するために、前のフェーズに基づいて構築されます。

体系的なテストアプローチに関する追加のガイダンスについては、AWS 「Well-Architected フレームワーク」の [「信頼性のテスト」](#) を参照してください。

フェーズ 1: MediaTailor の直接接続をテストする

まず、CDN が関与せずに MediaTailor 機能をテストし、ベースラインを確立します。

1. MediaTailor エンドポイントに直接マニフェストリクエストをテストします。

- HLS マルチバリエーションプレイリストリクエストをテストします。curl -v "https://your-emt-endpoint.mediatailor.region.amazonaws.com/v1/master/hls/config-name/master.m3u8"
- DASH MPD リクエストをテストします。curl -v "https://your-emt-endpoint.mediatailor.region.amazonaws.com/v1/dash/config-name/manifest.mpd"
- マニフェストレスポンスに予想される広告時間枠マーカーが含まれていることを確認する
- セグメント URLs オリジンを指していることを確認する

2. 広告挿入が正しく機能することを確認します。

- さまざまな広告ターゲティングパラメータでテストする
- 広告セグメントが適切にトランスコードされ、使用可能であることを確認する
- 広告時間枠のタイミングと期間を確認する
- 広告が利用できない場合のフォールバック動作のテスト

3. ベースラインパフォーマンスを測定します。

- マニフェストリクエストの応答時間を記録する
- 広告決定サーバーの応答時間を測定する
- セッションの作成と管理をテストする

成功基準: すべての直接 MediaTailor リクエストは、予想される広告コンテンツを含むマニフェストが適切にフォーマットされた HTTP 200 レスポンスを返します。

フェーズ 2: 基本的な CDN 統合をテストする

CDN をリクエストパスに追加し、基本的な機能をテストします。

1. CDN を使用してマニフェストリクエストをテストします。

- シンプルなルーティングルールで CDN を設定する
- CDN エンドポイントを介したマニフェストリクエストのテスト
- CDN が MediaTailor にリクエストを正しく転送することを確認する
- マニフェストレスポンスがキャッシュされていないことを確認する (TTL = 0)

2. テストセグメントのルーティング:

- コンテンツセグメントがオリジンサーバーにルーティングされていることを確認する
- 広告セグメントが MediaTailor 広告ストレージにルーティングされていることを確認する
- セグメントキャッシュ動作をテストする

3. CDN と直接パフォーマンスを比較します。

- 応答時間の差を測定する
- レスポンスの内容の違いを確認する
- エラー処理が正しく機能することを確認する

成功基準: CDN は、パフォーマンスへの影響を最小限に抑えながら MediaTailor およびオリジンサーバーへのリクエストを正常にプロキシする必要があります。

フェーズ 3: クエリパラメータの転送をテストする

クエリパラメータ転送を追加し、広告パーソナライゼーションをテストします。

1. CDN でクエリパラメータ転送を設定します。

- MediaTailor へのすべてのクエリパラメータの転送を有効にする
- セッションの初期化をテストする (セッション ID は最初のリクエスト時に MediaTailor によって自動的に生成されます)
- カスタムターゲティングパラメータを使用してテストする

2. 広告のパーソナライゼーションをテストする:

- 異なるパラメータが異なる広告レスポンスを生成することを確認する
- テストパラメータのエンコーディングと特殊文字
- パラメータが ADS に正しく渡されていることを確認します。

3. セッション管理を検証する:

- セッションの作成と永続性をテストする
- リクエスト間でセッション ID の整合性を検証する
- テストセッションの有効期限処理

成功基準: 広告コンテンツはクエリパラメータによって異なり、セッションは正しく管理されます。

フェーズ 4: ヘッダー転送をテストする

CDN にヘッダー転送を追加し、デバイス固有のターゲットをテストします。

1. すべてのヘッダーのヘッダー転送を設定します。最小要件については、「」を参照してください [MediaTailor CDN 統合に必要なヘッダー](#)。

2. テストデバイスターゲット:

- さまざまな User-Agent 文字列 (モバイル、デスクトップ、テレビ) でテストする
- デバイス固有の広告レスポンスを検証する
- 異なる IP アドレスで地理的ターゲットをテストする

3. CORS 処理を検証する:

- ウェブプレイヤーの互換性について CORS ヘッダーをテストする
- プリフライト OPTIONS リクエストが正しく機能することを確認する
- 異なるドメインからテストする

成功基準: デバイスと地理的ターゲティングが正しく動作し、ウェブプレイヤーが CORS エラーに遭遇しないようにする必要があります。

フェーズ 5: 包括的なシナリオテスト

複数のシナリオでテストして、堅牢なオペレーションを確保します。

1. さまざまなプレイヤータイプでテストします。
 - ウェブブラウザの HLS.js プレイヤー
 - HLS と DASH をサポートする Video.js プレイヤー
 - モバイルデバイスのネイティブプレイヤー
 - スマートテレビとセットトップボックスプレイヤー
2. さまざまなデバイスやプラットフォームでテストします。
 - モバイルデバイス (iOS、Android)
 - デスクトップブラウザ (Chrome、Firefox、Safari、Edge)
 - スマートTVsとストリーミングデバイス
 - 異なるオペレーティングシステムのバージョン
3. さまざまなコンテンツタイプをテストします。
 - ライブストリーミングコンテンツ
 - ビデオオンデマンド (VOD) コンテンツ
 - さまざまな動画形式とビットレート
 - 広告時間枠パターンが異なるコンテンツ
4. 広告ターゲティングシナリオをテストします。
 - さまざまな属性ターゲティングパラメータ
 - 異なるリージョンにわたる地理的ターゲティング
 - 時間ベースのターゲティング (異なる時刻)
 - カスタムビジネスロジックパラメータ

成功基準: プレイヤーとデバイスの組み合わせはすべて、適切な広告ターゲティングで正しく機能する必要があります。

フェーズ 6: 負荷とパフォーマンスのテスト

現実的な負荷条件下でパフォーマンスを検証します。

⚠ Important

負荷テストの前に、[AWS サポート](#)にお問い合わせください。負荷テストとパフォーマンステストを実行する前に、AWS サポートチケットを作成して、計画されたテストを MediaTailor サービスチームに通知します。これにより、以下が保証されます。

- サービスは予想される負荷レベルに対して準備されています
- テストウィンドウ中に適切な容量が利用可能である
- テストが本番稼働用インシデントと間違われない
- テストのベストプラクティスと制限に関するガイダンスを受け取る

サポートチケットに、予想される同時ユーザー、テスト期間、地理的リージョン、テストする予定の特定のシナリオを含めます。

1. 同時ユーザーシナリオをテストします。

- 複数の同時ビューワーをシミュレートする
- CDN スケーリングとキャッシュのパフォーマンスをテストする
- ロード中のオリジンサーバーのパフォーマンスをモニタリングする
- MediaTailor が同時セッションを処理できることを確認する

2. パフォーマンスメトリクスを測定します。

- ロード中の応答時間をモニタリングする
- キャッシュヒット率が期待を満たしていることを確認する (一般的なコンテンツでは>80%)
- さまざまなシナリオで最初のフレームまでの時間を測定する
- ピークロード中のエラー率を追跡する

3. フェイルオーバーシナリオをテストします。

- ADS が使用できない場合の動作をテストする
- オリジンサーバーのフェイルオーバーをテストする
- エラー処理と復旧を検証する
- CDN エッジロケーションフェイルオーバーをテストする

成功基準: システムは、予想される負荷下で許容可能なパフォーマンスを維持し、障害発生時に適切に低下させる必要があります。AWS [サポート](#) に連絡し、実行前にロードテストプランを承認してください。

CDN と MediaTailor の統合に関するデプロイ前テストチェックリスト

AWS Elemental MediaTailor コンテンツ配信ネットワーク (CDN) 統合は、本番デプロイ前に包括的なテストに合格する必要があります。設定変更を本番環境にデプロイする前に、このチェックリストを使用してください。

基本的な機能:

- ☐ マニフェストリクエストが HTTP 200 レスポンスを返す
- ☐ コンテンツセグメントが正しくロードされる
- ☐ 広告セグメントが正しくロードされる
- ☐ 広告時間枠が予想される時間に表示される
- ☐ 再生はコンテンツと広告の間でスムーズに移行します

設定の検証:

- ☐ クエリパラメータが正しく転送される
- ☐ 必要なヘッダーが正しく転送されている
- ☐ マニフェストキャッシュが無効になっている (TTL = 0)
- ☐ セグメントキャッシュが適切に設定されている
- ☐ CORS ヘッダーはウェブプレイヤー用に設定されています

クロスプラットフォームテスト:

- ☐ モバイルデバイスでテスト済み
- ☐ デスクトップブラウザでテスト済み
- ☐ さまざまなプレイヤータイプでテスト済み
- ☐ HLS 形式と DASH 形式の両方をテスト済み

パフォーマンス検証:

- □ロードテストの承認用に作成された AWS サポートチケット
- □応答時間がパフォーマンス目標を達成する
- □キャッシュヒット率は許容されます
- □エラー率が許容範囲内
- □モニタリングとアラートが設定されている

CDN および MediaTailor 統合のテストツールとユーティリティのリファレンス

AWS Elemental MediaTailor コンテンツ配信ネットワーク (CDN) 統合テストには、包括的な検証とデバッグのためのさまざまなツールが必要です。コンテンツ配信ネットワークおよび MediaTailor 統合テストで一般的に使用されるツールのリファレンスガイド。

curl - HTTP リクエストテスト

テストマニフェストリクエスト: `curl -v "https://your-cdn-domain.com/v1/master/hls/config/master.m3u8"`

ヘッダーを使用してテストします。 `curl -H "User-Agent: TestAgent/1.0" "https://your-cdn-domain.com/..."`

パラメータを使用してテストします。 `curl "https://your-cdn-domain.com/...?aws.sessionId=test123"`

ffprobe - HLS マニフェストの検証

HLS 構文を検証します。 `ffprobe -v quiet -print_format json -show_format "https://your-cdn-domain.com/master.m3u8"`

セグメント情報を確認します。 `ffprobe -v quiet -show_entries packet=pts_time "segment.ts"`

mp4box - DASH マニフェストの検証

DASH MPD を検証する: `mp4box -info "https://your-cdn-domain.com/manifest.mpd"`

セグメントのタイミングを確認します。 `mp4box -info segment.m4s`

ブラウザ開発者ツール

ネットワークタブでネットワークリクエストをモニタリングする

コンソールタブで CORS エラーを確認する

リクエスト/レスポンスヘッダーの検査

タイミングとパフォーマンスのメトリクスを分析する

その他のリソース:

- [CloudFront のトラブルシューティング](#) - 包括的な CDN トラブルシューティングガイド
- [CloudFront キャッシュヒット率の向上](#) - パフォーマンス最適化ガイダンス
- [AWS リソースのモニタリング](#) - パフォーマンスモニタリングのベストプラクティス

MediaTailor と CDN の統合に関する問題のトラブルシューティング

この包括的なトラブルシューティングガイドでは、サーバー側の広告挿入 (SSAI)、チャネルアセンブリ、統合など、すべての AWS Elemental MediaTailor 実装における一般的なコンテンツ配信ネットワーク (CDN) AWS Elemental MediaPackage 統合の問題について説明します。CDN と MediaTailor の統合で問題が発生した場合は、この体系的な診断アプローチを使用して根本原因をすばやく特定し、検証済みのソリューションを実装します。

このガイドは、特定のワークフローに関係なく、すべての MediaTailor CDN 統合に適用されます。特定のサービスまたはワークフローに固有の問題については、このガイドの最後にある関連するトラブルシューティングセクションを参照してください。

開始する前に、以下の項目を効率的なトラブルシューティングに備えてください。

- 問題を示すサンプル再生 URLs
- 問題が発生した期間の CDN アクセスログ
- MediaTailor 設定名と AWS リージョン
- プレイヤーのタイプとバージョン (HLS.js 1.4.0、Video.js 8.0 など)
- 問題が発生するデバイスとブラウザの情報

関連するトピック:

- 運用セットアップとトラブルシューティングの準備については、「」を参照してください。 [CDN 統合のトラブルシューティング](#)
- ログ分析とエラーコードリファレンスについては、「」を参照してください。 [CDN 統合ログ分析リファレンス](#)
- エスカレーションと追加のヘルプについては、「」を参照してください。 [CDN 統合のサポートを受ける](#)

MediaTailor と CDN 統合の診断チェックリスト

AWS Elemental MediaTailor コンテンツ配信ネットワーク (CDN) 統合の問題は、さまざまな方法で現れる可能性があります。このチェックリストを使用して、発生している問題のタイプをすばやく特定します。

1. 問題はすべてのビューワーまたは特定のビューワーに影響を与えていますか?
 - すべてのビューワー → CDN または MediaTailor の設定に関する問題の可能性
 - 特定の視聴者 → パーソナライゼーションまたはターゲティングの問題の可能性
2. マニフェストが正しくロードされていますか?
 - いいえ → CDN ルーティングまたは MediaTailor 接続の問題
 - はい、ただし間違ったコンテンツ → キャッシュまたはパーソナライゼーションの問題
3. セグメントが正しくロードされていますか?
 - コンテンツセグメントが失敗 → オリジン接続の問題
 - 広告セグメントが失敗 → 広告配信またはトランスコードの問題
4. 広告が正しく挿入されていますか?
 - 広告が表示されない → ADS の接続と設定を確認する
 - 間違った広告が表示される → 広告ターゲティングパラメータとパーソナライゼーションを確認する
 - 広告の再生に失敗 → 広告トランスコーディングとセグメントの可用性を確認する
5. 再生はスムーズで中断されませんか?
 - バッファリングの問題 → CDN キャッシュのパフォーマンスとオリジン応答時間を確認する
 - 再生エラー → マニフェスト構文とセグメントの可用性を確認する

- 広告移行の問題 → 広告時間枠のタイミングとセグメントの調整を確認する

6. 特定のエラーコードやメッセージはありますか？

- HTTP 4xx エラー → CDN のルーティングと設定を確認する
- HTTP 5xx エラー → オリジンサーバーと MediaTailor サービスの状態を確認する
- プレイヤー固有のエラー → マニフェスト形式とプレイヤーの互換性を確認する

診断に基づく次のステップ:

CDN 設定の問題

CDN ルーティングとキャッシュのトラブルシューティングの詳細については、「」を参照してください [MediaTailor と CDN の統合に関する問題のトラブルシューティング](#)。

マニフェストと再生の問題

マニフェストの検証と再生のトラブルシューティングについては、「」を参照してください [CDN 統合テスト手順](#)。

広告挿入とターゲティングの問題

ADS 接続や広告配信などの広告固有のトラブルシューティングについては、ワークフロー固有のトラブルシューティングドキュメントを参照してください。

パフォーマンスとモニタリングの問題

パフォーマンス分析とモニタリングの設定については、「」を参照してください [MediaTailor CDN オペレーションとパフォーマンスのモニタリング](#)。

ログ分析とエラーコード

詳細なログ分析とエラーコードリファレンスについては、「」を参照してください [MediaTailor の CDN 統合ログ分析とエラーコードリファレンス](#)。

テストと検証

包括的なテスト手順については、「」を参照してください [CDN と MediaTailor の統合のテストと検証](#)。

すぐにサポートが必要な場合、またはリンクされたリソースを使用して問題を解決できない場合は、エスカレーション手順 [CDN と MediaTailor の統合のサポートとトラブルシューティングのヘルプを取得する](#) について「」を参照してください。

CDN 統合テスト手順

MediaTailor CDN 統合を本番環境にデプロイする前に、適切なテストが不可欠です。これらのテスト手順は、さまざまなデバイスやプラットフォームの設定の問題、パフォーマンスの問題、互換性の問題を特定するのに役立ちます。

基本的な統合検証

CDN 統合が正しく機能していることを確認するには、以下の基本的なテストを実行します。

1. テストマニフェスト配信:

- CDN を介してマニフェストをリクエストし、有効なレスポンスが返されることを確認する
- マニフェストに予想されるコンテンツと広告挿入ポイントが含まれていることを確認する
- マニフェスト URLs オリジンではなく CDN ドメインを使用していることを確認する
- HLS または DASH 検証ツールを使用してマニフェスト構文を検証する

2. URL の書き換えを確認します。

- マニフェスト内のコンテンツセグメント URLs が CDN ドメインを指していることを確認する
- 広告セグメント URLs CDN ドメインを指していることを確認する
- すべての相対 URLs が適切に解決されていることを確認します。

3. コンテンツの再生をテストする:

- ビデオプレイヤーを介してコンテンツを再生し、スムーズ再生を検証する
- コンテンツと広告の両方が中断なく再生されることを確認する
- コンテンツと広告の適切な遷移を確認する
- 検索機能とスクラブ機能をテストする

4. CDN ルーティングを検証します。

- CDN アクセスログをモニタリングして、リクエストが正しくルーティングされていることを確認します。
- キャッシュヒット/ミスパターンが想定どおりであることを確認する
- オリジンリクエストがキャッシュミスに対してのみ発生することを確認する

高度なテスト手順

包括的な検証のために、以下の追加テストを実行します。

1. クロスプラットフォーム互換性テスト:

- 複数のデバイス (デスクトップ、モバイル、タブレット、スマートテレビ) でテストする
 - さまざまなブラウザ間の互換性を検証する
 - さまざまなビデオプレイヤー (HLS.js、Video.js、ネイティブプレイヤー) でテストする
 - さまざまなオペレーティングシステムで検証する
2. パフォーマンステスト:
- マニフェストリクエストの応答時間を測定する (ターゲット: キャッシュの場合は <100ms)
 - さまざまなビットレートでセグメントのロードパフォーマンスをテストする
 - 起動時間がパフォーマンス目標を満たしていることを確認する
 - さまざまなネットワーク条件でテストする
3. 広告追跡の検証:
- 広告追跡ビーコンが正しく起動することを確認する
 - 広告分析データの精度を確認する
 - テストインプレッションと完了の追跡
 - クリックスルー機能を検証する
4. エラー条件テスト:
- オリジンが一時的に使用できない場合のテスト動作
 - 不正な形式のリクエストの正常な処理を検証する
 - CDN フェイルオーバーシナリオをテストする
 - エラーメッセージの明確性と有用性を検証する

テスト環境を作成する

包括的な検証のために、本番稼働用設定をミラーリングするテスト環境を設定します。

1. テスト用に個別の CDN ディストリビューションを設定します。
 - 本番環境と同じキャッシュ動作でテスト CDN ディストリビューションを作成する
 - 本番稼働用セットアップを反映するテストオリジンを設定する
 - 個別のドメイン名を使用して、本番トラフィックとの競合を回避する
2. テスト MediaTailor 設定を作成します。
 - 本番環境と同じ設定でテスト再生設定をセットアップする
 - テスト広告決定サーバーエンドポイントを設定する
 - 本番広告形式に一致するテスト広告コンテンツを使用する

3. 体系的なテストプロセスを実装します。

- 設定変更のテストチェックリストを作成する
- チームのテスト手順を文書化する
- 可能な場合は自動テストを設定する

複数のシナリオでテストする

さまざまなシナリオや条件にわたって統合を検証し、包括的なカバレッジを確保します。

1. 複数のプレイヤータイプとデバイスでテストします。

- さまざまなビデオプレイヤー (ウェブ、モバイル、コネクテッド TV) でテストする
- さまざまなオペレーティングシステムとブラウザで検証する
- さまざまなネットワーク条件と接続速度でテストする

2. 自動テストスクリプトを作成します。

- マニフェストリクエストの検証を自動化する
- 広告挿入シナリオをテストするスクリプトを作成する
- トラフィックの多いシナリオのパフォーマンステストを実装する

3. 広告ターゲティングとパーソナライゼーションを検証する:

- さまざまなユーザープロファイルとターゲティングパラメータを使用してテストする
- 広告決定サーバー統合を検証する
- 広告が利用できない場合のフォールバックシナリオのテスト

ツールと手法のテスト

効果的なテストには、以下のツールと手法を使用します。

ブラウザ開発者ツール

ネットワークタブを使用して HTTP リクエストとレスポンスを検査する

コンソールで JavaScript エラーと警告をモニタリングする

レスポンスヘッダーとキャッシュ動作を検証する

パフォーマンス分析のタイミング情報を確認する

コマンドラインテスト

curl を使用して特定の URLs テストし、ヘッダーを検査します。

```
curl -I "https://your-cdn-domain.com/path/to/manifest.m3u8"
```

ダウンロードテストとタイミング分析に wget を使用する

より読みやすい HTTP テストのために httpie などのツールを採用する

ビデオプレイヤーのテスト

複数のプレイヤー実装でテストする

プレイヤーデバッグモードを使用して内部動作を検査する

プレイヤーイベントとエラーコールバックをモニタリングする

適応ビットレートの切り替え動作を検証する

CDN の分析とモニタリング

テスト中にリアルタイムの CDN メトリクスをモニタリングする

リクエストパターンのアクセスログを確認する

利用可能な場合は CDN 固有のテストツールを使用する

テスト検証用の一時的なアラートを設定する

その他の包括的なテスト方法と体系的な検証アプローチについては、「」を参照してください [CDN と MediaTailor の統合のテストと検証](#)。

MediaTailor の CDN マニフェスト 404 エラーのトラブルシューティング

AWS Elemental MediaTailor コンテンツ配信ネットワーク (CDN) マニフェスト 404 エラーは、再生の開始を妨げる一般的な統合の問題です。このセクションでは、マニフェスト配信の失敗を step-by-step トラブルシューティングします。

多変量プレイリスト、メディアプレイリスト、または MPD リクエストが 404 エラーを返す

クイック修正 (最初に試す):

1. URL の MediaTailor 設定名が完全に一致することを確認します (大文字と小文字が区別されます)

2. CDN なしで MediaTailor に対してマニフェスト URL を直接テストします。 `curl -v "https://your-emt-endpoint.mediatailor.region.amazonaws.com/v1/master/hls/config-name/master.m3u8"`
3. 直接テストが機能する場合は、CDN ルーティングルールでマニフェストリクエストを確認します。

クイック修正が機能しない場合:

症状: プレイヤーは再生の開始に失敗し、マニフェストリクエストは CDN ログに HTTP 404 エラーを返します。

エラーメッセージの例:

- ブラウザコンソール: "Failed to load resource: the server responded with a status of 404 (Not Found)"
- プレイヤーエラー: "MANIFEST_LOAD_ERROR"または "NETWORK_ERROR"
- CDN ログ: GET /v1/master/hls/example-config/master.m3u8 404

解決策:

マルチバリエーションプレイリスト、メディアプレイリスト、MPD リクエストを MediaTailor に転送するように CDN ルーティングルールが正しく設定されていることを確認します。

MediaTailor 設定が存在し、正しく設定されていることを確認します。

CDN 動作パターンが予想されるマニフェストリクエストパス (、 など*.mpd) *.m3u8と一致することを確認します。

MediaTailor の CDN マニフェスト配信の問題とエラーを診断する

AWS Elemental MediaTailor コンテンツ配信ネットワーク (CDN) マニフェスト配信の問題により、広告の適切な挿入と再生が妨げられる可能性があります。視聴者が多変数プレイリスト、メディアプレイリスト、または MPDs で不正確または一貫性のない広告を受け取った場合:

1. キャッシュされたマニフェストを確認します。
 - すべてのマルチバリエーションプレイリスト、メディアプレイリスト、MPD パスの TTL 設定が 0 に設定されていることを確認します。
 - TTL 設定にもかかわらず、CDN が多変数プレイリスト、メディアプレイリスト、または MPDs キャッシュしていないことを確認します。

- CDN ログのキャッシュステータスを確認する - マニフェストリクエストには、ではなく RefreshHitMiss または が表示されます Hit
2. CDN ルーティング設定を確認します。
 - マニフェストリクエストが、オリジンからキャッシュまたは配信されるのではなく、MediaTailor エンドポイントにルーティングされていることを確認します。
 - CDN 動作パターンがマニフェストパス (*.m3u8、*.mpd) と正しく一致することを確認します。
 - クエリパラメータがパーソナライズのために MediaTailor に転送されていることを確認する
 - MediaTailor に対してマニフェスト URLs を直接テストして CDN とサービスの問題を切り分ける
 3. ヘッダー転送設定を確認します。
 - 必要なヘッダーが転送されていることを確認する (「」を参照 [MediaTailor CDN 統合に必要なヘッダー](#))
 - User-Agent ヘッダーがデバイス固有の広告ターゲティングに転送されることを確認する
 - X-Forwarded-For ヘッダーがジオターゲティングに転送されていることを確認します
 - Accept-Encoding ヘッダーが圧縮サポートのために転送されていることを確認する
 4. マニフェストの内容と構造を検証します。
 - マニフェストに予想される広告挿入マーカが含まれていることを確認する (HLS の場合は EXT-X-CUE-OUT/IN)
 - マニフェストのセグメント URLs オリジンドメインではなく CDN ドメインを使用していることを確認する
 - 広告セグメントが正しく挿入され、アクセス可能であることを確認する
 - 検証ツールを使用してマニフェスト構文をテストする (HLS の場合は ffprobe、DASH の場合は mp4box)
 5. さまざまなシナリオをテストします。
 - さまざまなセッション IDs でテストして、パーソナライゼーションが機能していることを検証する
 - さまざまな地理的場所からテストしてジオターゲティングを検証する
 - さまざまな User-Agent 文字列を使用してテストし、デバイスのターゲット設定を検証する
 - CDN の有無にかかわらずマニフェストレスポンスを比較して違いを特定する

その他のトラブルシューティングリソース:

- CDN キャッシュ設定の詳細については、「」を参照してください。 [CDN と MediaTailor の統合のキャッシュ最適化](#)
- 包括的な CDN ルーティング設定については、「」を参照してください。 [MediaTailor の CDN ルーティング動作を設定する](#)
- ヘッダー転送の要件については、「」を参照してください。 [MediaTailor CDN 統合に必要なヘッダー](#)
- ログ分析とエラーコードについては、「」を参照してください。 [MediaTailor の CDN 統合ログ分析とエラーコードリファレンス](#)
- テスト手順と検証については、「」を参照してください。 [CDN と MediaTailor の統合のテストと検証](#)

成功基準: 解決すると、プレイヤーは正常に再生を開始し、広告が期待どおりに表示されるはずです。マニフェストリクエストは CDN ログで HTTP 200 ステータスコードを返し、マニフェストには適切にパーソナライズされた広告コンテンツを含める必要があります。

MediaTailor の CDN セグメントの配信とロードに関する問題のトラブルシューティング

AWS Elemental MediaTailor コンテンツ配信ネットワーク (CDN) セグメント配信の問題により、バッファリングと再生が中断される可能性があります。プレイヤーがセグメントをロードできない場合、またはバッファリングを経験できない場合:

1. CDN ルーティングルールを確認します。

- コンテンツセグメントが正しいオリジンにルーティングされていることを確認する
- 広告セグメントが正しい MediaTailor 広告ストレージロケーションにルーティングされていることを確認します。
- セグメントファイルの拡張子が CDN 動作パターンと一致することを確認する
- マニフェストURLs が正しい CDN ドメインを使用していることを確認する
- ルーティングと動作のパスパターンを設定する詳細な手順については、「」を参照してください。 [MediaTailor の CDN ルーティング動作を設定する](#)

2. CORS 設定を確認します。

- ウェブプレイヤーの場合は、CDN が CORS ヘッダーを通過しているか、適切に設定されていることを確認します。
- ブラウザ開発者ツールを使用してテストし、CORS 関連のエラーを特定する

- プリフライト OPTIONS リクエストが正しく処理されていることを確認する
3. セグメントのアクセシビリティとパフォーマンスをテストします。
 - 個々のセグメント URLs を直接テストしてアクセス可能であることを確認する
 - セグメントの応答時間を確認し、パフォーマンスのボトルネックを特定する
 - セグメントファイルのサイズが帯域幅ターゲットに適していることを確認する
 - さまざまな地理的場所からのセグメントロードをテストする
 4. セグメントの CDN キャッシュ動作を検証します。
 - コンテンツセグメントに適切な TTL 設定がある (通常はマニフェストより長い) ことを確認する
 - 広告セグメントがパーソナライゼーション要件に基づいて適切にキャッシュされていることを確認する
 - コンテンツセグメントと広告セグメントの両方のキャッシュヒット率をモニタリングする
 - キャッシュキーにキャッシュ効率を低下させる不要なパラメータが含まれていないことを確認する
 5. オリジンサーバーの接続性と状態を確認します。
 - オリジンサーバーがセグメントリクエストに正しく応答していることを確認する
 - ロード中のオリジンサーバーの容量と応答時間を確認する
 - オリジンサーバーに期待されるセグメントファイルが使用可能であることを確認する
 - 複数のオリジンが設定されている場合のオリジンフェイルオーバーシナリオのテスト
 6. 広告セグメント固有の問題をトラブルシューティングします。
 - 広告セグメントが適切にトランスコードされ、MediaTailor で使用できることを確認する
 - マニフェストで広告セグメント URLs が正しく生成されていることを確認します。
 - さまざまな広告ターゲティングパラメータを使用して広告セグメントのロードをテストする
 - セグメントが利用できなくなる可能性のある広告トランスコーディングの遅延をモニタリングする
 7. プレイヤーの互換性と動作を検証します。
 - さまざまなプレイヤータイプとバージョンでセグメントのロードをテストする
 - プレイヤーバッファ設定とセグメントリクエストパターンを確認する
 - 失敗したセグメントリクエストのプレイヤーエラー処理を検証する
 - 適応ビットレート切り替えとセグメント選択ロジックをテストする

- CDN のルーティングと動作の設定については、「」を参照してください。 [MediaTailor の CDN ルーティング動作を設定する](#)
- CDN キャッシュの最適化については、「」を参照してください。 [CDN と MediaTailor の統合のキャッシュ最適化](#)
- CORS 設定ガイドンスについては、「」を参照してください。 [MediaTailor の CDN 統合セキュリティのベストプラクティス](#)
- パフォーマンスのモニタリングと分析については、「」を参照してください。 [MediaTailor CDN オペレーションとパフォーマンスのモニタリング](#)
- 包括的なテスト手順については、「」を参照してください。 [CDN と MediaTailor の統合のテストと検証](#)
- ログ分析とエラー診断については、「」を参照してください。 [MediaTailor の CDN 統合ログ分析とエラーコードリファレンス](#)

成功基準: 解決されると、プレイヤーは中断をバッファリングせずにセグメントをスムーズにロードする必要があります。セグメントリクエストは、適切な応答時間で HTTP 200 ステータスコードを返し、コンテンツセグメントと広告セグメントの両方にアクセスして適切にキャッシュする必要があります。

MediaTailor の CDN セッション管理と追跡の問題を修正

AWS Elemental MediaTailor コンテンツ配信ネットワーク (CDN) セッション管理は、広告のパーソナライゼーションと追跡を適切に行うために不可欠です。セッション関連のエラーが発生したり、リクエスト間で動作に一貫性がない場合:

1. セッション ID の整合性を確認します。
 - プレイヤーが 1 回の再生セッションのすべてのリクエストで同じセッション ID を維持していることを検証する
 - CDN ログをチェックして、セッション IDs が正しく転送されていることを確認します。
 - セッション IDs クエリパラメータで適切に URL エンコードされていることを確認します。
 - CloudWatch Logs を使用して、リクエスト間でセッション ID の整合性を検証する (以下の検証ステップを参照)
2. セッションの初期化を検証する:
 - 最初のマニフェストリクエストが正常にセッションを作成することを確認する
 - セッションパラメータの適切な転送を確認する (例: `aws.sessionId`)

- デバッグログを使用してセッションの初期化を検証する (以下のデバッグログの設定を参照)
3. 詳細なセッショントラブルシューティングのためにデバッグログ記録を有効にします。
- サーバー側のレポートの場合: 再生リクエスト?aws.logMode=DEBUGに を追加します。

```
GET <mediatailorURL>/v1/master/<hashed-account-id>/<origin-id>/<asset-id>?
aws.logMode=DEBUG
```

- クライアント側のレポートの場合: セッション初期化リクエスト本文"logMode": "DEBUG"に含める
 - 重要: DEBUG値は大文字と小文字が区別されます
 - 最大 10 のアクティブなデバッグセッションを同時に許可
4. CloudWatch Logs クエリを使用してセッション動作を検証します。
- デバッグセッションがアクティブであることを確認します。

```
fields @timestamp, @message
| filter sessionId = "your-session-id-here"
| filter eventType = "SESSION_INITIALIZED" # client-side reporting
or mediaTailorPath like "/v1/master" # server-side reporting HLS
or mediaTailorPath like "/v1/dash" # server-side reporting DASH
```

- セッションのすべてのイベントを表示します。

```
fields @timestamp, @message, eventType, mediaTailorPath
| filter sessionId = "your-session-id-here"
| sort @timestamp asc
```

- セッションのマニフェスト生成を確認します。

```
fields @timestamp, responseBody, @message
| filter mediaTailorPath like "/v1/master/" and eventType = "GENERATED_MANIFEST"
and sessionId = "your-session-id-here"
```

5. CDN を介したセッションパラメータ転送をテストします。
- MediaTailor に対してセッションパラメータを使用してマニフェストリクエストを直接テストする (CDN をバイパス)
 - CDN の有無にかかわらずセッション動作を比較して転送の問題を特定する
 - CDN クエリパラメータ転送設定にセッション関連のパラメータが含まれていることを確認する
 - CDN がセッション固有のレスポンスをキャッシュしないことを確認する

一般的なセッションエラーメッセージ:

- `ConflictException` (HTTP 409) - 同じセッションに対する複数の同時プレイリストリクエスト。解決策: HLS 仕様に従ってプレイヤーがプレイリストを一度に 1 つずつリクエストすることを確認する
- `NotFoundException` (HTTP 404) - セッションが使用できないか、設定が存在しません。解決策: 設定の有効性を確認し、セッションを再初期化する
- `BadRequestException` (HTTP 400) - 無効なセッション ID または不適切な形式のリクエスト。解決策: リクエストのフォーマットとセッション ID の有効性を検証する

その他のトラブルシューティングリソース:

- デバッグログの設定とフィールドリファレンスの詳細については、「」を参照してください。[AWS Elemental MediaTailor デバッグログの生成](#)
- CloudWatch Logs クエリの例とログ分析については、「」を参照してください。[Amazon CloudWatch Logs に直接 AWS Elemental MediaTailor ログを書き込む](#)
- CDN クエリパラメータ転送設定については、「」を参照してください。[MediaTailor の CDN ルーティング動作を設定する](#)
- 包括的なエラーコードリファレンスについては、「」を参照してください。[MediaTailor からの再生のトラブルシューティング](#)

成功基準: 解決されると、セッションは正しく初期化され、リクエスト間で一貫したセッション IDs が維持され、デバッグログにはエラーなしで適切なSESSION_INITIALIZEDイベントとマニフェスト生成が表示されます。

MediaTailor の CDN 広告時間枠のタイミングと同期の問題を解決する

AWS Elemental MediaTailor コンテンツ配信ネットワーク (CDN) 広告時間枠のタイミングは、コンテンツマーカと正確に同期する必要があります。広告が間違ったタイミングで表示される場合、または広告時間枠のタイミングに一貫性がない場合:

1. コンテンツ内の広告ブレイクマーカを確認します。
 - SCTE-35 マーカがオリジンコンテンツに正しく配置されていることを確認します。
 - 広告時間枠の期間が実際の広告コンテンツの長さと一致することを確認する
 - 広告時間枠のタイミングがコンテンツ境界と一致していることを確認する
 - オリジンマニフェストで SCTE-35 マーカの形式とタイミングの精度を検証する

- さまざまなコンテンツタイプ (ライブと VOD) で広告ブレイクマーカートをテストする
2. CDN キャッシュがタイミングに与える影響を確認します。
 - タイミングドリフトを防ぐために、マニフェスト TTL が 0 に設定されていることを確認します。
 - 時間的制約のあるパラメータがキャッシュされていないことを確認する
 - コンテンツソース、MediaTailor、CDN 間のクロック同期の問題を確認する
 - 長時間実行されるライブストリームのタイミングドリフトをモニタリングする
 - CDN エッジサーバーと NTP の時刻同期を検証する
 3. SCTE-35 マーカーの実装を検証します。
 - EXT-X-DATERANGE タグに適切な SCTE35-OUT および DURATION 仕様が含まれていることを確認する
 - 明示的なキューインタイミングを使用する場合は、SCTE35-OUT マーカーと SCTE35-IN マーカーのペアを確認します。
 - START-DATE タイムスタンプが実際のコンテンツのタイミングと一致することを確認する
 - さまざまな SCTE-35 マーカー形式 (期間ベースマーカーとペアマーカー) をテストする
 4. さまざまなシナリオで広告時間枠のタイミングをテストします。
 - 広告時間枠のタイミングと直接 MediaTailor リクエストと CDN リクエストを比較する
 - さまざまな CDN エッジロケーション間でタイミングの一貫性をテストする
 - さまざまなプレイヤータイプとバッファ動作で広告時間枠のタイミングを検証する
 - ピークトラフィック期間中のタイミング精度をモニタリングする
 5. ログとモニタリングを使用してタイミングの問題をデバッグします。
 - デバッグログを有効にして広告時間枠処理のタイミングを追跡する
 - 広告挿入タイミングパターンの CloudWatch メトリクスをモニタリングする
 - CDN ログでタイミング関連のリクエストパターンを確認する
 - プレイヤーデバッグツールを使用して、クライアントの観点から広告時間枠のタイミングを検証する

予想されるタイミング許容値:

- 広告時間枠のタイミングは、コンテンツ内の SCTE-35 マーカーと一致する必要があります
- 広告期間は、広告決定サーバーのレスポンスで指定された期間と一致する必要があります
- コンテンツソース、MediaTailor、CDN 間のクロック同期は 1 秒以内である必要があります

- SCTE-35 マーカーのタイミングは、実際のコンテンツタイミングから 100 ミリ秒以内の精度である必要があります

その他のトラブルシューティングリソース:

- SCTE-35 マーカーの形式と実装については、「」を参照してください。 [MediaTailor 広告挿入用のコンテンツソースの統合](#)
- デバッグログの設定とタイミング分析については、「」を参照してください。 [AWS Elemental MediaTailor デバッグログの生成](#)
- CDN キャッシュの設定とタイミングへの影響については、「」を参照してください。 [CDN と MediaTailor の統合のキャッシュ最適化](#)
- タイミング検証を含む包括的なテスト手順については、「」を参照してください。 [CDN と MediaTailor の統合のテストと検証](#)
- 広告挿入のタイミングとパフォーマンスのモニタリングについては、「」を参照してください。 [MediaTailor CDN オペレーションとパフォーマンスのモニタリング](#)

成功基準: 解決すると、広告ブレイクは SCTE-35 マーカーで指定された時刻に正確に表示され、すべての CDN エッジロケーションとプレイヤータイプで一貫したタイミングになります。デバッグログには、ドリフトや同期エラーなしで正確な広告時間枠処理のタイミングが表示されます。

MediaTailor の CDN パフォーマンスを最適化し、レイテンシーの問題を解決する

AWS Elemental MediaTailor コンテンツ配信ネットワーク (CDN) の統合パフォーマンスは、視聴者のエクスペリエンスと広告配信品質に直接影響します。応答時間が遅い、またはパフォーマンスが低下した場合:

パフォーマンス測定手法

パフォーマンスの問題をトラブルシューティングする前に、ベースライン測定値と継続的なモニタリングを確立します。

1. 主要なパフォーマンスメトリクスを測定します。

- 応答時間: マニフェストリクエストは 200 ミリ秒以内に完了し、セグメントリクエストは 100 ミリ秒以内に完了する必要があります
- キャッシュヒット率: コンテンツセグメント >95%、広告セグメント >90%

- オリジンリクエストボリューム: キャッシュが最適化されている場合、合計リクエストの 5% 未満である必要があります
- 最初のフレームまでの時間: 初期再生は 2~3 秒以内に開始する必要があります

2. パフォーマンス測定ツールを使用します。

- CDN 分析ダッシュボード: キャッシュパフォーマンス、応答時間、エラー率のモニタリング
- CloudWatch メトリクス: GetManifest.Latency を含む MediaTailor サービスメトリクスを追跡する
- ブラウザ開発者ツール: クライアント側のパフォーマンスとネットワークタイミングを測定する
- コマンドラインツール: curl とタイミングオプションを使用して特定のリクエストを測定する

3. 継続的モニタリングを実装します。

- 応答時間の短縮に関する自動パフォーマンスアラートを設定する
- さまざまな地理的リージョンでパフォーマンスをモニタリングする
- ピークトラフィック期間中のパフォーマンスを追跡する
- 設定変更の前後にパフォーマンスメトリクスを比較する

パフォーマンス測定リソース:

- 包括的なパフォーマンスモニタリングの設定については、「」を参照してください。 [MediaTailor CDN オペレーションとパフォーマンスのモニタリング](#)
- パフォーマンステストの手順については、「」を参照してください。 [CDN と MediaTailor の統合のテストと検証](#)
- CloudWatch メトリクスとモニタリングについては、「」を参照してください。 [Amazon CloudWatch メトリクス AWS Elemental MediaTailor によるモニタリング](#)

CDN キャッシュのパフォーマンスの問題

キャッシュパフォーマンスの問題は、最も一般的な CDN 統合の問題の 1 つです。これらの問題はすべての MediaTailor 実装に影響し、視聴者のエクスペリエンスとコストに大きな影響を与える可能性があります。

キャッシュヒット率が低い

症状: オリジンリクエスト量の増加、レイテンシーの増加、帯域幅コストの増加、視聴者のエクスペリエンスの低下

ターゲット値:

- コンテンツセグメント: 95% 以上のキャッシュヒット率
- 広告セグメント: 90% 以上のキャッシュヒット率
- マニフェスト: 実装によって異なります (パーソナライズされたマニフェストはキャッシュしないでください)

一般的な原因:

- 異なるコンテンツタイプの TTL 設定が正しくない
- キャッシュキー設定に不要なクエリパラメータが含まれている
- オリジンからのキャッシュコントロールヘッダーが正しく設定されていない
- 頻繁なキャッシュの無効化または消去
- 地理的分布の問題 (コンテンツがエッジロケーションにキャッシュされない)

解決方法:

1. TTL 設定を確認して最適化します。

- コンテンツセグメント: セグメント期間以上に TTL を設定する
- 広告セグメント: 再利用可能な広告の TTL を 24 時間以上に設定する
- 静的アセット: TTL を 24 時間以上に設定する

包括的な TTL の推奨事項とキャッシュ最適化戦略については、「」を参照してください[CDN と MediaTailor の統合のキャッシュ最適化](#)。

2. キャッシュキー設定を最適化します。

- キャッシュキーから不要なクエリパラメータを削除する
- コンテンツに影響するパラメータのみが含まれていることを確認する
- パラメータの順序と大文字と小文字の区別を正規化する

3. オリジンキャッシュコントロールヘッダーが正しく設定されていることを確認する

4. トラフィックの多い実装には、オリジンシールド (または同等の CDN 機能) を実装します。

オリジンシールド機能は主要な CDN で使用できますが、名前が異なる場合があります (CloudFront Origin Shield、Fastly Shield、Cloudflare Argo Tiered Cache など)。CDN でこの機能が提供されていない場合は、[AWS サポート](#)に問い合わせるときに MediaTailor で有効にできます。

5. キャッシュ無効化戦略を確認し、不要なページを減らす

検証ステップ:

1. CDN 分析ダッシュボードを使用してキャッシュヒット率をモニタリングする
2. curl で特定の URLs をテストしてキャッシュヘッダーを検証する
3. 変更前と変更後のオリジンリクエストボリュームを比較する

高オリジンリクエストボリューム

症状: MediaTailor オリジンに到達するリクエストの予期しない数の増加、オリジンサーバーの負荷の増加、コストの増加

予想されるパターン: キャッシュヒット率が最適である場合、オリジンリクエストはビューワーリクエスト全体の 5% 未満である必要があります

一般的な原因:

- TTL 値が低いためキャッシュミスが発生する
- キャッシュキーの断片化 (一意のキャッシュキーが多すぎます)
- キャッシュされたコンテンツがないリージョンでの地理的トラフィックの急増
- 頻繁なキャッシュの無効化

解決方法:

1. リクエストパターンを分析してキャッシュミスの原因を特定する
2. コンテンツタイプと更新頻度に基づいて TTL 設定を最適化する
3. 新しいコンテンツにキャッシュウォーミング戦略を実装する
4. オリジンシールドの実装を検討する (異なる名前のメジャー CDN 「」を参照) [Origin Shield の実装](#)

アラートしきい値: オリジンリクエストが合計リクエストの 10% を超えた場合、またはベースラインよりも 50% 増加した場合にアラートを設定する

一般的な HTTP エラーの解決

CDN 統合の HTTP エラーは、多くの場合、設定の問題やサービスの問題を示します。これらのエラーパターンは、すべての MediaTailor 実装で一貫しています。

404 Not Found エラー

症状: マニフェストまたはセグメントリクエストが HTTP 404 を返し、プレイヤーがコンテンツをロードできない、プレイヤーログに「MANIFEST_LOAD_ERROR」

一般的な原因:

- CDN オリジン設定が正しくない (MediaTailor エンドポイント URL が正しくない)
- キャッシュ動作パスパターンがないか、正しくない
- CDN 設定での URL の書き換えの問題
- MediaTailor 設定名または再生エンドポイントエラー
- ライブコンテンツのタイミングの問題 (将来のセグメントのリクエスト)

診断手順:

1. MediaTailor オリジンに対して同じ URL を直接テストする (CDN をバイパス)
2. CDN オリジン設定が MediaTailor 再生エンドポイントと一致することを確認する
3. CDN キャッシュ動作パスのパターンと優先順位を確認する
4. リクエストルーティングの詳細について CDN アクセスログを確認する
5. MediaTailor 設定名とリージョン設定を検証する

解決方法:

- MediaTailor 再生エンドポイントと一致するように CDN オリジン設定を修正する
- キャッシュ動作パスパターンを更新してリクエストを適切にルーティングする
- 該当する場合、URL 書き換えルールを修正
- MediaTailor 設定が存在し、アクティブであることを確認する

403 禁止エラー

症状: リクエストが HTTP 403 を返す、アクセス拒否メッセージ、認証の失敗

一般的な原因:

- MediaTailor に必要なクエリパラメータがないか、正しくない
- CDN が必要なヘッダーまたはパラメータを転送しない
- IP アドレスの制限または地理的ブロック
- 認証トークンの問題 (署名URLs を使用する場合)

解決方法:

- 必要なクエリパラメータがすべて含まれ、転送されていることを確認します。
- ヘッダーおよびパラメータ転送の CDN 設定を確認する

- IP の制限と地理的設定を確認する
- 認証トークンと署名プロセスを検証する

400 不正なリクエストエラー

症状: リクエストが HTTP 400 を返す、不正な形式のリクエストエラー、パラメータ検証の失敗

一般的な原因:

- 不正な形式のクエリパラメータまたは URL エンコードの問題
- 無効なパラメータ値または形式
- 特定の MediaTailor 機能に必要なパラメータがない
- URL の長さの制限を超えました

解決方法:

- クエリパラメータ形式と URL エンコーディングを検証する
- MediaTailor API 要件に対してパラメータ値を確認する
- すべての必須パラメータが含まれていることを確認する
- URL の長さを確認し、パラメータの最適化を検討する

5xx サーバーエラー

症状: リクエストが HTTP 500、502、503、または 504 エラーを返す、断続的なサービス障害

一般的な原因:

- MediaTailor サービスの問題または容量の制限
- CDN オリジン接続の問題
- オリジンレスポンスの遅延によるタイムアウトの問題
- 一時的なサービスの低下

解決方法:

- MediaTailor AWS のサービスステータスのサービスヘルスダッシュボードを確認する
- CDN オリジンの接続とタイムアウトの設定を確認する
- エクスポネンシャルバックオフを使用して再試行ロジックを実装する
- MediaTailor CloudWatch メトリクスをモニタリングしてサービスの状態を確認する
- 問題が解決しない場合は AWS 、サポートにお問い合わせください。

1. ベースラインパフォーマンスを測定します。

- MediaTailor へのマニフェストリクエストの応答時間を直接テストする (ターゲット: <200ms)
- マニフェストリクエストの CDN 応答時間を測定する (ターゲット: キャッシュヒットの場合は <100ms)
- オリジンと CDN の両方からのセグメントのロード時間を確認する

2. CDN のパフォーマンスを分析します。

- コンテンツセグメントのキャッシュヒット率を確認する (ターゲット: 一般的なコンテンツの場合は >80%)
- オリジンシールド (または同等の CDN 機能) が有効で、オリジンと同じ AWS リージョンで設定されていることを確認します。異なる CDN の機能に異なる名前を使用する
- CDN エッジロケーションのパフォーマンスと地理的分散をモニタリングする

パフォーマンスベンチマーク:

- マニフェスト生成の応答時間をモニタリングし、ベースラインパフォーマンスと比較します。
- CDN キャッシュヒットはオリジンリクエストよりも大幅に高速です
- ADS 応答時間がマニフェスト生成の遅延を引き起こさないようにする

その他のトラブルシューティングリソース:

- 包括的なパフォーマンス最適化戦略については、「」を参照してください。 [CDN と MediaTailor の統合のパフォーマンス最適化ガイド](#)
- オリジンシールドの実装の詳細については、「」を参照してください。 [Origin Shield の実装](#)
- CDN キャッシュの最適化については、「」を参照してください。 [CDN と MediaTailor の統合のキャッシュ最適化](#)
- パフォーマンスのモニタリングとメトリクスについては、「」を参照してください。 [MediaTailor CDN オペレーションとパフォーマンスのモニタリング](#)
- パフォーマンステストの手順については、「」を参照してください。 [CDN と MediaTailor の統合のテストと検証](#)

成功基準: 解決すると、応答時間はターゲットベンチマーク (マニフェストが 200 ミリ秒未満、セグメントが 100 ミリ秒未満) を満たし、キャッシュヒット率はほとんどのコンテンツタイプで 90% を超え、オリジンリクエストボリュームは合計リクエストの 5% 未満である必要があります。パフォーマンスは、すべての地理的リージョンとデバイスタイプで一貫している必要があります。

MediaTailor のデバイスとプラットフォーム間で CDN の一貫性のない動作を修正

AWS Elemental MediaTailor コンテンツ配信ネットワーク (CDN) 統合は、すべてのデバイスとプラットフォームで一貫した広告配信を提供する必要があります。広告の動作がデバイス間で異なる場合:

1. すべての CDN 動作にわたって一貫したヘッダー転送を確保します。
 - User-Agent、X-Forwarded-For、およびカスタムターゲティングヘッダーが一貫して転送されていることを確認します。
 - ヘッダー転送ルールが関連するすべてのキャッシュ動作に適用されることを確認する
2. プレイヤーが CDN 設定と互換性があることを確認します。
 - 複数のプレイヤータイプ (HLS.js、Video.js、ネイティブプレイヤー) でテストして、プレイヤー固有の問題を特定する
 - プレイヤー固有のヘッダー要件または URL 処理の違いを確認する
3. 複数のデバイスタイプでテストして、デバイス固有の問題を特定します。
 - テストにモバイルデバイス、タブレット、スマートTVs、デスクトップブラウザを含める
 - さまざまなオペレーティングシステムとブラウザバージョンをテストする
 - デバイス固有の広告ターゲティングが正しく機能することを確認する

これらのトラブルシューティング手順を実行してもサポートが必要な場合は、「」を参照してください [CDN 統合のサポートを受ける](#)。

トラブルシューティングの準備

CDN 統合の問題が発生した場合のトラブルシューティングを簡素化するためのツールとプロセスを設定します。プロアクティブ準備により、問題発生時のトラブルシューティングを迅速かつ効果的に行うことができます。

包括的なログ記録を有効にする

詳細なログは、CDN 統合の問題の診断に不可欠です。トラブルシューティング中に必要な情報をキャプチャするようにログ記録を設定します。

1. 詳細な CDN アクセスログを有効にします。
 - MediaTailor リクエストを処理するすべてのキャッシュ動作のログ記録を設定する

- ログエントリにクエリ文字列とカスタムヘッダーを含める
- パターンと異常を特定するためのログ分析ツールを設定する
- ライブイベント中に問題をすぐに検出するためのリアルタイムログを有効にする
- 履歴トラブルシューティングデータを維持するようにログ保持ポリシーを設定する

2. MediaTailor ログ記録を設定します。

- MediaTailor 設定のアクセスログを有効にする
- 一元的なログ管理のために CloudWatch ロググループを設定する
- エラーパターンを識別するようにログフィルターを設定する

3. オリジンサーバーのログ記録を設定します。

- コンテンツオリジンサーバーで詳細なアクセスログを有効にする
- ログにリクエストヘッダーとレスポンスコードを含める
- オリジンサーバーのパフォーマンスメトリクスをモニタリングする

診断リクエストヘッダーを追加する

カスタムヘッダーは、CDN を介してリクエストを追跡し、ルーティングの問題を特定するのに役立ちます。

1. CDN 診断ヘッダーを設定します。

- 各リクエストに一意的識別子を追加する (例: X-Request-ID)
- リクエストヘッダーに CDN 固有の情報を含める
- エッジロケーションまたは POP (ポイントオブプレゼンス) 情報を追加して、地理的ルーティングを追跡する
- キャッシュ動作分析にキャッシュステータスヘッダー (Hit、Miss、RefreshHit) を含める

2. デバッグ用のレスポンスヘッダーを追加します。

- サーバー識別ヘッダーを含める
- パフォーマンス分析のタイミング情報を追加する
- マニフェストリクエストのキャッシュコントロールヘッダーを含める

ベースラインパフォーマンスメトリクスを確立する

~~通常のパフォーマンス範囲を文書化して、トラブルシューティング中に異常をすばやく特定します。~~

1. ベースラインメトリクスを記録します。
 - さまざまなコンテンツタイプのキャッシュヒット率
 - 応答時間のパーセンタイル (P50、P95、P99)
 - ステータスコード別のエラー率
 - 時刻別のリクエストボリュームパターン
2. 期待されるパフォーマンスを文書化します。
 - ターゲットキャッシュヒット率 (コンテンツの場合は 95% 以上、広告の場合は 90% 以上)
 - 許容応答時間 (<100ms キャッシュ、<500ms オリジン)
 - 最大許容エラー率 (4xx の場合は <1%、5xx の場合は <0.1%)
3. パフォーマンスダッシュボードの作成: ベースライン値と比較した現在のメトリクスを表示するモニタリングダッシュボードを設定します。

トラブルシューティングツールを準備する

効果的なトラブルシューティングに必要なツールとアクセス許可を設定します。

1. コマンドラインツール:
 - curl HTTP リクエストとレスポンスをテストするための
 - dig DNS トラブルシューティング nslookup 用の または
 - マニフェスト検証用の HLS/DASH 検証ツール
 - ログ分析ツール (grep、awk、または特殊なログアナライザー)
2. アクセス許可:
 - 設定レビュー用の CDN 管理コンソールアクセス
 - 設定検証のための MediaTailor コンソールアクセス
 - メトリクスとログ分析のための CloudWatch アクセス
 - バックエンドのトラブルシューティングのためのオリジンサーバーアクセス
3. ドキュメント:
 - ネットワークアーキテクチャ図
 - CDN と MediaTailor の設定ドキュメント
 - エスカレーション手順の連絡先情報

ワークフロー固有のトラブルシューティングガイド

このユニバーサルトラブルシューティングガイドでは、すべての MediaTailor CDN 統合における一般的な問題について説明します。特定のワークフローまたはサービスに固有の問題については、以下の特殊なトラブルシューティングリソースを参照してください。

サーバー側の広告挿入 (SSAI) のトラブルシューティング

広告挿入の失敗、広告移行の問題、パーソナライゼーションの問題など、SSAI 固有の問題については、ワークフロー固有の SSAI トラブルシューティングドキュメントを参照してください。

SSAI 固有の一般的な問題:

- 広告挿入の失敗と空の広告ブレイク
- 広告移行のタイミングと同期の問題
- パーソナライズとターゲティングの問題
- 広告の追跡と分析の不一致

チャンネルアセンブリのトラブルシューティング

マニフェスト生成の問題やタイムシフト機能などのチャンネルアセンブリ固有の問題については、「チャンネルアセンブリワークフロードキュメント」を参照してください。

一般的なチャンネルアセンブリの問題:

- マニフェスト生成エラーとコンパイルエラー
- タイムシフトウィンドウと DVR 機能の問題
- ソースコンテンツの可用性とフェイルオーバーの問題
- プログラムスケジュールとメタデータの同期

MediaPackage 統合のトラブルシューティング

マニフェストフィルタリングや EMP エンドポイントの問題など、MediaPackage 固有の問題については、「」を参照してください [CDN 統合のトラブルシューティング](#)。

MediaPackage 統合の一般的な問題:

- マニフェストフィルタリングパラメータエラー
- MediaPackage エンドポイントの接続の問題
- EMP 固有のキャッシュ動作の問題

- MediaPackage オリジン認証の問題

CloudFront 固有のトラブルシューティング

CloudFront 固有の設定とセットアップの問題については、「CloudFront 統合ドキュメント」を参照してください。

CloudFront の一般的な問題:

- ディストリビューション設定とキャッシュ動作の設定
- オリジンアクセスアイデンティティとセキュリティ設定
- CloudFront 固有のエラーコードとレスポンス
- 地理的制限とエッジロケーションの問題

その他のリソース:

- パフォーマンス最適化のガイダンスについては、「」を参照してください。 [CDN パフォーマンスの最適化](#)
- モニタリングとアラートの設定については、「」を参照してください。 [CDN モニタリング](#)
- 一般的なサポートとサポートについては、「」を参照してください。 [CDN 統合のサポートを受ける](#)

MediaTailor の CDN 統合ログ分析とエラーコードリファレンス

AWS Elemental MediaTailor コンテンツ配信ネットワーク (CDN) 統合ログは、パフォーマンスとエラーに関する貴重なインサイトを提供します。このガイドでは、CDN ログ (コンテンツ配信ネットワークから) と、CDN 統合のトラブルシューティングに関連する MediaTailor ログとエラーコードの両方について説明します。このリファレンスは、コンテンツ配信ネットワークログとエラーコードが MediaTailor 統合について何を伝えているかを理解する必要がある場合に使用します。このガイドは、ログエントリとエラーメッセージを解釈して問題の根本原因を特定するのに役立ちます。

関連するトピック:

- 詳細なトラブルシューティング手順についてはstep-by-step 「」を参照してください。 [CDN 統合のトラブルシューティング](#)
- プロアクティブなモニタリングと防止戦略については、「」を参照してください。 [MediaTailor CDN オペレーションとパフォーマンスのモニタリング](#)

- エスカレーションと追加のヘルプについては、「」を参照してください。 [CDN 統合のサポートを受ける](#)
- 包括的な MediaTailor ログ記録設定とリソースについては、「」を参照してください。 [MediaTailor ログ記録設定リソース](#)
- CloudFront ログ形式のリファレンスについては、[CloudFront アクセスログ形式](#)」を参照してください。
- HTTP ステータスコードリファレンスについては、[「HTTP レスポンスステータスコード」](#)を参照してください。

MediaTailor の CDN ログ解釈と分析手法

AWS Elemental MediaTailor コンテンツ配信ネットワーク (CDN) 統合は、リクエストパターンの理解と問題の特定に役立つ詳細なログを生成します。コンテンツ配信ネットワークログを分析するときは、次の主要な指標に焦点を当てます。

注: 次の例では、Amazon CloudFront の用語を使用しています。他の CDN プロバイダーも、同様の概念に異なる用語を使用する場合があります。

HTTP ステータスコード

- 200 - 成功したリクエスト
- 404 - リソースが見つかりません (ルーティングルールと MediaTailor 設定を確認)
- 403 - アクセス拒否 (CDN セキュリティ設定とオリジンアクセス許可を確認する)
- 502/503/504 - オリジンサーバーエラー (MediaTailor サービスのヘルスとオリジン接続を確認)

キャッシュ動作インジケータ

- Hit - CDN キャッシュから提供されるコンテンツ
- Miss - オリジンから取得されたコンテンツ
- RefreshHit - オリジンで検証されたキャッシュされたコンテンツ
- Error - リクエストでエラーが発生しました

モニタリングするリクエストパターン

- マニフェストリクエストでは通常、TTL 設定が低いMissか、または低いRefreshHitことが原因です。
- 人気のあるコンテンツHitにはコンテンツセグメントが表示されます

- パーソナライゼーションMissにより広告セグメントが表示されることがある

サービスヘルスマモニタリング

CDN ログを分析して問題をトラブルシューティングするときは、MediaTailor のサービスの状態をチェックして、問題がサービスに関連しているかどうかを確認します。

AWS サービスヘルスダッシュボード

現在の MediaTailor サービスステータスと進行中のサービスイベントを確認する

アクセス: [AWS サービスヘルスダッシュボード](#)

これを使用する: CDN ログに広範な 5xx エラーまたはサービスタイムアウトが表示される

AWS Personal Health ダッシュボード

アカウント固有のサービスヘルス通知とメンテナンスイベントを表示する

アクセス: [AWS Personal Health Dashboard](#)

これを使用する: アカウント固有のサービスヘルス情報またはメンテナンス通知が必要な場合

Amazon CloudWatch メトリクスのモニタリング

CloudWatch メトリクスを使用して MediaTailor サービスヘルスインジケータをモニタリングします。

- `GetManifest.Errors` - マニフェスト生成エラーを追跡する
- `GetManifest.Latency` - 応答時間のパフォーマンスをモニタリングする
- `AdDecisionServer.Errors` - 広告サーバーの接続問題をモニタリングする
- `Origin.Errors` - オリジンサーバー接続の問題を追跡する

アクセス: [CloudWatch コンソール](#)

これを使用する: 詳細なサービスパフォーマンスメトリクスと履歴の傾向が必要な場合

サービスヘルスのトラブルシューティングワークフロー:

1. Service AWS Health Dashboard で現在のサービスステータスを確認する
2. AWS Personal Health Dashboard でアカウント固有の通知を確認する

3. サービスパフォーマンスインジケータの CloudWatch メトリクスをモニタリングする
4. サービスのヘルスステータスを CDN ログパターンとエラー率と関連付ける
5. サービスの状態が正常である場合は、CDN 設定とオリジンサーバーの問題に焦点を当てます。

CDN ログ分析リソース

- [CloudFront アクセスログ](#) - CDN ログの形式とフィールドに関する完全なガイド
- [CloudFront リアルタイムログ](#) - リアルタイムログのストリーミングと分析
- [CloudWatch Logs Insights を使用したログデータの分析](#) - 高度なログクエリと分析
- [ワークロードリソースのモニタリング](#) - Well-Architected のモニタリングパターン

トピック

- [MediaTailor エラーコードと CDN 統合のトラブルシューティング](#)
- [MediaTailor の CDN ログ分析ツールとモニタリング手法](#)
- [MediaTailor ログ記録設定リソース](#)

MediaTailor エラーコードと CDN 統合のトラブルシューティング

AWS Elemental MediaTailor エラーコードは、コンテンツ配信ネットワーク (CDN) を通過するとき、統合の問題に関する特定の情報を提供します。MediaTailor がコンテンツ配信ネットワークを介してエラーを返す場合、これらのコードは特定の問題を示します。

注: これらのエラーコードは MediaTailor によって返され、CDN に渡されます。CDN 固有のエラーについては、CDN プロバイダーのドキュメントを参照してください。

400 Bad Request

一般的な原因: 不正な形式のクエリパラメータ、必須パラメータの欠落、無効なセッション IDs

チェック: CDN クエリ文字列転送設定、パラメータエンコーディング

403 Forbidden

一般的な原因: 存在しない、またはアクセス許可がないセグメントをリクエストするクライアントプレイヤー。広告セグメントの場合、MediaTailor は存在しないセグメントを指定している可能性があります ([AWS サポート](#) に連絡して調査してください)。コンテンツセグメント、オリジンプロバイダーのアクセス制限、または認証の問題の場合

チェック: 広告セグメントの場合: MediaTailor が無効なセグメント URLs を生成している場合は、[AWS サポート](#)にお問い合わせください。コンテンツセグメントの場合: オリジンサーバーのアクセス許可、認証情報、アクセスポリシーを検証します。リクエストされた URLs にセグメントが存在するかどうかを確認する

404 Not Found

一般的な原因: MediaTailor 設定が存在しないか、非アクティブ、誤った再生 URL パス、存在しないリソースへのマニフェストまたはセグメントリクエスト

チェック: MediaTailor 設定が存在し、アクティブであることを確認する、再生 URL が ManifestEndpointPrefix から と一致することを確認する
GetPlaybackConfiguration、CDN ルーティングルールが正しい MediaTailor エンドポイントにリクエストを転送することを確認する

500 Internal Server Error

一般的な原因: オリジンサーバーの問題、ADS 接続の問題、マニフェスト処理エラー

チェック: オリジンサーバーのヘルス、ADS レスポンスの有効性、MediaTailor サービスのステータス

502 不正なゲートウェイ

一般的な原因: HLS プレイリストのアラインメントの問題 (SCTE マーカーがプレイリスト間で整列されていない、一部のプレイリストに SCTE マーカーがない、ビットレートバリエーション間で広告ブレイクのタイミングが一貫していない) UnsupportedManifestException により、オリジンサーバーに到達できない、無効なオリジンレスポンス

チェック: オリジンサーバー接続とファイアウォールルール、DNS 解決、すべてのビットレートバリエーションにわたる HLS プレイリストの整合性、すべてのプレイリストにわたる同じセグメントでの SCTE-35 マーカーの整列、すべてのプレイリストに必要な広告ブレイクマーカーが含まれていることを確認する

エラーコード分析リソース:

- [MediaTailor のトラブルシューティング](#) - MediaTailor エラーコードリファレンスを完了する
- [CloudFront エラーレスポンスのトラブルシューティング](#) - CDN 固有のエラー分析
- [HTTP レスポンスステータスコード](#) - 包括的な HTTP ステータスコードリファレンス
- [障害管理](#) - Well-Architected エラー処理パターン

MediaTailor の CDN ログ分析ツールとモニタリング手法

AWS Elemental MediaTailor コンテンツ配信ネットワーク (CDN) 統合は、効率的な分析ツールと手法を必要とする大量のログデータを生成します。これらのアプローチを使用して、コンテンツ配信ネットワークと MediaTailor ログを効率的に分析します。

- コマンドライン分析: grep、awk、などのツールを使用してログパターン sort をフィルタリングおよび分析する
- Amazon CloudWatch Logs Insights: 高度な分析のために SQL のような構文で CDN ログと MediaTailor ログをクエリする
- サードパーティー製ツール: 包括的なモニタリングとアラートのためのログ分析プラットフォームを検討する
- カスタムダッシュボード: CDN メトリクスと MediaTailor パフォーマンスデータを組み合わせたビジュアライゼーションを作成する

ログ分析や複雑なエラーパターンの解釈についてさらにサポートが必要な場合は、「」を参照してください [CDN 統合のサポートを受ける](#)。

ログ分析ツールとリソース:

- [CloudWatch Logs Insights](#) - ログ分析用の SQL のようなクエリ
- [Amazon OpenSearch Service](#) - 高度なログ検索と分析
- [CloudWatch ダッシュボード](#) - カスタム可視化とモニタリング
- [運用上の可視性のための分散システムの計測](#) - 高度なオブザーバビリティパターン
- [ワークロードオブザーバビリティの設計](#) - Well-Architected オブザーバビリティガイダンス

MediaTailor ログ記録設定リソース

MediaTailor は、CDN ログに加えて、広告挿入、マニフェスト生成、サービスインタラクションをモニタリングするための包括的なログ記録機能を提供します。これらのリソースを使用して、MediaTailor ログを設定および分析します。

MediaTailor マニフェストとサービスログ

マニフェスト生成、オリジンインタラクション、サービスイベントのログを設定および分析します。

- [AWS Elemental MediaTailor マニフェストログの説明とイベントタイプ](#) - MediaTailor マニフェストログとイベントタイプの完全なリファレンス
- [AWS Elemental MediaTailor ログの表示](#) - MediaTailor ログを表示および解釈するためのガイド

提供されたログ設定

コストを最適化して、複数の宛先への柔軟なログ配信を設定します。

- [提供されたログを使用して AWS Elemental MediaTailor ログを送信する](#) - Amazon S3、Firehose、または CloudWatch Logs へのログ配信を設定する
- [AWS Elemental MediaTailor ログ記録戦略の移行](#) - 既存のログ記録設定の移行ガイド

CloudWatch Logs の統合

MediaTailor ログを CloudWatch と統合してモニタリングと分析を行います。

- [AWS Elemental MediaTailor ログの表示](#) - CloudWatch Logs の設定と分析
- [CloudWatchInsights](#) - 高度なログクエリと分析

広告固有のログ記録

広告挿入のパフォーマンスをモニタリングし、広告関連の問題をトラブルシューティングします。

- ADS インタラクションログ - 広告決定サーバーの通信とエラーをモニタリングする

CDN ログと MediaTailor ログの主な違い:

- CDN ログ: コンテンツ配信ネットワークからのリクエスト/レスポンスパターン、キャッシュ動作、ネットワークレベルのエラーを表示する
- MediaTailor ログ: 広告挿入の詳細、マニフェスト生成イベント、オリジンインタラクション、サービス固有のエラーを表示する
- 複合分析: 両方のログタイプを一緒に使用して、ストリーミングワークフローを完全に可視化します。

CDN と MediaTailor の両方のログ記録を組み合わせた包括的なモニタリングについては、「」を参照してください[MediaTailor CDN オペレーションとパフォーマンスのモニタリング](#)。

で MediaTailor と CDN を自動化する CloudFormation

を使用したコンテンツ配信ネットワーク (CDN) AWS Elemental MediaTailor による自動化により、広告挿入ワークフローが AWS CloudFormation 合理化され、信頼性が向上します。

このセクションでは、AWS CloudFormation (AWS infrastructure as code service) を使用して、コンテンツ配信ネットワーク (CDN) AWS Elemental MediaTailor を自動的にセットアップする方法について説明します。前のセクションで説明したように、この統合を手動で設定できますが、を使用すると、単一のテンプレートでプロセス全体を自動化することで、時間 CloudFormation を節約し、エラーを削減できます。

を初めて使用する場合は CloudFormation、必要なすべての AWS リソースを定義するテンプレートファイルを作成できるサービスです。このテンプレートをデプロイすると、はそれらのリソース CloudFormation を自動的に作成して設定し、それらが正しく連携するようにします。

詳細については CloudFormation、[CloudFormation 「ユーザーガイド」](#) を参照してください。の MediaTailor リソースタイプの詳細については CloudFormation、[AWS 「::MediaTailor リソースタイプのリファレンス」](#) を参照してください。

トピック

- [MediaTailor と CDN の統合 CloudFormation に を使用する理由](#)
- [CDN と MediaTailor の統合の CloudFormation デプロイを準備する](#)
- [CDN と MediaTailor の統合用の CloudFormation テンプレートをデプロイする](#)
- [CDN と MediaTailor の統合に CloudFormation デプロイされたリソースを使用する](#)
- [CDN と MediaTailor の統合に関する CloudFormation デプロイのテストと検証](#)
- [CDN と MediaTailor の統合に関する一般的な CloudFormation デプロイの問題をトラブルシューティングする](#)
- [CDN と MediaTailor の統合の CloudFormation テンプレートをカスタマイズする](#)
- [AWS CloudFormation AWS Elemental MediaTailor および Amazon CloudFront 統合の テンプレートリファレンス](#)

MediaTailor と CDN の統合 CloudFormation に を使用する理由

AWS Elemental MediaTailor による オートメーション AWS CloudFormation は、ストリーミングワークフローを管理するブロードキャストプロフェッショナルに大きな利点をもたらします。MediaTailor をコンテンツ配信ネットワーク (CDN) で手動で設定すると、時間がかかり、エラーが発生しやすくなります。CloudFormation 自動化を使用すると、次の利点があります。

- 整合性: 毎回同じ設定がデプロイされるようにし、人為的なミスを減らします。
- バージョン管理: インフラストラクチャをコードとしてバージョン管理システムに保存し、変更を追跡します。

- 迅速なデプロイ: 手動設定ではなく、複雑な設定を数分でデプロイします。
- 環境レプリケーション: 開発、テスト、本番環境間で設定を簡単にレプリケートできます。
- ドキュメント: テンプレート自体は、インフラストラクチャのドキュメントとして機能します。

自動ワークフローと手動設定の比較は次のとおりです。

手動セットアップ (複数ステップ)	自動セットアップ (単一テンプレート)
MediaTailor 再生設定を作成する	パラメータを使用して 1 つの CloudFormation テンプレートをデプロイする
CloudFront デイストリビューションを作成する	
キャッシュ動作を設定する	
セキュリティ設定をセットアップする	

CloudFront で MediaTailor を設定するための自動ワークフローは、次のステップに従います。

1. コンテンツオリジンと広告サーバーパラメータを使用して CloudFormation テンプレートをデプロイする
2. CloudFormation は、必要なすべてのリソースを作成および設定します。
 - 広告挿入の MediaTailor 再生設定
 - 適切なキャッシュ動作を持つ CloudFront デイストリビューション
 - コンテンツ保護のセキュリティ設定
3. CloudFormation 出力を使用して広告対応ストリーム URLs にアクセスする
4. 動的に挿入された広告でコンテンツをストリーミングする

CDN と MediaTailor の統合の CloudFormation デプロイを準備する

AWS Elemental MediaTailor を使用した デプロイには、特定の前提条件と準備手順 AWS CloudFormation が必要です。を使用して MediaTailor と Amazon CloudFront CloudFormation を統合する前に、以下があることを確認してください。

- MediaTailor、CloudFront、および CloudFormation リソースを作成するアクセス許可を持つ AWS アカウント
- 動画コンテンツがホストされているコンテンツオリジン (AWS Elemental MediaPackage Amazon S3 や別のオリジンサーバーなど)
- VAST リクエストに応答できる広告決定サーバー (ADS)

CloudFormation テンプレートをデプロイする前に、以下の必須パラメータを収集します。

AdServerUrl

動的広告挿入用の VAST 広告サーバーの URL。テスト用に静的 VAST エンドポイントが用意されています。

ContentOriginDomainName

プロトコルを使用しないコンテンツオリジンのドメイン名 (、*mediapackage-domain.mediapackagev2.us-west-2.amazonaws.com*、*bucket.s3.amazonaws.com*、*custom-origin.example.com* など)。http:// または https:// プレフィックスやパスを含めないでください。

ContentOriginType

コンテンツオリジンのタイプ:

- mediapackagev2: AWS Elemental MediaPackage オリジンの場合
- s3: Amazon S3 バケットオリジンの場合
- custom: 他のオリジンタイプの場合

テンプレートは、パーソナライズされた広告でコンテンツを配信するために連携する複数の AWS リソースを作成します。各コンポーネントの動作を以下に示します。

オリジンアクセスコントロール

オリジンアクセスコントロール (OAC) は、オリジンサーバーから直接ではなく CloudFront 経由でのみコンテンツにアクセスできるようにするセキュリティ機能です。これにより、コンテンツを不正アクセスから保護できます。

MediaPackage および Amazon S3 オリジンの場合、テンプレートはコンテンツへのアクセスを保護するためのオリジンアクセスコントロール (OAC) リソースを作成します。

MediaTailor の再生設定

MediaTailor 再生設定は、広告挿入を処理するコアコンポーネントです。オリジンからコンテンツを受信し、広告サーバーから広告をリクエストし、それらを各ビューワのパーソナライズされたストリームに結合します。

テンプレートは、以下の設定で MediaTailor 再生設定を作成します。

- CloudFront ディストリビューションを指す動画コンテンツソース
- 指定した VAST エンドポイントに設定された広告決定サーバー URL
- ライブストリーム中の広告挿入のライブプリロール設定
- 適切なセグメント URL プレフィックスを持つ CDN 設定

CloudFront 配信

CloudFront ディストリビューションは、低レイテンシーで世界中の視聴者にコンテンツを配信します。さまざまなタイプのリクエスト (マニフェスト、コンテンツセグメント、広告セグメント) を処理し、適切なオリジンにルーティングします。

CDNs、いくつかの重要な用語があります。

オリジン

元のコンテンツが保存されているサーバー (MediaPackage や Amazon S3 など)

キャッシュ動作

さまざまなタイプのコンテンツがどのようにキャッシュおよび配信されるかを決定するルール

キャッシュポリシー

コンテンツのキャッシュ期間とキャッシュに影響するリクエストコンポーネントを制御する設定

テンプレートは、以下のコンポーネントを使用して CloudFront ディストリビューションを作成します。

- 3 つのオリジン:
 - コンテンツオリジン (MediaPackage、Amazon S3、またはカスタム)
 - MediaTailor マニフェストオリジン

- MediaTailor セグメントのオリジン
- 適切なパターンで動作をキャッシュします。
 - コンテンツセグメントのデフォルトの動作
 - MediaTailor 広告セグメントの動作 (/tm/*)
 - MediaTailor インタースティシャルメディアの動作 (/v1/i-media/*)
 - パーソナライズされたマニフェストの動作 (/v1/*)
 - セグメントリダイレクトリクエストの動作 (/segment/*)
- 動作ごとに最適化されたキャッシュポリシー:
 - CachingOptimized キャッシュ可能なコンテンツの
 - CachingDisabled パーソナライズされたマニフェスト用
- 適切なヘッダー転送を確保するためのオリジンリクエストポリシー
- CORS サポートのレスポンスヘッダーポリシー

CDN と MediaTailor の統合用の CloudFormation テンプレートをデプロイする

AWS Elemental MediaTailor AWS CloudFormation テンプレートを使用したデプロイは、テンプレートが作成する内容を理解すると簡単です。このプロセスには約 15～30 分かかり、ほとんどの時間は CloudFront ディストリビューションのデプロイを待つために費やされます。

CloudFormation テンプレートをデプロイし、自動広告挿入ワークフローを設定するには:

MediaTailor CloudFormation テンプレートをデプロイするには

1. CloudFormation テンプレートを AWS Elemental MediaTailor GitHub リポジトリからダウンロードするか、 からコピーします [AWS CloudFormation AWS Elemental MediaTailor および Amazon CloudFront 統合の テンプレートリファレンス](#)。
2. [CloudFormation コンソール](#)を開きます。
3. スタックの作成 > 新しいリソースを使用する (標準) を選択します。
4. テンプレートを指定 で、テンプレートファイルをアップロード を選択し、テンプレートをアップロードします。
5. スタック名を入力し、必要なパラメータの値を指定します。
 - AdServerUrl: VAST 広告サーバーの URL (例: <https://your-ad-server.com/vast>)

- ContentOriginDomainName: プロトコルを使用しないコンテンツオリジンのドメイン名 (などで *mediapackage-domain.mediapackagev2.us-west-2.amazonaws.com*)
 - ContentOriginType: コンテンツオリジンのタイプを選択します。
 - mediapackagev2: AWS Elemental MediaPackage オリジンの場合
 - s3: Amazon S3 バケットオリジンの場合
 - custom: 他のオリジンタイプの場合
6. 設定を確認し、スタックの作成を選択します。
 7. スタックの作成が完了するまで待ちます。通常は 5 ~ 10 分かかります。イベントタブで進行状況をモニタリングできます。
 8. 完了したら、出力タブに移動して、HLS マニフェストと DASH マニフェストURLs を見つけます。

Note

をコンテンツオリジン AWS Elemental MediaPackage として使用している場合は、MediaPackage エンドポイントが正しく設定され、アクセス可能であることを確認します。詳細については、「[MediaPackage CDN 統合](#)」を参照してください。

CDN と MediaTailor の統合に CloudFormation デプロイされたリソースを使用する

AWS Elemental MediaTailor AWS CloudFormation スタックによってデプロイされた リソースには、広告挿入でコンテンツにアクセスするために使用する重要な出力がいくつか用意されています。CloudFormation スタックが正常に作成されたら、出力を使用して広告が挿入されたコンテンツにアクセスする方法を理解する必要があります。これは、手動セットアップで MediaTailor URLs を使用する方法と似ていますが、CloudFormation デプロイではこれらの URLs が自動的に提供されます。

デプロイが成功すると、CloudFormation スタックには、広告挿入を使用してコンテンツにアクセスするために使用する重要な出力がいくつか用意されます。

CloudFrontDomainName

CloudFront ディストリビューションのドメイン名 (など *d1234abcdef.CloudFront.net*)

HlsManifestUrl

広告挿入を含む HLS マニフェストの基本 URL (例: `https://d1234abcdef.CloudFront.net/v1/master/12345/my-playback-config/`)

DashManifestUrl

広告挿入を含む DASH マニフェストの基本 URL (例: `https://d1234abcdef.CloudFront.net/v1/dash/12345/my-playback-config/`)

MediaTailorPlaybackConfigName

作成された MediaTailor 再生設定の名前 (`my-stack-PlaybackConfig` など)

再生 URLs

広告を含むコンテンツの完全な再生 URL を作成するには、CloudFormation 出力のベース URL を特定のマニフェストパスと組み合わせる必要があります。これは、既存のコンテンツを広告挿入システムに接続するため、ブロードキャストプロフェッショナルが理解すべき重要なステップです。

1. 出力から適切なマニフェスト URL から開始します。

```
HlsManifestUrl: https://d1234abcdef.CloudFront.net/v1/master/12345/my-playback-config/
```

2. 特定のマニフェストパスを追加します。

```
Your manifest path: channel/index.m3u8
```

3. 完全な再生 URL は次のようになります。

```
https://d1234abcdef.CloudFront.net/v1/master/12345/my-playback-config/channel/index.m3u8
```

動的に挿入された広告でコンテンツを再生するには、ビデオプレーヤーでこの URL を使用します。

i Tip

マニフェストパスが不明な場合は、オリジンサーバーを確認してください。MediaPackage オリジンの場合、これはエンドポイントの HLS または DASH マニフェストへのパスです。Amazon S3 オリジンの場合、これはバケット内のマニフェストファイルへのパスです。

MediaTailor URL 構造の詳細については、「」を参照してください[MediaTailor との CDN 統合を設定する](#)。

ビデオプレーヤーを設定する

再生 URL を取得したら、それを使用するようにビデオプレーヤーを設定する必要があります。ブロードキャストプロフェッショナルの場合、これは任意の HLS または DASH ストリームにプレーヤーを設定するのに似ていますが、ストリームにはパーソナライズされた広告が含まれます。一般的なHLS.js プレイヤーを使用した簡単な例を次に示します。

```
<!DOCTYPE html>
<html>
<head>
  <title>MediaTailor Playback Example</title>
  <script src="https://cdn.jsdelivr.net/npm/hls.js@latest"></script>
</head>
<body>
  <video id="video" controls style="width: 640px; height: 360px;"></video>

  <script>
    const video = document.getElementById('video');
    const mediaUrl = 'https://<replaceable>d1234abcdef.CloudFront.net</replaceable>/v1/master/<replaceable>12345</replaceable>/<replaceable>my-playback-config</replaceable>/<replaceable>channel/index.m3u8</replaceable>';

    if (Hls.isSupported()) {
      const hls = new Hls();
      hls.loadSource(mediaUrl);
      hls.attachMedia(video);
    } else if (video.canPlayType('application/vnd.apple.mpegurl')) {
      video.src = mediaUrl;
    }
  </script>
</body>
```

```
</html>
```

次のようなプロのブロードキャストプレイヤーを使用することもできます。

- JW プレイヤー
- Bitmovin プレイヤー
- THEOplayer
- Video.js

MediaTailor とのプレイヤー統合の詳細については、「」を参照してください[MediaTailor 広告サーバー統合要件](#)。

CDN と MediaTailor の統合に関する CloudFormation デプロイのテストと検証

AWS Elemental MediaTailor デプロイの検証は、本番稼働前のブロードキャストプロフェッショナルにとって重要なステップです。このセクションでは、広告が正しく挿入され、コンテンツがスムーズに配信されるように、デプロイをテストする方法について説明します。

CloudFormation テンプレートをデプロイしたら、以下の手順に従ってセットアップが正しく機能していることを確認します。

MediaTailor と CloudFront の統合をテストするには

1. コンソールで CloudFormation すべてのリソースが正常に作成されたことを確認します。
2. MediaTailor コンソールで [MediaTailor](#) 再生設定がアクティブであることを確認します。
3. CloudFront [コンソールで CloudFront](#) ディストリビューションがデプロイされ、有効になっていることを確認します。
4. サンプルマニフェストを使用して再生をテストします。
 1. 「」の説明に従って、フル再生 URL を作成します[再生 URLs](#)。
 2. HLS または DASH をサポートするビデオプレイヤー (VLC、JW Player、AWS コンソールプレイヤーなど) を使用します。
 3. コンテンツの再生と広告が予想されるブレイクポイントに挿入されていることを確認します。
5. CloudWatch の MediaTailor ログに広告挿入エラーがないか確認します。

広告挿入をテストするときは、成功の指標を探します。

- コンテンツと広告間のスムーズな移行
- 広告は、予想されるブレイクポイント (プリロール、ミッドロール、ポストロール) に表示されます。
- 広告品質がコンテンツ品質と一致する
- 広告の移行中にバッファリングエラーや再生エラーが発生しない

テスト手順の詳細については、「」を参照してください[AWS Elemental MediaTailor 広告挿入動作について](#)。包括的な CDN 統合のテストと検証については、「」を参照してください[CDN と MediaTailor の統合のテストと検証](#)。

CDN と MediaTailor の統合に関する一般的な CloudFormation デプロイの問題をトラブルシューティングする

AWS Elemental MediaTailor デプロイの問題は、デプロイ中または再生中のオートメーションでも発生する可能性があります。ブロードキャストプロフェッショナルとして、これらの問題のトラブルシューティング方法を理解することは、広告挿入で信頼性の高いストリーミングサービスを維持するのに役立ちます。

CloudFormation デプロイまたは結果の MediaTailor と CloudFront の統合で問題が発生した場合は、以下の一般的な問題と解決策を参照してください。

CloudFormation デプロイの問題

スタックの作成が「リソースの作成に失敗しました」エラーで失敗する

考えられる原因:

- 無効なコンテンツオリジンドメイン名形式
- リソースを作成するアクセス許可が不十分

解決策: CloudFormation イベントタブで特定のリソースエラーを確認します。コンテンツオリジンドメイン名がプロトコルプレフィックスまたはパスなしで正しくフォーマットされていることを確認します。IAM ロールに、必要なすべてのリソースを作成するための十分なアクセス許可があることを確認します。

CloudFront デистриビューションのデプロイに時間がかかる

原因: CloudFront デистриビューションが完全にデプロイされるまでに通常 15～30 分かかります。

解決策: これは正常な動作です。テストする前に、デистриビューションが「デプロイ済み」状態になるまで待ちます。

再生と広告挿入の問題

コンテンツは再生されるが広告は挿入されない

考えられる原因:

- 広告決定サーバーが応答しない、または空の VAST を返さない
- コンテンツに広告マーカースが含まれていない

解決策: 広告サーバーにアクセス可能であり、有効な VAST レスポンスを返すことを確認します。コンテンツに適切な広告マーカース (ライブコンテンツの場合は SCTE-35 マーカース、VOD の場合は広告ブレイクタグ) があることを確認します。

403 コンテンツにアクセスするときに禁止されるエラー

考えられる原因:

- オリジンアクセスコントロールが正しく設定されていない
- オリジンバケットまたはエンドポイントのアクセス許可の問題

解決策: Amazon S3 オリジンの場合は、バケットポリシーが CloudFront デистриビューションからのアクセスを許可していることを確認します。MediaPackage オリジンの場合、オリジンアクセスコントロールが正しく設定され、エンドポイントにアクセスできることを確認します。

再生エラーまたはバッファリング

考えられる原因:

- キャッシュ動作パスパターンがコンテンツパスと一致しません
- オリジンドメインの設定が正しくない

解決策: キャッシュ動作に、リクエストを適切なオリジンにルーティングするための正しいパスパターンがあることを確認します。CloudFront ログをチェックして、リクエストを処理しているオリジンを確認し、それが想定どおりであることを確認します。

ブロードキャストプロフェッショナルにとって、以下の追加のトラブルシューティングのヒントが役立ちます。

- Amazon CloudWatch Logs Insights を使用して MediaTailor ログで特定のエラーパターンを検索する
- 本番稼働用広告サーバーを使用する前に、まずシンプルな VAST 広告サーバー (テンプレートで提供されているデフォルトサーバーなど) でテストします。
- コンソールの MediaTailor マニフェストインスペクターツールを使用してコンテンツの広告マーカを検証する
- ブラウザのデベロッパーツールのネットワークトラフィックをチェックして、広告サーバーへのリクエストが正しく行われているかどうかを確認します。

追加のトラブルシューティングについては、MediaTailor と CloudFront の両方の CloudWatch ログをチェックして、特定のエラーを特定します。

CDN と MediaTailor の統合の CloudFormation テンプレートをカスタマイズする

AWS Elemental MediaTailor テンプレートのカスタマイズにより、ブロードキャストプロフェッショナルは特定のワークフロー要件に合わせて AWS CloudFormation テンプレートを適応させることができます。基本的なテンプレートは多くのシナリオで機能しますが、これらのカスタマイズは、より複雑なニーズに対応するのに役立ちます。

以下の例は、テンプレートに追加できる YAML コードスニペットを示しています。YAML または CloudFormation 構文に精通していない場合は、開発者または AWS ソリューションアーキテクトと協力してこれらの変更を行うことを検討してください。

テンプレートは、特定のワークフロー要件を満たす CloudFormation ようにカスタマイズできます。

オリジンの追加または変更

複数のコンテンツソース (プライマリソースやバックアップソース、異なるコンテンツライブラリなど) を使用するブロードキャストワークフローでは、CloudFront デистриビューションにオリジンを追加できます。

```
Origins:  
  # Add a new origin for additional content
```

```
- Id: SecondaryContentOrigin
  DomainName: secondary-content.example.com
  CustomOriginConfig:
    OriginProtocolPolicy: 'https-only'
    OriginSSLProtocols:
      - TLSv1.2
```

次に、対応するキャッシュ動作を追加して、特定のパターンをこのオリジンにルーティングします。

```
CacheBehaviors:
  - PathPattern: '/secondary-content/*'
    TargetOriginId: SecondaryContentOrigin
    ViewerProtocolPolicy: 'https-only'
    CachePolicyId: 658327ea-f89d-4fab-a63d-7e88639e58f6 # Managed-CachingOptimized
```

カスタムキャッシュポリシーを作成する

特定のキャッシュ要件 (品質選択パラメータやビューワー認証など) を持つブロードキャストワークフローでは、マネージドキャッシュポリシーを使用する代わりにカスタムキャッシュポリシーを作成できます。TTL 値とキャッシュ戦略の詳細については、「」を参照してください [CDN と MediaTailor の統合のキャッシュ最適化](#)。

```
# Define a custom cache policy
CustomCachePolicy:
  Type: AWS::CloudFront::CachePolicy
  Properties:
    CachePolicyConfig:
      Name: !Sub '${AWS::StackName}-CustomCachePolicy'
      DefaultTTL: 86400 # 24 hours
      MaxTTL: 31536000 # 1 year
      MinTTL: 1 # 1 second
      ParametersInCacheKeyAndForwardedToOrigin:
        CookiesConfig:
          CookieBehavior: none
        HeadersConfig:
          HeaderBehavior: none
        QueryStringsConfig:
          QueryStringBehavior: whitelist
          QueryStrings:
            - quality
            - format
```



```
# Reference the custom policy in a cache behavior
CacheBehaviors:
  - PathPattern: '/custom-path/*'
    TargetOriginId: ContentOrigin
    ViewerProtocolPolicy: 'https-only'
    CachePolicyId: !Ref CustomCachePolicy
```

MediaTailor 設定の強化

高度な広告挿入機能を必要とするブロードキャストワークフローでは、広告プリフェッチ (レイテンシーを低減するため)、パーソナライゼーションしきい値、バンパー広告などのオプションを使用して MediaTailor 設定を強化できます。

```
MediaTailorPlaybackConfig:
  Type: AWS::MediaTailor::PlaybackConfiguration
  Properties:
    # Add ad prefetching for improved performance
    AvailSuppression:
      Mode: BEHIND_LIVE_EDGE
      Value: 00:00:00
    # Add personalization parameters
    PersonalizationThresholdSeconds: 2
    # Add bumper ads
    Bumper:
      StartUrl: https://example.com/bumper-start.mp4
      EndUrl: https://example.com/bumper-end.mp4
    # Other existing properties...
```

MediaTailor 設定オプションの詳細については、「」を参照してください [AWS Elemental MediaTailor を使用して広告を挿入する](#)。

セキュリティ機能を追加する

特定のセキュリティ要件 (地理的制限や DDoS 攻撃に対する保護など) を持つブロードキャストワークフローでは、AWS WAF 統合と地理的制限を追加できます。

```
# Create a AWS WAF Web ACL
WebACL:
  Type: AWS::WAFv2::WebACL
  Properties:
    Name: !Sub '${AWS::StackName}-WebACL'
```

```
Scope: CloudFront
DefaultAction:
  Allow: {}
VisibilityConfig:
  SampledRequestsEnabled: true
  CloudWatchMetricsEnabled: true
  MetricName: !Sub '${AWS::StackName}-WebACL'
Rules:
  - Name: RateLimitRule
    Priority: 0
    Action:
      Block: {}
    VisibilityConfig:
      SampledRequestsEnabled: true
      CloudWatchMetricsEnabled: true
      MetricName: RateLimitRule
    Statement:
      RateBasedStatement:
        Limit: 1000
        AggregateKeyType: IP

# Reference the AWS WAF Web ACL in the CloudFront distribution
CloudFrontDistribution:
  Type: AWS::CloudFront::Distribution
  Properties:
    DistributionConfig:
      WebACLId: !GetAtt WebACL.Arn
      # Add geo-restriction
      Restrictions:
        GeoRestriction:
          RestrictionType: whitelist
          Locations:
            - US
            - CA
            - GB
      # Other existing properties...
```

CloudFormation テンプレートの詳細については、[AWS CloudFormation 「ユーザーガイド」](#)を参照してください。

ブロードキャスト固有の CloudFormation テンプレートと例については、[AWS Media Services Tools GitHub リポジトリ](#)を参照してください。

AWS CloudFormation AWS Elemental MediaTailor および Amazon CloudFront 統合の テンプレートリファレンス

AWS Elemental MediaTailor Amazon CloudFront との統合は、次の完全な AWS CloudFormation テンプレートを使用して自動化できます。

```

AWSTemplateFormatVersion: '2010-09-09'
Description: | CloudFormation template that sets up AWS Elemental MediaTailor
  integration with CloudFront Distribution
  for server-side ad insertion. This template supports various content origins
  including MediaPackage, Amazon S3,
  and custom origins, making it versatile for different streaming architectures.

Parameters:
  AdServerUrl:
    Type: String
    Default: 'https://d1kbmkziz9rksx.CloudFront.net/VASTEndpoint.xml'
    Description: URL of the VAST ad server for dynamic ad insertion. Static VAST
    endpoint provided for testing.

  ContentOriginDomainName:
    Type: String
    Description: |
      Domain name of your content origin without protocol (e.g., mediapackage-
      domain.mediapackagev2.us-west-2.amazonaws.com,
      mybucket.s3.amazonaws.com, or custom-origin.example.com).
      Do not include http:// or https:// prefixes or any paths.
    AllowedPattern: "^(([a-zA-Z0-9]|[a-zA-Z0-9][a-zA-Z0-9\\-]*[a-zA-Z0-9])\\.)*([A-Za-
    z0-9]|[A-Za-z0-9][A-Za-z0-9\\-]*[A-Za-z0-9])$"
    ConstraintDescription: Must be a valid domain name (e.g., example.com) without
    protocol or path components. IP addresses are not allowed.

  ContentOriginType:
    Type: String
    AllowedValues:
      - mediapackagev2
      - s3
      - custom
    Default: mediapackagev2
    Description: |
      The type of content origin:
      - mediapackagev2: AWS Elemental MediaPackage V2
      - s3: Amazon S3 bucket

```

- custom: Any other custom origin

Resources:

```
#-----
# Origin Access Control (for securing MediaPackage V2 and Amazon S3 origins)
#-----
CloudFrontOriginAccessControl:
  Type: AWS::CloudFront::OriginAccessControl
  Condition: IsNotCustomOrigin
  Properties:
    OriginAccessControlConfig:
      Name: !Sub '${AWS::StackName}-OAC'
      OriginAccessControlOriginType: !Ref ContentOriginType
      SigningBehavior: always
      SigningProtocol: sigv4
      Description: Origin Access Control for content origin

#-----
# MediaTailor Playback Configuration
#-----
MediaTailorPlaybackConfig:
  Type: AWS::MediaTailor::PlaybackConfiguration
  Properties:
    Name: !Sub '${AWS::StackName}-PlaybackConfig'
    # The video content source should point to your CloudFront distribution
    VideoContentSourceUrl: !Sub 'https://${CloudFrontDistribution.DomainName}/'
    # The Ad Decision Server URL is where MediaTailor will request ads
    AdDecisionServerUrl: !Ref AdServerUrl
    # Configuration for pre-roll ads during live streams
    LivePreRollConfiguration:
      AdDecisionServerUrl: !Ref AdServerUrl
      MaxDurationSeconds: 30
    # CDN configuration for integrating with CloudFront
    CdnConfiguration:
      AdSegmentUrlPrefix: '/'
      ContentSegmentUrlPrefix: '/'
    # Set a reasonable manifest segment timeout
    ManifestProcessingRules:
      AdMarkerPassthrough:
        Enabled: false

#-----
# CloudFront Distribution
#-----
```

```

CloudFrontDistribution:
  Type: AWS::CloudFront::Distribution
  Properties:
    DistributionConfig:
      Enabled: true
      HttpVersion: http2and3
      IPV6Enabled: true
      Comment: !Sub 'Distribution for MediaTailor ad insertion with
${ContentOriginType} origin'

      # Default cache behavior points to the content origin
      DefaultCacheBehavior:
        TargetOriginId: ContentOrigin
        ViewerProtocolPolicy: 'https-only'
        # Using managed policies for optimal performance and simplicity
        CachePolicyId: 658327ea-f89d-4fab-a63d-7e88639e58f6 # Managed-
CachingOptimized
        OriginRequestPolicyId: 88a5eaf4-2fd4-4709-b370-b4c650ea3fcf # Managed-
HostHeaderOnly
        ResponseHeadersPolicyId: eaab4381-ed33-4a86-88ca-d9558dc6cd63 # Managed-
CORS-with-preflight-and-SecurityHeadersPolicy
        Compress: true

      # Define all the origins needed for the workflow
      Origins:
        # Main content origin (MediaPackage, Amazon S3, or Custom)
        - Id: ContentOrigin
          DomainName: !Ref ContentOriginDomainName
          # Apply Origin Access Control for secure origins
          OriginAccessControlId: !If [IsNotCustomOrigin, !GetAtt
CloudFrontOriginAccessControl.Id, !Ref "AWS::NoValue"]
          # For custom origins, we need a CustomOriginConfig
          CustomOriginConfig:
            OriginProtocolPolicy: 'https-only'
            OriginSSLProtocols:
              - TLSv1.2
            OriginKeepaliveTimeout: 5
            OriginReadTimeout: 30
            HTTPPort: 80
            HTTPSPort: 443

        # MediaTailor Manifests Origin - handles manifest manipulation for ad
insertion
        - Id: MediaTailorManifests

```

```

    DomainName: !Sub 'manifests.mediatailor.${AWS::Region}.amazonaws.com'
    CustomOriginConfig:
      OriginProtocolPolicy: 'https-only'
      OriginSSLProtocols:
        - TLSv1.2
      OriginKeepaliveTimeout: 5
      OriginReadTimeout: 30
      # Origin Shield improves caching efficiency
    OriginShield:
      Enabled: true
      OriginShieldRegion: !Ref AWS::Region

# MediaTailor Segments Origin - handles personalized ads
- Id: MediaTailorSegments
  DomainName: !Sub 'segments.mediatailor.${AWS::Region}.amazonaws.com'
  CustomOriginConfig:
    OriginProtocolPolicy: 'https-only'
    OriginSSLProtocols:
      - TLSv1.2
    OriginKeepaliveTimeout: 5
    OriginReadTimeout: 30

# Cache behaviors to route specific request patterns to the right origin
CacheBehaviors:
  # Handle MediaTailor segment requests for ad content which are cache-able
  - PathPattern: '/tm/*'
    TargetOriginId: MediaTailorSegments
    ViewerProtocolPolicy: 'https-only'
    CachePolicyId: 658327ea-f89d-4fab-a63d-7e88639e58f6 # Managed-
    CachingOptimized
    OriginRequestPolicyId: 88a5eaf4-2fd4-4709-b370-b4c650ea3fcf # Managed-
    HostHeaderOnly
    ResponseHeadersPolicyId: eaab4381-ed33-4a86-88ca-d9558dc6cd63 # Managed-
    CORS-with-preflight-and-SecurityHeadersPolicy
    Compress: true

  # Handle MediaTailor interstitial (SGAI) media requests which are cache-able
  - PathPattern: '/v1/i-media/*'
    TargetOriginId: MediaTailorManifests
    ViewerProtocolPolicy: 'https-only'
    CachePolicyId: 658327ea-f89d-4fab-a63d-7e88639e58f6 # Managed-
    CachingOptimized
    OriginRequestPolicyId: 88a5eaf4-2fd4-4709-b370-b4c650ea3fcf # Managed-
    HostHeaderOnly

```

```

        ResponseHeadersPolicyId: eaab4381-ed33-4a86-88ca-d9558dc6cd63 # Managed-
CORS-with-preflight-and-SecurityHeadersPolicy
        Compress: true

        # Handle MediaTailor Personalized manifests which are not cache-able
        - PathPattern: '/v1/*'
          TargetOriginId: MediaTailorManifests
          ViewerProtocolPolicy: 'https-only'
          CachePolicyId: 4135ea2d-6df8-44a3-9df3-4b5a84be39ad # Managed-
CachingDisabled
          OriginRequestPolicyId: 59781a5b-3903-41f3-afcb-af62929ccde1 # Managed-
AllViewer
          ResponseHeadersPolicyId: eaab4381-ed33-4a86-88ca-d9558dc6cd63 # Managed-
CORS-with-preflight-and-SecurityHeadersPolicy
          Compress: true

          # Handle MediaTailor segment *redirect* requests which are not cache-able
          (used for server side reporting)
          - PathPattern: '/segment/*'
            TargetOriginId: MediaTailorManifests
            ViewerProtocolPolicy: 'https-only'
            CachePolicyId: 4135ea2d-6df8-44a3-9df3-4b5a84be39ad # Managed-
CachingDisabled
            OriginRequestPolicyId: 59781a5b-3903-41f3-afcb-af62929ccde1 # Managed-
AllViewer
            ResponseHeadersPolicyId: eaab4381-ed33-4a86-88ca-d9558dc6cd63 # Managed-
CORS-with-preflight-and-SecurityHeadersPolicy
            Compress: true

Conditions:
  IsNotCustomOrigin: !Not [!Equals [!Ref ContentOriginType, 'custom']]

Outputs:
  CloudFrontDomainName:
    Description: Domain name of the CloudFront distribution
    Value: !GetAtt CloudFrontDistribution.DomainName

  HlsManifestUrl:
    Description: URL for HLS manifest with ads inserted (append your manifest path)
    Value: !Sub 'https://
${CloudFrontDistribution.DomainName}${MediaTailorPlaybackConfig.HlsConfiguration.ManifestEndpoint

  DashManifestUrl:
    Description: URL for DASH manifest with ads inserted (append your manifest path)

```

```
Value: !Sub 'https://  
${CloudFrontDistribution.DomainName}${MediaTailorPlaybackConfig.DashConfiguration.ManifestEndpo  
  
MediaTailorPlaybackConfigName:  
Description: Name of the MediaTailor playback configuration  
Value: !Ref MediaTailorPlaybackConfig
```

MediaTailor の本番稼働対応 CloudFront 設定

この CloudFront デイストリビューション設定は、MediaTailor コンテンツをサーバー側の広告挿入で大規模に配信するために必要なすべてを提供します。この設定をコピーし、特定のオリジンと要件に合わせてカスタマイズします。

この設定の目的

この設定により、最適なキャッシュとパフォーマンスですべての MediaTailor リクエストタイプを処理する、本稼働対応の CloudFront デイストリビューションが作成されます。これには、リクエストを正しくルーティングし、コンテンツを適切にキャッシュする 4 つのキャッシュ動作を持つ 3 つのオリジン (コンテンツ、MediaTailor セグメント、MediaTailor マニフェスト) が含まれます。

この設定を使用するタイミング

この設定は、サーバー側の広告挿入を必要とするライブストリーミング、video-on-demand、ハイブリッドワークフローに最適です。

3 オリジンアーキテクチャ

MediaTailor は、3 つのオリジンアーキテクチャパターンを使用して、コンテンツ配信と広告挿入のパフォーマンスを最適化します。各オリジンは、広告挿入ワークフローで特定の目的を果たします。

コンテンツオリジン

これは、MediaTailor にフィードする実際のコンテンツオリジンです。例えば、V AWS Elemental MediaPackage V2 や別のコンテンツ配信サービスなどです。このオリジンは、広告挿入前に元のコンテンツを提供します。一般的な例としては次のようなものがあります。

- MediaPackage V2 パッケージ設定
- サードパーティーのコンテンツ配信ネットワーク
- オンプレミスストリーミングサーバー
- 静的コンテンツを含む Amazon S3 バケット

MediaTailor セグメントのオリジン

このオリジンはホスト名を使用

し `segments.mediatailor.region.amazonaws.com`、MediaTailor がエンコードした後に実際の広告セグメントを提供します。これらは広告を含む動画セグメントです。このオリジンは以下を処理します。

- コンテンツと同じ形式のトランスコードされた広告セグメント
- `/segment/*` パスパターンからリダイレクトされたリクエスト
- サーバー側の広告挿入用に処理された広告セグメント

MediaTailor マニフェストオリジン

このオリジンはホスト名を使用し `manifests.mediatailor.region.amazonaws.com`、指定されたリージョンの再生設定の AWS リージョンホスト名として使用できます。MediaTailor は、リクエストのパスに基づいて正しい再生設定を選択します。このオリジンは以下を提供します。

- ビューワー固有の広告挿入を使用したパーソナライズされた HLS および DASH マニフェスト
- キャッシュ可能なコンテンツのサーバーガイド広告挿入 (SGAI) マニフェスト
- サーバー側のレポートの広告追跡とビーコン処理

オリジンホスト名 を使用すると `manifests.mediatailor.region.amazonaws.com`、指定されたリージョンにあり、リクエストパスに再生設定名を含めると、同じ CloudFront デイストリビューションで動作する複数の再生設定を持つことができます。例えば、次のようになります。

- `https://your-distribution.cloudfront.net/v1/master/playback-config-1/manifest.m3u8`
- `https://your-distribution.cloudfront.net/v1/master/playback-config-2/manifest.m3u8`

コンテンツタイプに基づいてオリジンリクエストポリシーを選択し、適切な機能を確保しながらキャッシュポイズニングを防止します。主な違いは、キャッシュ可能なコンテンツとキャッシュ不可能なコンテンツです。

- マニフェスト (キャッシュ不可): 動的コンテンツに必要なすべてのヘッダーを転送 AllViewer するために使用します。マニフェストはキャッシュされないため、キャッシュポイズニングのリスクはありません。
- セグメント (キャッシュ可能): 最適なパフォーマンス None のために を使用します。

- S3 オリジン: Amazon S3 バケットCORS-S3originに使用
- MediaPackage オリジン: MediaPackage V2 エンドポイントCORS-S3originに使用

Origins							Edit	Delete	Create origin
Filter origins by property or value							< 1 > ⚙		
Origin name	Origin domain	Origin path	Origin type	Origin Shield region	Origin access				
☐ ContentOrigin	zrljic.egress.kv9ydh.mediapackagev2.us-west-2.amazonaws.com		MediaPackage V2	us-west-2	E2SRJY3NEMC428				
☐ MediaTailorSegments	segments.mediatailor.us-west-2.amazonaws.com		MediaTailor	-	-				
☐ MediaTailorManifests	manifests.mediatailor.us-west-2.amazonaws.com		MediaTailor	us-west-2	-				

キャッシュ動作の優先順位と設定

MediaTailor では、さまざまなタイプのリクエストを適切に処理するために、特定のキャッシュ動作設定が必要です。CDNs処理し、最初の一致するパスパターンに動作を使用するため、キャッシュ動作の優先順位は重要です。この優先順位を理解することは、トラブルシューティングに不可欠です。

- 優先順位 0: ほとんどの特定のパターン (など/tm/*) が最初に評価されます
- 優先順位が高い: より具体的なパターンが順番に評価される
- デフォルトの動作: 他のパターンと一致しないすべてのリクエストをキャッチします

リクエストが想定どおりに動作しない場合は、パスパターンが意図しない方法で重複していないことを確認します。

Behaviors							Save	Move up	Move down	Edit	Delete	Create behavior
Filter behaviors by property or value												
Preced...	Path patt...	Origin or origin group	Viewer pr...	Cache policy name	Origin request policy name	Response headers policy name						
☐ 0	/tm/*	MediaTailorSegments	HTTPS only	Managed-CachingOptimized	Managed-CORS-S3Origin	Managed-CORS-with-preflight-and-SecurityHeadersPolicy						
☐ 1	/i-media/*	MediaTailorManifests	HTTPS only	Managed-CachingOptimized	Managed-CORS-S3Origin	Managed-CORS-with-preflight-and-SecurityHeadersPolicy						
☐ 2	/v1/*	MediaTailorManifests	HTTPS only	Managed-CachingDisabled	Managed-AllViewerAndCloudFrontHeaders-2022-06	Managed-CORS-with-preflight-and-SecurityHeadersPolicy						
☐ 3	/segment/*	MediaTailorManifests	HTTPS only	Managed-CachingDisabled	Managed-AllViewerAndCloudFrontHeaders-2022-06	Managed-CORS-with-preflight-and-SecurityHeadersPolicy						
☐ 4	Default (*)	ContentOrigin	HTTPS only	Managed-CachingOptimized	CustomOriginRequestPolicyWithWithLLHLSDirectives	Managed-CORS-with-preflight-and-SecurityHeadersPolicy						

優先順位 0: 広告セグメントのパス動作

この動作は、セグメントパス動作からのリダイレクトされたリクエストを処理し、実際の広告セグメントを処理します。CloudFront は、/tm/*パスパターンを持つすべてのリクエストに次の動作を適用します。これは、広告セグメント配信が中断のない再生に不可欠であるため、最も優先度の高い動作です。

- パスパターン: /tm/*

このパターンに一致する URLs の例:

- `https://your-distribution.cloudfront.net/tm/ad-segment-001.ts`
- `https://your-distribution.cloudfront.net/tm/transcoded-ad.m4s`
- オリジン: `segments.mediatailor.region.amazonaws.com` ドメインで作成したオリジン。

これは、オリジンに関する前のセクションの例の MediaTailorSegments です。

- キャッシュポリシー: Managed-CachingOptimized

CloudFront マネージドキャッシュポリシーの内容の詳細については、CloudFront ユーザーガイドの [CachingOptimized](#)」を参照してください。サードパーティー CDN の 管理ポリシーから、これらの同じ設定を使用することもできます。

- オリジンリクエストポリシー: None
- レスponseヘッダーポリシー: Managed-CORS-with-preflight-and-SecurityHeadersPolicy

レスponseヘッダーポリシーに含まれる内容の詳細については、CloudFront ユーザーガイドの [CORS-with-preflight-and-SecurityHeadersPolicy](#)」を参照してください。

これらの設定を他の CDN s

CloudFront 以外の CDN を使用している場合は、以下を達成する同等の設定を探します。

パスパターンマッチング

MediaTailor 広告セグメントを処理するパス/tm/*パターンに特定の動作を設定する
キャッシュキーの設定

キャッシュキーに Originヘッダーを含めて、レスponseが異なるオリジンに対して個別に
キャッシュされるようにします。

ヘッダー転送

Origin ヘッダーおよびその他の CORS 関連ヘッダーをオリジンに転送する

レスponseヘッダー管理

Access-Control-Allow-Origin ヘッダーがレスponseに存在することを確認するように
CDN を設定する

特定の用語と設定オプションは CDN プロバイダーによって異なりますが、基盤となる原則は変わりません。

優先順位 1: サーバーガイド広告挿入動作

この動作は、お客様がキャッシュ可能なマニフェストを提供するガイドモードを設定するときに [MediaTailor サーバーガイド広告挿入の概要と実装](#) (SGAI) リクエストを処理します。CloudFront は、`/v1/i-media/*` パスパターンを持つすべてのリクエストに次の動作を適用します。SGAI では、マニフェストがビューワー固有ではないため、キャッシュパフォーマンスが向上します。

- パスパターン: `/v1/i-media/*` (SGAI の iMedia パス)

このパターンに一致する URLs の例:

- `https://your-distribution.cloudfront.net/v1/i-media/your-config/manifest.m3u8`
- `https://your-distribution.cloudfront.net/v1/i-media/your-config/playlist.mpd`
- オリジン: `manifests.mediataylor.region.amazonaws.com` ドメインで作成したオリジン。

これは、オリジンに関する前のセクションの例の MediaTailorManifests です。

- キャッシュポリシー: `Managed-CachingOptimized`

CloudFront マネージドキャッシュポリシーの内容の詳細については、CloudFront ユーザーガイドの [CachingOptimized](#)」を参照してください。サードパーティー CDN の 管理ポリシーから、これらの同じ設定を使用することもできます。

- オリジンリクエストポリシー: `None`
- レスポンスヘッダーポリシー: `Managed-CORS-with-preflight-and-SecurityHeadersPolicy`

レスポンスヘッダーポリシーに含まれる内容の詳細については、CloudFront ユーザーガイドの [CORS-with-preflight-and-SecurityHeadersPolicy](#)」を参照してください。

優先順位 2: パーソナライズされたマニフェストの動作

この動作は、パーソナライズされたマニフェストリクエストを処理します。CloudFront は、`/v1/*` パスパターンを持つすべてのリクエストに次の動作を適用します。CloudFront は、ビューワー固有の広告コンテンツ URLs が含まれているため、以下の動作を適用し、パーソナライズされたマニ

フェストをキャッシュしません。これらの動作は、/v1/*パスパターンを持つすべてのリクエストに適用されます。これは、各ビューワーがパーソナライズされた広告挿入を含む一意のマニフェストを受け取る主要な MediaTailor 機能です。

- パスパターン: /v1/* (標準の V1 MediaTailor リクエスト)

このパターンに一致する URLs の例:

- `https://your-distribution.cloudfront.net/v1/master/your-config/manifest.m3u8`
- `https://your-distribution.cloudfront.net/v1/dash/your-config/manifest.mpd`
- オリジン: `manifests.mediatailor.region.amazonaws.com` ドメインで作成したオリジン。

これは、オリジンに関する前のセクションの例の MediaTailorManifests です。

- キャッシュポリシー: `Managed-CachingDisabled`

キャッシュポリシーの内容の詳細については、CloudFront ユーザーガイドの [CachingDisabled](#)」を参照してください。

- オリジンリクエストポリシー: `AllViewer`

パーソナライズされたマニフェストの場合は、`AllViewer`ポリシーを使用して、動的コンテンツに必要なすべてのヘッダーを転送します。

- レスポンスヘッダーポリシー: `Managed-CORS-with-preflight-and-SecurityHeadersPolicy`

レスポンスヘッダーポリシーに含まれる内容の詳細については、CloudFront ユーザーガイドの [CORS-with-preflight-and-SecurityHeadersPolicy](#)」を参照してください。

Accept-Encoding ヘッダー

CDN はビューワーの `Accept-Encoding` ヘッダーを保持することをお勧めします。このヘッダーは、パーソナライズされたマニフェストの圧縮に関する指示を MediaTailor に与えます。

CloudFront では、`AllViewerAndCloudFrontHeaders` オリジンリクエストポリシーにビューワーからの `Accept-Encoding` ヘッダーのパススルーが含まれます。別の CDN を使用する場合は、このヘッダーが保持されていることを確認します。

以下は、MediaTailor が `Accept-Encoding` ヘッダーを処理する方法です。

- レガシーデバイス: gzip をサポートしていない古いスマートTVsは Accept-Encoding ヘッダーを送信しないため、MediaTailor は非圧縮マニフェストを返します
- 最新のデバイス: iPhones、Chrome ブラウザ、およびその他の最新のクライアントが Accept-Encoding ヘッダーを送信し、MediaTailor が配信前にマニフェストを圧縮できるようにします。

優先順位 3: サーバー側のビーコンパスの動作

この動作は、のリダイレクトにつながる MediaTailor へのリクエストを処理します[サーバー側の追跡](#)。これらのリクエストはビーコンを追跡するために不可欠であるため、すべてのリクエストは MediaTailor によって処理される必要があります。CloudFront は、/segment/*パスパターンを持つすべてのリクエストに次の動作を適用します。

- パスパターン: /segment/*

このパターンに一致する URLs の例:

- `https://your-distribution.cloudfront.net/segment/tracking-beacon-123`
- `https://your-distribution.cloudfront.net/segment/ad-request-456.ts`
- オリジン: manifests.mediatailor.*region*.amazonaws.com ドメインで作成したオリジン。

これは、オリジンに関する前のセクションの例の MediaTailorManifests です。

- キャッシュポリシー: Managed-CachingDisabled

キャッシュポリシーの内容の詳細については、CloudFront ユーザーガイドの[CachingDisabled](#)」を参照してください。

- オリジンリクエストポリシー: AllViewer

サーバー側のビーコンリクエストの場合、AllViewerポリシーを使用して、追跡に必要なすべてのヘッダーを転送します。これらのリクエストはキャッシュされないため、キャッシュポイズニングのリスクはありません。

- レスponseヘッダーポリシー: Managed-CORS-with-preflight-and-SecurityHeadersPolicy

レスponseヘッダーポリシーに含まれる内容の詳細については、CloudFront ユーザーガイドの[CORS-with-preflight-and-SecurityHeadersPolicy](#)」を参照してください。

MediaTailor がこれらのリクエストを処理すると、実際のセグメントの場所を指すパスを持つ 302 リダイレクトレスポンスが返されます。たとえば、へのリクエストは、セグメントオリジン/`tm/encoded-ad-segment.ts`の にリダイレクト/`segment/ad123.ts`される場合があります。

優先順位 4: コンテンツオリジンパスの動作

リクエストパスが他のパターンのいずれとも一致しない場合、CloudFront はデフォルトの動作を適用します。この動作は、MediaTailor からの処理なしで、リクエストをコンテンツオリジンに直接送信します。これにより、必要に応じてコンテンツオリジン (MediaPackage V2 など) に直接アクセスできます。CloudFront は、以前のパスパターンを含まないすべてのリクエストに次の動作を適用します。

- パスパターン: (*)
- オリジン: コンテンツオリジンのドメインで作成したオリジン。

これは、オリジンに関する前のセクションの例の ContentOrigin です。

- キャッシュポリシー: Managed-CachingOptimized

CloudFront マネージドキャッシュポリシーの内容の詳細については、CloudFront ユーザーガイドの [CachingOptimized](#)」を参照してください。サードパーティー CDN の 管理ポリシーから、これらの同じ設定を使用することもできます。

Note

低レイテンシーの HLS 実装では、標準ポリシーの代わりに低レイテンシー HLS (LLH) ディレクティブでカスタムキャッシュCachingOptimizedポリシーを使用することを確認してください。

- オリジンリクエストポリシー: None
- レスponsヘッダーポリシー: Managed-CORS-with-preflight-and-SecurityHeadersPolicy

デフォルトのコンテンツオリジン動作は通常、広告セグメント動作と同じ CORS キャッシュポイズニングリスクに直面しませんが、Managed-CORS-with-preflight-and-SecurityHeadersPolicyレスポンスヘッダーポリシーを使用し、キャッシュキーに Originヘッダーを含めることをお勧めします。これにより、すべてのコンテンツタイプで一貫した CORS 処理が保証され、ウェブベースのプレイヤーで発生する可能性のある再生の問題が防止されます。

コンテンツセグメントの場合、Managed-CachingOptimizedキャッシュポリシーは優れたパフォーマンスを提供し、Managed-CORS-with-preflight-and-SecurityHeadersPolicyレスポンスヘッダーポリシーは適切な CORS 処理を保証します。この組み合わせにより、CORS ヘッダーを必要とするウェブベースのプレイヤーとの互換性を維持しながら、効率的なキャッシュが可能になります。

広告セグメントとコンテンツセグメントの両方に一貫した CORS 処理を適用すると、より信頼性の高い再生エクスペリエンスが得られ、トラブルシューティングが簡素化されます。適切な CORS 設定がないと、コンテンツと広告間の移行時にプレイヤーに一貫性のない動作が発生する可能性があります。

レスポンスヘッダーポリシーに含まれる内容の詳細については、CloudFront ユーザーガイドの[CORS-with-preflight-and-SecurityHeadersPolicy](#)」を参照してください。

CDN と MediaTailor の統合のサポートとトラブルシューティングのヘルプを取得する

AWS Elemental MediaTailor CDN 統合の問題は、診断と解決が複雑な場合があります。セルフサービスのトラブルシューティングでは解決できない CDN と MediaTailor の統合の問題に関する追加のヘルプが必要な場合は、このガイドを使用してください。

エスカレーションする前に、以下のセルフサービスオプションを試してください。

- のトラブルシューティングステップに従います。 [CDN 統合のトラブルシューティング](#)
- を使用してログとエラーコードを分析する [CDN 統合ログ分析リファレンス](#)
- でモニタリング設定を確認する [モニタリングツールのセットアップ](#)
- サービス固有の問題については、[MediaTailor トラブルシューティングガイド](#)を参照してください。
- [AWS re:Post](#) で同様の問題やコミュニティソリューションを検索する
- [AWS ナレッジセンター](#)で一般的な統合パターンを確認する

トピック

- [MediaTailor CDN の問題を AWS サポートにエスカレーションするタイミング](#)
- [サポートに連絡する前に MediaTailor CDN 情報を収集する](#)

- [MediaTailor CDN 統合のサポートリソースとチャネル](#)
- [MediaTailor CDN 統合サポートケースのベストプラクティス](#)

MediaTailor CDN の問題を AWS サポートにエスカレーションするタイミング

AWS Elemental MediaTailor セルフサービスのトラブルシューティングで問題が解決しない場合は、CDN 統合の問題を AWS サポートにエスカレーションする必要があります。以下の場合は、サポートへの AWS エスカレーションを検討してください。

注: テクニカルサポートには有料 AWS サポートプランが必要です。サポートプランの詳細と対応時間については、[「AWS サポートプラン」](#)を参照してください。

- 問題が本番トラフィックまたは収益に影響する
- 解決せずにすべてのトラブルシューティング手順に従った
- エラーパターンはサービスレベルの問題を示唆する
- 複雑な設定シナリオについてはサポートが必要です
- 最適化作業を行ってもパフォーマンスの問題は続く

サポートに連絡する前に MediaTailor CDN 情報を収集する

AWS Elemental MediaTailor CDN 統合のサポートケースでは、効果的なトラブルシューティングのために特定の設定と診断情報が必要です。AWS サポートに連絡する前に、この情報を収集して解決を迅速化します。

ヒント: サポートケースを作成する前にこの情報を準備すると、解決時間が大幅に短縮され、サポートエンジニアが特定の設定を理解するのに役立ちます。

- MediaTailor 設定の詳細:
 - 設定名と AWS リージョン
 - 再生設定 ARN
 - ADS URL と統合タイプ
- CDN 設定の詳細:
 - CDN ディストリビューション ID または設定名
 - マニフェストとセグメントのキャッシュ動作設定

- オリジン設定とルーティングルール
- エラー情報:
 - 特定のエラーメッセージと HTTP ステータスコード
 - 問題が発生したときのタイムスタンプ
 - 問題を示すサンプル URLs
 - 問題に関連する CDN および MediaTailor ログエントリ
- テスト情報:
 - トラブルシューティングのために既に実行したステップ
 - 問題が発生したデバイスとプレイヤー
 - 問題がすべてのコンテンツまたは特定のストリームに影響するかどうか
 - 問題の頻度とパターン (断続的、一貫性、時間ベース)

MediaTailor CDN 統合のサポートリソースとチャネル

AWS Elemental MediaTailor CDN 統合のサポートは、問題の解決と実装の最適化に役立つ複数のチャネルを通じて利用できます。

- AWS サポート: AWS マネジメントコンソールを使用してサポートケースを作成し、テクニカルサポートを受ける

アクセス: [AWS サポートセンター](#)

- AWS re:Post: AWS 関連する質問とコミュニティサポートのためのコミュニティ主導の Q&A プラットフォーム

アクセス: [AWS re:Post](#)

- AWS ドキュメント: MediaTailor および CDN サービスの包括的なガイド

アクセス: [MediaTailor ドキュメント](#)と [CloudFront ドキュメント](#)

- AWS トレーニング: メディアサービスと CDN 最適化のコースと認定

アクセス: [AWS トレーニングと認定](#)

- AWS ナレッジセンター: 一般的な AWS 問題とベストプラクティスに関する厳選された記事

アクセス: [AWS ナレッジセンター](#)

- AWS Trusted Advisor: 最適化とベストプラクティスに関する自動レコメンデーション

アクセス: [AWS Trusted Advisor](#)

- AWS Personal Health Dashboard: サービスのヘルスとメンテナンスの通知

アクセス: [AWS Personal Health Dashboard](#)

MediaTailor CDN 統合サポートケースのベストプラクティス

AWS Elemental MediaTailor CDN 統合のサポートケースは、これらのベストプラクティスに従うとより効率的に解決されます。最速の解像度を得るには:

- ビジネスへの影響に基づいて適切な重要度レベルを選択する
- 最初のケース送信ですべての関連情報を入力する
- 特定の例と再現手順を含める
- 関連するログファイルと設定スクリーンショットをアタッチする
- エンジニアからの追加情報のリクエストにすみやかに対応する

その他のサポートリソース:

- [AWS サポートケース管理](#) - サポートケースの作成と管理に関するガイド
- [AWS サポートプラン](#) - サポートプランの特徴と対応時間を比較する
- [ワークロードをサポートする準備をする](#) - 運用準備のための Well-Architected ガイダンス
- [AWS Well-Architected フレームワーク](#) - でワークロードを構築および運用するためのベストプラクティス AWS

のセキュリティ AWS Elemental MediaTailor

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 お客様とお客様の間の責任共有です。[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。は、お客様が安全に使用できるサービス AWS も提供します。サードパーティーの監査者は、[AWS コンプライアンスプログラム](#)コンプライアンスプログラムの一環として、当社のセキュリティの有効性を定期的にテストおよび検証。が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 AWS Elemental MediaTailor、「[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、ユーザーは、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、MediaTailor を使用する際の責任共有モデルの適用方法を理解するのに役立ちます。以下のトピックでは、セキュリティおよびコンプライアンス上の目的を達成するために MediaTailor を設定する方法を説明します。また、MediaTailor リソースのモニタリングや保護に役立つ他の AWS サービスの使用方法についても説明します。

トピック

- [でのデータ保護 AWS Elemental MediaTailor](#)
- [の Identity and Access Management AWS Elemental MediaTailor](#)
- [のコンプライアンス検証 AWS Elemental MediaTailor](#)
- [の耐障害性 AWS Elemental MediaTailor](#)
- [MediaTailor のインフラストラクチャセキュリティ](#)
- [サービス間での混乱した使節の防止](#)
- [MediaTailor でのログ記録とモニタリング](#)

でのデータ保護 AWS Elemental MediaTailor

AWS [責任共有モデル](#) は、AWS Elemental MediaTailorでのデータ保護に適用されます。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。ユーザーは、このインフラストラクチャでホストされ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、「[Data Privacy FAQChina](#)」を参照してください。欧州におけるデータ保護に関する情報については、[General Data Protection Regulation \(GDPR\) Center](#) 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 アイデンティティセンターまたは 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 は必須ですが、TLS 1.3 を推奨します。
- 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。CloudTrail 証跡を使用して AWS アクティビティをキャプチャする方法については、「AWS CloudTrail ユーザーガイド」の[CloudTrail 証跡の使用](#)」を参照してください。
- 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 などの高度な管理された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 140-3 検証済み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 \(FIPS\) 140-3](#)」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を、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含めないことを強くお勧めします。これは、コンソール、API、または SDK を使用して MediaTailor AWS CLI または他の AWS のサービスを使用する場合も同様です。AWS SDKs タグ、または名前に使用される自由記述のテキストフィールドに入力したデータは、請求または診断ログに使用される場合があります。外部サーバーに URL を提供する場合、そのサーバーへのリクエストを検証できるように、認証情報を URL に含めないことを強くお勧めします。

データ暗号化

AWS Elemental MediaTailor は、コンテンツマニフェストの管理、またはサーバー、CDNs、プレイヤーとの通信において、データを暗号化または復号しません。MediaTailor は、顧客データ、またはその他機密情報の入力をユーザーに求めません。

顧客アカウント番号、クレジットカード情報、サインイン認証情報などの機密情報を自由形式のフィールドやクエリパラメータに入れないでください。これは、コンソール AWS Elemental MediaTailor、API、SDKs、AWS Command Line Interface () など、のすべての使用に適用されます AWS CLI。サービスに入力したデータはすべて、診断ログの内容として取得される可能性があります。

外部サーバーへの URL を指定するときは、そのサーバーへのリクエストを検証するための暗号化されていない認証情報を URL に含めないでください。

の Identity and Access Management AWS Elemental MediaTailor

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 MediaTailor リソースの使用を許可する (アクセス許可を付与する) かを制御します。IAM は、追加料金なしで使用できる AWS のサービス です。

トピック

- [オーディエンス](#)
- [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](#)
- [が IAM と AWS Elemental MediaTailor 連携する方法](#)
- [のアイデンティティベースのポリシーの例 AWS Elemental MediaTailor](#)
- [のリソースベースのポリシーの例 AWS Elemental MediaTailor](#)
- [AWS の 管理ポリシー AWS Elemental MediaTailor](#)
- [MediaTailor 向けのサービスリンクロールの使用](#)
- [AWS Elemental MediaTailor ID とアクセスのトラブルシューティング](#)

オーディエンス

AWS Identity and Access Management (IAM) の使用方法は、ロールによって異なります。

- サービスユーザー - 機能にアクセスできない場合は、管理者にアクセス許可をリクエストします (「[AWS Elemental MediaTailor ID とアクセスのトラブルシューティング](#)」を参照)。
- サービス管理者 - ユーザーアクセスを決定し、アクセス許可リクエストを送信します (「[が IAM と AWS Elemental MediaTailor 連携する方法](#)」を参照)
- IAM 管理者 - アクセスを管理するためのポリシーを作成します (「[のアイデンティティベースのポリシーの例 AWS Elemental MediaTailor](#)」を参照)

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。、IAM ユーザー AWS アカウントのルートユーザー、または IAM ロールを引き受けることで認証される必要があります。

AWS IAM アイデンティティセンター (IAM Identity Center)、シングルサインオン認証、Google/Facebook 認証情報などの ID ソースからの認証情報を使用して、フェデレーテッド ID としてサインインできます。サインインの詳細については、「AWS サインイン ユーザーガイド」の「[AWS アカウントにサインインする方法](#)」を参照してください。

プログラムによるアクセスの場合、は SDK と CLI AWS を提供してリクエストを暗号化して署名します。詳細については、「IAM ユーザーガイド」の「[API リクエストに対するAWS 署名バージョン 4](#)」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、まず、すべての AWS のサービス および リソースへの完全なアクセス権を持つ AWS アカウント root ユーザーと呼ばれる 1 つのサインインアイデンティティから始めます。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。ルートユーザー認証情報を必要とするタスクについては、「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、人間のユーザーが一時的な認証情報 AWS のサービス を使用して にアクセスするには、ID プロバイダーとのフェデレーションを使用する必要があります。

フェデレーテッド ID は、エンタープライズディレクトリ、ウェブ ID プロバイダー、または ID Directory Service ソースの認証情報 AWS のサービス を使用して にアクセスするユーザーです。フェデレーテッドアイデンティティは、一時的な認証情報を提供するロールを引き受けます。

アクセスを一元管理する場合は、AWS IAM アイデンティティセンターをお勧めします。詳細については、「AWS IAM アイデンティティセンター ユーザーガイド」の「[IAM アイデンティティセンターとは](#)」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、特定の個人やアプリケーションに対する特定のアクセス許可を持つアイデンティティです。長期認証情報を持つ IAM ユーザーの代わりに一時的な認証情報を使用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「[ID プロバイダーとのフェデレーションを使用して にアクセスする必要がある AWS](#)」を参照してください。

[IAM グループ](#)は、IAM ユーザーの集合を指定し、大量のユーザーに対するアクセス許可の管理を容易にします。詳細については、「IAM ユーザーガイド」の「[IAM ユーザーに関するユースケース](#)」を参照してください。

IAM ロール

[IAM ロール](#)は、特定のアクセス許可を持つアイデンティティであり、一時的な認証情報を提供します。ユーザーから [IAM ロール \(コンソール\) に切り替えるか、または API オペレーションを呼び出すことで、ロール](#)を引き受けることができます。AWS CLI AWS 詳細については、「IAM ユーザーガイド」の「[ロールを引き受けるための各種方法](#)」を参照してください。

IAM ロールは、フェデレーションユーザーアクセス、一時的な IAM ユーザーのアクセス許可、クロスアカウントアクセス、クロスサービスアクセス、および Amazon EC2 で実行するアプリケーションに役立ちます。詳細については、IAM ユーザーガイドの [IAM でのクロスアカウントリソースアクセス](#) 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、ID AWS 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場合のアクセス許可を定義します。は、プリンシパル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。ほとんどのポリシーは JSON ドキュメント AWS として に保存されます。JSON ポリシードキュメントの詳細については、「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、ポリシーを使用して、どのプリンシパルがどのリソースに対して、どのような条件でアクションを実行できるかを定義することで、誰が何にアクセスできるかを指定します。

デフォルトでは、ユーザーやロールにアクセス許可はありません。IAM 管理者は IAM ポリシーを作成してロールに追加し、このロールをユーザーが引き受けられるようにします。IAM ポリシーは、オペレーションの実行方法を問わず、アクセス許可を定義します。

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のポリシーは、アイデンティティ (ユーザー、グループ、またはロール) にアタッチできる JSON アクセス許可ポリシードキュメントです。これらのポリシーは、アイデンティティがどのリソースに対してどのような条件下でどのようなアクションを実行できるかを制御します。アイデンティティベースポリシーの作成方法については、IAM ユーザーガイドの [カスタマー管理ポリシーでカスタム IAM アクセス許可を定義する](#) を参照してください。

アイデンティティベースのポリシーは、インラインポリシー (単一の ID に直接埋め込む) または管理ポリシー (複数の ID にアタッチされたスタンドアロンポリシー) にすることができます。管理ポリシーとインラインポリシーのいずれかを選択する方法については、「IAM ユーザーガイド」の「[管理ポリシーとインラインポリシーのいずれかを選択する](#)」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。例としては、IAM ロール信頼ポリシーや Amazon S3 バケットポリシーなどがあります。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。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。

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。

その他のポリシータイプ

AWS は、より一般的なポリシータイプによって付与されるアクセス許可の上限を設定できる追加のポリシータイプをサポートしています。

- アクセス許可の境界 – アイデンティティベースのポリシーで IAM エンティティに付与することのできるアクセス許可の数の上限を設定します。詳細については、「IAM ユーザーガイド」の「[IAM エンティティのアクセス許可境界](#)」を参照してください。

- サービスコントロールポリシー (SCP) - AWS Organizations内の組織または組織単位の最大のアクセス許可を指定します。詳細については、「AWS Organizations ユーザーガイド」の「[サービスコントロールポリシー](#)」を参照してください。
- リソースコントロールポリシー (RCP) – は、アカウント内のリソースで利用できる最大数のアクセス許可を定義します。詳細については、「AWS Organizations ユーザーガイド」の「[リソースコントロールポリシー \(RCP\)](#)」を参照してください。
- セッションポリシー – ロールまたはフェデレーションユーザーの一時セッションを作成する際にパラメータとして渡される高度なポリシーです。詳細については、「IAM ユーザーガイド」の「[セッションポリシー](#)」を参照してください。

複数のポリシータイプ

1つのリクエストに複数のタイプのポリシーが適用されると、結果として作成されるアクセス許可を理解するのがさらに難しくなります。が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 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。

が IAM と AWS Elemental MediaTailor 連携する方法

IAM を使用して MediaTailor へのアクセスを管理する前に、MediaTailor で使用できる IAM 機能を確認してください。

で使用できる IAM 機能 AWS Elemental MediaTailor

IAM 機能	MediaTailor のサポート
アイデンティティベースのポリシー	あり
リソースベースのポリシー	はい
ポリシーアクション	あり
ポリシーリソース	いいえ
ポリシー条件キー (サービス固有)	はい
ACL	なし

IAM 機能	MediaTailor のサポート
ABAC (ポリシー内のタグ)	部分的
一時認証情報	あり
プリンシパルアクセス権限	あり
サービスロール	いいえ
サービスリンクロール	はい

MediaTailor 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については、「IAM ユーザーガイド」の[AWS 「IAM と連携する のサービス」](#)を参照してください。

MediaTailor の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のポリシーのサポート: あり

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 許可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ポリシーの作成方法については、「IAM ユーザーガイド」の「[カスタマー管理ポリシーでカスタム IAM アクセス許可を定義する](#)」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。JSON ポリシーで使用するすべての要素について学ぶには、「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。

MediaTailor のアイデンティティベースのポリシーの例

MediaTailor 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の[アイデンティティベースのポリシーの例 AWS Elemental MediaTailor](#)。

MediaTailor 内のリソースベースのポリシー

リソースベースのポリシーのサポート: あり

MediaTailor サービスは、1 種類のリソースベースのポリシーのみをサポートします。チャンネルポリシーはチャンネルにアタッチされているため、チャンネルポリシーと呼ばれます。このポリシーは、チャンネルでアクションを実行できるプリンシパルを定義します。

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コントロールできます。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、[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、全体のアカウント、または別のアカウントの IAM エンティティを、リソースベースのポリシーのプリンシパルとして指定します。詳細については、IAM ユーザーガイドの[IAM でのクロスアカウントリソースアクセス](#)を参照してください。

リソースベースのポリシーをチャンネルにアタッチする方法については、「」を参照してください [MediaTailor コンソールを使用してチャンネルを作成する](#)。

MediaTailor 内のリソースベースのポリシーの例

MediaTailor リソース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 [リソースベースのポリシーの例 AWS Elemental MediaTailor](#)。

MediaTailor のポリシーアクション

ポリシーアクションのサポート: あり

管理者は JSON AWS 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対してどのような条件下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。このアクションは関連付けられたオペレーションを実行するためのアクセス許可を付与するポリシーで使用されます。

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

MediaTailor 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を使用します。

```
mediatailor
```

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": [  
    "mediatailor:action1",  
    "mediatailor:action2"  
]
```

MediaTailor 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。[のアイデンティティベースのポリシーの例 AWS Elemental MediaTailor](#)。

MediaTailor のポリシーリソース

ポリシーリソースのサポート: なし

AWS Elemental MediaTailor では、ポリシーでのリソース ARNs の指定はサポートされていません。

MediaTailor のポリシー条件キー

サービス固有のポリシー条件キーのサポート: あり

管理者は JSON AWS 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対してどのような条件下でアクションを実行できるかということです。

Condition 要素は、定義された基準に基づいてステートメントが実行される時期を指定します。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して、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。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。

MediaTailor 条件キーのリストについては、「サービス認可リファレンス」の「[の条件キー AWS Elemental MediaTailor](#)」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、「[で定義されるアクション AWS Elemental MediaTailor](#)」を参照してください。

AWS Elemental MediaTailor はサービス固有の条件キーを提供しませんが、一部のグローバル条件キーの使用をサポートしています。すべての AWS グローバル条件キーを確認するには、「AWS

Identity and Access Management 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。

MediaTailor ACLs

ACL のサポート: なし

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アクセス許可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。

MediaTailor での ABAC

ABAC (ポリシー内のタグ) のサポート: 一部

属性ベースのアクセス制御 (ABAC) は、タグと呼ばれる属性に基づいてアクセス許可を定義する認可戦略です。IAM エンティティと AWS リソースにタグをアタッチし、プリンシパルのタグがリソースのタグと一致するときにオペレーションを許可するように ABAC ポリシーを設計できます。

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ありです。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 認可でアクセス許可を定義する](#)」を参照してください。ABAC をセットアップする手順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「[属性ベースのアクセス制御 \(ABAC\) を使用する](#)」を参照してください。

MediaTailor には、Partial の値を使用します。

MediaTailor での一時的な認証情報の使用

一時的な認証情報のサポート: あり

一時的な認証情報は AWS、リソースへの短期的なアクセスを提供し、フェデレーションまたはスウィッチロールの使用時に自動的に作成されます。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 AWS 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「[IAM の一時的な認証情報](#)」および「[AWS のサービスと IAM との連携](#)」を参照してください。

MediaTailor のクロスサービスプリンシパルアクセス許可

転送アクセスセッション (FAS) のサポート: あり

転送アクセスセッション (FAS) は、 を呼び出すプリンシパルのアクセス許可と AWS のサービス、ダウストリームサービス AWS のサービス へのリクエストをリクエストする を使用します。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。

MediaTailor のサービスロール

サービスロールのサポート: なし

AWS Elemental MediaTailor はサービスロールをサポートしていません。

MediaTailor のサービスにリンクされたロール

サービスリンクロールのサポート: あり

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、サービスリンクロールのアクセス許可を表示できますが、編集することはできません。

MediaTailor 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、「 」を参照してください [MediaTailor 向けのサービスリンクロールの使用](#)。

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

デフォルトでは、ユーザーとロールには MediaTailor リソースを作成または変更するアクセス許可はありません。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。

これらのサンプルの 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 ポリシーを作成する \(コンソール\)](#)」を参照してください。

各リソースタイプの ARN の形式など、MediaTailor 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、「サービス認可リファレンス」の「[のアクション、リソース、および条件キー AWS Elemental MediaTailor](#)」を参照してください。ARNs

トピック

- [ポリシーに関するベストプラクティス](#)
- [MediaTailor コンソールの使用](#)
- [自分の権限の表示をユーザーに許可する](#)

ポリシーに関するベストプラクティス

ID ベースのポリシーは、アカウント内で誰かが MediaTailor リソースを作成、アクセス、または削除できるかどうかを決定します。これらのアクションでは、AWS アカウントに費用が発生する場合があります。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 – ユーザーとワークロードにアクセス許可の付与を開始するには、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。これらはで使えます 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することで、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 [ジョブ機能のAWS マネージドポリシー](#) を参照してください。
- 最小特権を適用する – IAM ポリシーでアクセス許可を設定する場合は、タスクの実行に必要な許可のみを付与します。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、最小特権アクセス許可とも呼ばれています。IAM を使用して許可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 でのポリシーとアクセス許可](#) 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。たとえば、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、サービスアクションがなどの特定のを通じて使用されている場合に 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 CloudFormation。詳細については、IAM ユーザーガイドの [IAM JSON ポリシー要素:条件](#) を参照してください。
- IAM アクセスアナライザーを使用して IAM ポリシーを検証し、安全で機能的な権限を確保する – IAM アクセスアナライザーは、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 でポリシーを検証する](#) 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、MFA をオンにしてセキュリティを強化します。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 を使用した安全な API アクセス](#) 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。

MediaTailor コンソールの使用

AWS Elemental MediaTailor 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 MediaTailor 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引き続き MediaTailor コンソールを使用できるようにするには、MediaTailor *ConsoleAccess* または *ReadOnly* AWS 管理ポリシーもエンティティにアタッチします。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": "*"
}
]
}

```

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

リソースベースのポリシーをチャンネルにアタッチする方法については、「」を参照してください [MediaTailor コンソールを使用してチャンネルを作成する](#)。

トピック

- [匿名アクセス](#)
- [クロスアカウントアクセス](#)

匿名アクセス

次のAllowポリシーを検討してください。このポリシーを有効にすると、MediaTailor はポリシー内のチャンネルリソースに対する mediatailor:GetManifest アクションへの匿名アクセスを許可します。これは、*region* が AWS リージョン、*accountID* が AWS アカウント ID、*channelName* がチャンネルリソースの名前である場合に発生します。

クロスアカウントアクセス

次のAllowポリシーを検討してください。このポリシーを有効にすると、MediaTailor はアカウント間でポリシー内のチャンネルリソースに対する `mediatailor:GetManifest` アクションを許可します。これは、**region** が AWS リージョン、**accountID** が AWS アカウント ID、**channelName** がチャンネルリソースの名前である場合に発生します。

AWS の 管理ポリシー AWS Elemental MediaTailor

ユーザー、グループ、ロールにアクセス許可を追加するには、自分でポリシーを記述するよりも AWS 管理ポリシーを使用する方が簡単です。チームに必要な権限のみを提供する [IAM カスタマーマネージドポリシーを作成する](#) には時間と専門知識が必要です。すぐに開始するには、AWS マネージドポリシーを使用できます。これらのポリシーは、一般的なユースケースをターゲット範囲に含めており、AWS アカウントで利用できます。AWS 管理ポリシーの詳細については、IAM ユーザーガイドの「[AWS 管理ポリシー](#)」を参照してください。

AWS サービスは、AWS 管理ポリシーを維持および更新します。AWS 管理ポリシーのアクセス許可は変更できません。サービスでは新しい機能を利用できるようにするために、AWS マネージドポリシーに権限が追加されることがあります。この種類の更新はポリシーがアタッチされている、すべてのアイデンティティ (ユーザー、グループおよびロール) に影響を与えます。新しい機能が立ち上げられた場合や、新しいオペレーションが使用可能になった場合に、各サービスが AWS マネージドポリシーを更新する可能性が最も高くなります。サービスは AWS マネージドポリシーからアクセス許可を削除しないため、ポリシーの更新によって既存のアクセス許可が損なわれることはありません。

さらに、は、複数のサービスにまたがるジョブ関数の マネージドポリシー AWS をサポートしています。例えば、ReadOnlyAccess AWS 管理ポリシーは、すべての AWS サービスとリソースへの読み取り専用アクセスを提供します。サービスが新機能を起動すると、は新しいオペレーションとリソースの読み取り専用アクセス許可 AWS を追加します。ジョブ機能のポリシーの一覧および詳細については、「IAM ユーザーガイド」の「[AWS のジョブ機能のマネージドポリシー](#)」を参照してください。

AWS マネージドポリシー: AWSElementalMediaTailorFullAccess

AWSElementalMediaTailorFullAccess ポリシーを IAM アイデンティティにアタッチできます。これは、再生設定と、プログラムやチャンネルなどのチャンネルアセンブリリソースを作成および管理する必要があるユーザーにとって便利なポリシーです。このポリシーは、へのフルアクセスを

許可するアクセス許可を付与します AWS Elemental MediaTailor。これらのユーザーは、MediaTailor リソースの作成、更新、および削除を実行できます。

このポリシーのアクセス許可を確認するには、「AWS マネージドポリシーリファレンス」の[AWSElementalMediaTailorFullAccess](#)」を参照してください。

AWS マネージドポリシー: AWSElementalMediaTailorReadOnly

AWSElementalMediaTailorReadOnly ポリシーを IAM アイデンティティにアタッチできます。これは、再生設定と、プログラムやチャンネルなどのチャンネルアセンブリリソースを表示する必要があるユーザーにとって便利なポリシーです。このポリシーは、への読み取り専用アクセスを許可するアクセス許可を付与します AWS Elemental MediaTailor。これらのユーザーが、MediaTailor リソースの作成、更新、または削除を実行することはできません。

このポリシーのアクセス許可を確認するには、「AWS マネージドポリシーリファレンス」の[AWSElementalMediaTailorReadOnly](#)」を参照してください。

AWS 管理ポリシーに対する MediaTailor の更新

このサービスがこれらの変更の追跡を開始してからの MediaTailor の AWS マネージドポリシーの更新に関する詳細を表示します。このページへの変更に関する自動アラートを受けるには、MediaTailor [のドキュメント履歴 AWS Elemental MediaTailor](#)の RSS フィードをサブスクライブしてください。

変更	説明	日付
MediaTailor が新しいマネージドポリシーを追加	MediaTailor が、以下のマネージドポリシーを追加しました。 AWSElementalMediaTailorReadOnly AWSElementalMediaTailorFullAccess	2021 年 11 月 24 日
MediaTailor が変更の追跡を開始	MediaTailor は AWS 、管理ポリシーの変更の追跡を開始しました。	2021 年 11 月 24 日

MediaTailor 向けのサービスリンクロールの使用

AWS Elemental MediaTailor は AWS Identity and Access Management (IAM) [サービスにリンクされたロール](#)を使用します。サービスリンクロールは、MediaTailor. に直接リンクされた特殊なタイプの IAM ロールです。サービスにリンクされたロールは MediaTailor によって事前定義されており、サービスがユーザーに代わって他の AWS サービスを呼び出すために必要なすべてのアクセス許可が含まれています。

必要な許可を手動で追加する必要がないため、サービスリンクロールは MediaTailor のセットアップを容易にします。サービスリンクロールの許可は MediaTailor が定義し、別段の定義がない限り、MediaTailor のみがそのロールを引き受けることができます。定義される許可は信頼ポリシーと許可ポリシーに含まれており、その許可ポリシーを他の IAM エンティティにアタッチすることはできません。

サービスリンクロールを削除するには、最初に関連リソースを削除する必要があります。これは、リソースにアクセスするための許可を誤って削除できないため、MediaTailor リソースを保護します。

サービスリンクロールをサポートする他のサービスについては、[IAM と連携するAWS のサービス](#)を参照の上、サービスリンクロール列がはいになっているサービスを検索してください。サービスにリンクされた役割に関するドキュメントをサービスで表示するには[はい] リンクを選択してください。

MediaTailor サービスリンクロールがサポートされるリージョン

MediaTailor は、このサービスを利用できるすべてのリージョンでサービスリンクロールの使用をサポートします。詳細については、「[AWS リージョンとエンドポイント](#)」を参照してください。

トピック

- [MediaTailor のサービスリンクロール許可](#)
- [MediaTailor のサービスリンクロールの作成](#)
- [MediaTailor のサービスリンクロールの編集](#)
- [MediaTailor のサービスリンクロールの削除](#)

MediaTailor のサービスリンクロール許可

MediaTailor は、AWSServiceRoleForMediaTailor という名前のサービスリンクロールを使用します。MediaTailor はこのサービスリンクロールを使用して CloudWatch を呼び出し、ロググループ、ログストリーム、およびログイベントの作成と管理を行います。このサービスリンクロールは、マネージドポリシー AWSMediaTailorServiceRolePolicy にアタッチされます。

AWSServiceRoleForMediaTailor サービスリンクロールは、以下のサービスを信頼してロールを引き受けます。

- `mediatailor.amazonaws.com`

ロールの許可ポリシーは、指定されたリソースに対して以下のアクションを実行することを MediaTailor に許可します。

- アクション: `arn:aws:logs:*:*:log-group:/aws/MediaTailor/*:log-stream:*` での `logs:PutLogEvents`
- アクション: `logs:CreateLogStream`, `logs:CreateLogGroup`, `logs:DescribeLogGroups`, `logs:DescribeLogStreams` 上で `arn:aws:logs:*:*:log-group:/aws/MediaTailor/*`

サービスにリンクされたロールの作成、編集、削除を IAM エンティティ (ユーザー、グループ、ロールなど) に許可するには、権限を設定する必要があります。詳細については、IAM ユーザーガイドの「[サービスリンクロールの許可](#)」を参照してください。

MediaTailor のサービスリンクロールの作成

サービスリンクロールを手動で作成する必要はありません。AWS マネジメントコンソール、AWS Command Line Interface (AWS CLI)、または AWS API でセッションログ記録を有効にすると、MediaTailor によってサービスにリンクされたロールが作成されます。

Important

このサービスリンクロールはこのロールでサポートされている機能を使用する別のサービスでアクションが完了した場合にアカウントに表示されます。MediaTailor がサービスリンクロールのサポートを開始した 2021 年 9 月 15 日より前にこのサービスを使用していた場合、MediaTailor によってアカウントに AWSServiceRoleForMediaTailor ロールが作成されています。詳細については、「[IAM アカウントに新しいロールが表示される](#)」を参照してください。

このサービスリンクロールを削除した後で再度作成する必要がある場合は、同じ手順でアカウントにロールを再作成できます。セッションロギングを有効にすると、MediaTailor が再度ユーザーのためにサービスリンクロールを作成します。

IAM コンソールを使用して、MediaTailor ユースケースでサービスリンクロールを作成することもできます。AWS CLI または AWS API で、サービス名を使用して `mediatailor.amazonaws.com` サービスにリンクされたロールを作成します。詳細については、「IAM ユーザーガイド」の「[サービスにリンクされたロールの作成](#)」を参照してください。このサービスリンクロールを削除しても、同じ方法でロールを再作成できます。

MediaTailor のサービスリンクロールの編集

MediaTailor は、`AWSServiceRoleForMediaTailor` サービスリンクロールの編集を許可しません。サービスリンクロールの作成後は、さまざまなエンティティがロールを参照する可能性があるため、ロール名を変更することはできません。ただし、IAM を使用してロールの説明を編集することはできます。詳細については、IAM ユーザーガイドの「[サービスにリンクされたロールの編集](#)」を参照してください。

MediaTailor のサービスリンクロールの削除

サービスリンクロールを必要とする機能やサービスが不要になった場合は、ロールを削除することをお勧めします。そうすることで、積極的にモニタリングまたは保守されていない未使用のエンティティを排除できます。ただし、手動で削除する前に、サービスリンクロールのリソースをクリーンアップする必要があります。

Note

リソースをクリーンアップしようとするときに MediaTailor サービスがロールが使用している場合、削除が失敗する可能性があります。失敗した場合は数分待ってから操作を再試行してください。

AWSServiceRoleForMediaTailor が使用する MediaTailor リソースをクリーンアップする

- ログ設定用に MediaTailor が作成したサービスリンクロールを削除するには、まずアカウント内のすべてのログ設定を非アクティブにする必要があります。ログ設定を非アクティブ化するには、有効な割合値を 0 に設定します。そうすることで、対応する再生設定のすべてのセッションログインがオフになります。詳細については、「[ログ設定の非アクティブ化](#)」を参照してください。

サービスリンクロールを IAM で手動削除するには

IAM コンソール、AWS Command Line Interface (AWS CLI)、または AWS API を使用して、AWSServiceRoleForMediaTailor サービスにリンクされたロールを削除します。詳細については、「[IAM ユーザーガイド](#)」の「サービスリンクロールの削除」を参照してください。

AWS Elemental MediaTailor ID とアクセスのトラブルシューティング

以下の情報は、MediaTailor と 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と修正に役立ちます。

トピック

- [MediaTailor でアクションを実行する権限がありません](#)
- [iam:PassRole を実行する権限がありません](#)
- [自分の 以外のユーザーに MediaTailor リソース AWS アカウント へのアクセスを許可したい](#)
- [MediaTailor が Amazon S3 バケットにアクセスできない](#)

MediaTailor でアクションを実行する権限がありません

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、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 mediatailor:*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:
mediatailor:GetWidget on resource: my-example-widget
```

この場合、mediatailor:*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。サインイン認証情報を提供した担当者が管理者です。

iam:PassRole を実行する権限がありません

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 MediaTailor にロールを渡すことができるようにする必要があります。

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、既存のロールをそのサービスに渡すことができます。そのためには、サービスにロールを渡すアクセス許可が必要です。

次の例のエラーは、という IAM ユーザーがコンソールを使用して MediaTailor marymajor でアクションを実行しようとするとき発生します。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アクセス許可が必要です。Mary には、ロールをサービスに渡すアクセス許可がありません。

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

この場合、Mary のポリシーを更新してメアリーに 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 MediaTailor 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、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:

- MediaTailor がこれらの機能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 [が IAM と AWS Elemental MediaTailor 連携する方法](#)。
- 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 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 [「サードパーティーが所有する へのアクセスを提供する AWS アカウント」](#) 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 \(ID フェデレーション\) へのアクセスの許可](#) を参照してください。

- 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使用方法の違いについては、「IAM ユーザーガイド」の「[IAM でのクロスアカウントのリソースへのアクセス](#)」を参照してください。

MediaTailor が Amazon S3 バケットにアクセスできない

Amazon S3 バケットからコンテンツにアクセスしようとしたときに MediaTailor がエラーを返す場合は、次の一般的な原因を確認してください。

Amazon S3 アクセス許可の問題

エラー症状:

- 502 BadGatewayException Amazon S3 関連メッセージを含む
- 403 Forbidden MediaTailor ログのエラー
- Amazon S3 オリジンからのコンテンツのロードに失敗する

解決策:

1. MediaTailor サービスロールに必要な Amazon S3 アクセス許可があることを確認します。
2. Amazon S3 バケットポリシーで MediaTailor アクセスが許可されていることを確認します。

HTTPS アクセスの問題

エラー症状:

- SSL/TLS ハンドシェイクの失敗
- 502 Bad Gateway HTTPS Amazon S3 URL にアクセスする場合 URLs

解決策:

1. Amazon S3 バケットポリシーで HTTPS が要求されていることを確認します。
 2. MediaTailor 設定が Amazon S3 コンテンツの HTTPS URLs を使用していることを確認します。
- Amazon S3

クロスアカウント Amazon S3 アクセス

エラー症状:

- Access Denied Amazon S3 バケットが異なるアカウントにある場合のエラー
- MediaTailor はクロスアカウントロールを引き受けることができません

解決策:

1. MediaTailor のサービスロールを信頼するようにコンテンツアカウントの Amazon S3 バケットポリシーを更新する
2. MediaTailor のサービスロールにクロスアカウントアクセス許可を追加する
3. 両方のアカウントに適切な信頼関係が設定されていることを確認します。

のコンプライアンス検証 AWS Elemental MediaTailor

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、[AWS のサービス「コンプライアンスプログラムによる対象範囲内」](#)の「コンプライアンス」を参照して、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、[「Downloading Reports in AWS Artifact」](#)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。を使用する際のコンプライアンス責任の詳細については AWS のサービス、[AWS「セキュリティドキュメント」](#)を参照してください。

の耐障害性 AWS Elemental MediaTailor

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。は、低レイテンシー、高スループット、高度に冗長なネットワークで接続された、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ー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、フォールトトレランス、および拡張性が優れています。

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。

MediaTailor は、AWS グローバルインフラストラクチャに加えて、データの耐障害性とバックアップの二重をサポートするのに役立ついくつかの機能を提供しています。

MediaTailor のインフラストラクチャセキュリティ

マネージドサービスである AWS Elemental MediaTailor 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 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環境を AWS 設計するには、「Security Pillar AWS Well-Architected Framework」の「[Infrastructure Protection](#)」を参照してください。

AWS 公開された API コールを使用して、ネットワーク経由で MediaTailor にアクセスします。クライアントは次をサポートする必要があります。

- Transport Layer Security (TLS)。TLS 1.2 が必須で、TLS 1.3 をお勧めします。
- 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。

これらの API オペレーションは任意のネットワークの場所から呼び出すことができますが、MediaTailor はソース IP アドレスに基づく制限を含むリソースベースのアクセスポリシーをサポートしています。MediaTailor ポリシーを使用して、特定の Amazon Virtual Private Cloud (Amazon VPC) エンドポイントまたは特定の VPCs からのアクセスを制御することもできます。これにより、実質的に、ネットワーク内の特定の VPC からのみ、特定の MediaTailor リソースへの AWS ネットワークアクセスが分離されます。

サービス間での混乱した使節の防止

混乱した使節の問題は、あるアクションを実行する許可を持たないエンティティが、そのアクションを実行するようにより高い特権のエンティティを強制できるというセキュリティ問題です。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、AWS では、アカウント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を提供しています。

リソースポリシーで [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、別のサービス AWS Elemental MediaTailor に付与するアクセス許可をリソースに制限することをお勧めします。グローバル条件コンテキストキーの両方を使用しており、それらが同じポリシーステートメントで使用される場合、aws:SourceAccount 値と aws:SourceArn 値のアカウントが同じアカウント ID を使用する必要があります。

aws:SourceArn の値は、お使いのリージョンおよびアカウント内の、MediaTailor が CloudWatch のログを発行する対象再生設定である必要があります。ただし、これが適用されるのは、MediaTailor が Amazon CloudWatch のログをアカウントに発行できるようにする [MediaTailorLogger](#) ロールを使用している場合のみです。[サービスリンクロール](#) を使用して MediaTailor が CloudWatch のログを発行できるようにしている場合には適用されません。

混乱した使節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 で 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は、aws:SourceArn グローバルコンテキスト条件キーを使用して、ARN の未知部分をワイルドカード (*) で表します。例えば、arn:aws:*servicename*::123456789012:* などです。

以下は、混乱した使節の問題を防止するために、MediaTailor で 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する方法の例です。

MediaTailor でのログ記録とモニタリング

このセクションでは、セキュリティ上の目的で AWS Elemental MediaTailor 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。MediaTailor でのログ記録およびモニタリングの詳細については、「[AWS Elemental MediaTailor リソースのモニタリングとタグ付け](#)」を参照してください。

モニタリングは、および AWS Elemental MediaTailor 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、MediaTailor 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いくつかのツールが用意されています。

Amazon CloudWatch アラーム

Amazon CloudWatch アラームを使用して、指定した期間にわたって 1 つのメトリクスを確認します。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 (自動スケーリング) ポリシーに通知が送信されます。CloudWatch アラームは、特定の状態にあるという理由では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、「[the section called “CloudWatch メトリクスによるモニタリング”](#)」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 MediaTailor。CloudTrail が収集した情報を使用して、MediaTailor に対して行われたリクエスト、リクエスト元の IP アドレス、リクエスト者、リクエスト日時、および追加の詳細情報を確認できます。詳細については、「[AWS Elemental MediaTailor API コールの記録](#)」を参照してください。

AWS Trusted Advisor

Trusted Advisor は、数十万の AWS 顧客へのサービス提供から学んだベストプラクティスを活用します。 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行います。AWS のすべてのお客様は、5 つの Trusted Advisor チェックにアクセスできます。ビジネスまたはエンタープライズサポートプランをご利用のお客様は、すべての Trusted Advisor チェックを表示できます。

詳細については、「[AWS Trusted Advisor](#)」を参照してください。

AWS Elemental MediaTailor リソースのモニタリングとタグ付け

モニタリングは、およびその他の AWS Elemental MediaTailor 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上で重要な部分です。には、MediaTailor をモニタリングし、問題が発生したときに報告し、必要に応じて自動アクションを実行するための以下のモニタリングツール AWS が用意されています。

- Amazon CloudWatch は、AWS リソースと で実行しているアプリケーションを AWS 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。例えば、CloudWatch で Amazon EC2 インスタンスの CPU 使用率などのメトリクスを追跡し、必要に応じて新しいインスタンスを自動的に起動できます。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 Logs を使用すると、広告決定サーバー (ADS) とのすべてのやり取りからログファイルをモニタリング、保存、アクセスできます。AWS Elemental MediaTailor は、広告リクエスト、リダイレクト、レスポンス、およびレポートリクエストとレスポンスのログを出力します。ADS およびオリジンサーバーからのエラーも、Amazon CloudWatch 内のロググループに発行されます。MediaTailor は、スキップされた広告とスキップされた理由に関する情報も提供します。詳細については、「[広告スキップのトラブルシューティング](#)」を参照してください。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できます。一般的な情報については、[Amazon CloudWatch Logs ユーザーガイド](#)を参照してください。ADS ログに関する情報と、Amazon CloudWatch Logs Insights を使用した分析のためにこのログにアクセスする方法については、「[AWS Elemental MediaTailor Amazon CloudWatch Logs Insights での ADS ログ分析](#)」を参照してください。

トピック

- [AWS Elemental MediaTailor ログの表示](#)
- [Amazon CloudWatch メトリクス AWS Elemental MediaTailor によるモニタリング](#)
- [AWS Elemental MediaTailor API コールの記録](#)
- [AWS Elemental MediaTailor チャンネルアセンブリアラートの受信](#)
- [AWS Elemental MediaTailor リソースのタグ付け](#)
- [ワークフローモニターを使用した AWS メディアサービスのモニタリング](#)

AWS Elemental MediaTailor ログの表示

MediaTailor は、チャンネルと再生設定のさまざまなマイルストーンとアクティビティを記述するログを出力します。これらのログを使用して、ワークフローを可視化し、サービスの問題をトラブルシューティングできます。以下のトピックでは、ログとログ記録オプションについて説明します。

トピック

- [AWS Elemental MediaTailor ADS ログの説明とイベントタイプ](#)
- [AWS Elemental MediaTailor マニフェストログの説明とイベントタイプ](#)
- [AWS Elemental MediaTailor トランスコードログの説明とイベントタイプ](#)
- [提供されたログを使用して AWS Elemental MediaTailor ログを送信する](#)
- [Amazon CloudWatch Logs に直接 AWS Elemental MediaTailor ログを書き込む](#)
- [AWS Elemental MediaTailor ログのボリュームの制御](#)
- [AWS Elemental MediaTailor ログとイベントのフィルタリング](#)
- [AWS Elemental MediaTailor デバッグログの生成](#)

AWS Elemental MediaTailor ADS ログの説明とイベントタイプ

以下のセクションでは、MediaTailor が広告決定サーバー (ADS) でイベントを記述するために出力するログについて説明します。これらはAdDecisionServerInteractionsログです。

トピック

- [AdDecisionServerInteractions のイベント](#)
- [ADS ログの説明](#)
- [ADS ログの JSON スキーマ](#)

AdDecisionServerInteractions のイベント

MediaTailor と広告決定サーバー (ADS) とのやり取り中に、次のイベントが出力されます。

Log	説明
AD_MARKER_FOUND	MediaTailor はマニフェストで広告マーカースを見つけました。

Log	説明
BEACON_FIRED	追跡ビーコンが起動され、広告イベントが報告されました。サーバー側のレポートモード (デフォルト) では、MediaTailor はビーコンを起動します。クライアント側のレポートモードでは、再生デバイスがビーコンを起動します。
EMPTY_VAST_RESPONSE	ADS は、ゼロ広告を含む空の VAST レスポンスを返しました。
EMPTY_VMAP_RESPONSE	ADS は空の VMAP レスポンスを返しました。
ERROR_ADS_INVALID_RESPONSE	ADS は 200 以外のステータスコードを返しました。
ERROR_ADS_IO	MediaTailor が ADS との通信中にエラーが発生しました。
ERROR_ADS_RESPONSE_PARSE	MediaTailor で ADS レスポンスの解析中にエラーが発生しました。
ERROR_ADS_RESPONSE_UNKNOWN_ROOT_ELEMENT	ADS レスポンスに無効なルート要素が含まれています。
ERROR_ADS_TIMEOUT	ADS への MediaTailor リクエストがタイムアウトしました。
ERROR_DISALLOWED_HOST	ADS ホストは許可されていません。
ERROR_FIRING_BEACON_FAILED	MediaTailor が追跡ビーコンの発射に失敗しました。
ERROR_PERSONALIZATION_DISABLED	このセッションでは、広告挿入は無効になっています。
ERROR_UNKNOWN	MediaTailor で ADS リクエスト中に不明なエラーが発生しました。
ERROR_UNKNOWN_HOST	ADS ホストが不明です。

Log	説明
ERROR_VAST_INVALID_MEDIA_FILE	VAST に無効または欠落しているMediaFile 要素Adがあります。
ERROR_VAST_INVALID_VAST_AD_TAG_URI	VAST レスポンスに無効な が含まれてい ますVASTAdTagURI 。
ERROR_VAST_MISSING_CREATIVES	VAST にはゼロまたは複数のCreatives 要 素Adが含まれています。1 つのCreatives 要 素が必要です。
ERROR_VAST_MISSING_IMPRESSION	VAST にはゼロImpression 要素Adが含まれ ています。少なくとも 1 つのImpression 要 素が必要です。
ERROR_VAST_MISSING_MEDIAFILES	VAST にはゼロまたは複数のMediaFiles 要 素Adが含まれています。1 つのMediaFile s 要素が必要です。
ERROR_VAST_MISSING_OVERLAYS	MediaTailor は、広告サーバーから非線形クリ エイティブを受け取りませんでした。
ERROR_VAST_MULTIPLE_LINEAR	VAST には複数のLinear要素Adが含まれてい ます。
ERROR_VAST_MULTIPLE_TRACKING_EVENTS	VAST には複数のTrackingEvents 要素Adが 含まれています。
ERROR_VAST_REDIRECT_EMPTY_RESPONSE	VAST リダイレクトリクエストが空のレスポンスを返しました。
ERROR_VAST_REDIRECT_FAILED	VAST リダイレクトリクエストでエラーが発生 しました。
ERROR_VAST_REDIRECT_MULTIPLE_VAST	VAST リダイレクトリクエストは複数の広告を 返しました。
FILLED_AVAIL	MediaTailor は avail を正常に入力しました。

Log	説明
FILLED_OVERLAY_AVAIL	MediaTailor はオーバーレイ表示を正常に塗りつぶしました。
INTERSTITIAL_VOD_FAILURE	ADS リクエストまたはレスポンスで、VOD プレイリストのインタースティシャル表示のフィル中に問題が発生しました。広告は挿入されません。
INTERSTITIAL_VOD_SUCCESS	MediaTailor は、VOD プレイリストのインタースティシャル表示を正常に入力しました。
MAKING_ADS_REQUEST	MediaTailor は ADS に広告をリクエストしています。
MODIFIED_TARGET_URL	MediaTailor がアウトバウンドターゲット URL を変更しました。
NON_AD_MARKER_FOUND	MediaTailor はマニフェストで実行不可能な広告マーカースを見つけました。
PRE_ADS_REQUEST_FUNCTION_COMPLETED	pre-ads リクエストフックの個々の関数が完了しました。これはオプトインイベントタイプです。
PRE_ADS_REQUEST_FUNCTION_ERROR	pre-ads リクエストフックの個々の関数が失敗しました。
PRE_ADS_REQUEST_HOOK_ERROR	広告前のリクエストフックの実行に失敗しました。
PRE_ADS_REQUEST_HOOK_SUMMARY	成功またはエラーステータスを含む、広告前のリクエストフック実行の概要。これはオプトインイベントタイプです。
RAW_ADS_RESPONSE	MediaTailor は raw ADS レスポンスを受信しました。

Log	説明
REDIRECTED_VAST_RESPONSE	MediaTailor は、VAST リダイレクトの後、VAST レスポンスを受信しました。
VAST_REDIRECT	VAST 広告レスポンスにはリダイレクトが含まれています。
VAST_RESPONSE	MediaTailor は VAST レスポンスを受信しました。
VOD_TIME_BASED_AVAIL_PLAN_SUCCESS	MediaTailor は、VOD テンプレートの時間ベースの可用性プランを正常に作成しました。
VOD_TIME_BASED_AVAIL_PLAN_VAST_RESPONSE_FOR_OFFSET	MediaTailor は、VOD テンプレートの時間ベースの可用性プランを作成しています。MediaTailor は、タイムオフセットの VAST レスポンスを受信しました。
VOD_TIME_BASED_AVAIL_PLAN_WARNING_NO_ADVERTISEMENTS	ADS リクエストまたはレスポンスで、VOD テンプレートの時間ベースの可用性プランの作成中に問題が発生しました。広告は挿入されません。
WARNING_NO_ADVERTISEMENTS	MediaTailor で、表示のフィル中に問題が発生しました。広告は挿入されません。
WARNING_URL_VARIABLE_SUBSTITUTION_FAILED	MediaTailor は、ADS URL の動的変数を置き換えることはできません。URL 設定を確認します。
WARNING_VPAID_AD_DROPPED	スレートの欠落により VPAID 広告がドロップされたか、セッションがサーバー側のレポートを使用します。

ADS ログの説明

このセクションでは、ADS ログの説明の構造と内容について説明します。JSON エディタで独自に調べるには、[the section called “ADS ログの JSON スキーマ”](#) のリストを使用します。

ADS ログの各イベントには、CloudWatch Logs によって生成された標準フィールドが含まれています。詳細については、「[CloudWatch Logs Insights を使用したログデータの分析](#)」を参照してください。

ADS ログのプロパティ

このセクションでは、ADS ログのプロパティについて説明します。

プロパティ	タイプ	必須	説明
adsRequestUrl	string	false	MediaTailor によって行われた ADS リクエストの完全な URL。
avail	avail 型のオブジェクト	false	MediaTailor が広告で埋め込む avail に関する情報。現在、FILLED_AVAIL イベントタイプの場合、これは MediaTailor が最初に avail を検出したときに MediaTailor によって作成されたプランです。最終的に avail がどのように満たされるかは、コンテンツの再生方法に応じて、このプランとは異なる場合があります。
awsAccountId	string	true	セッションに使用された MediaTailor 設定

プロパティ	タイプ	必須	説明
			の AWS アカウント ID。
customerId	string	true	複数のログエントリを関連付けるために使用できる AWS アカウント ID のハッシュ化されたバージョン。
eventDescription	string	true	MediaTailor サービスによって提供される、このログメッセージをトリガーしたイベントの簡単な説明。デフォルトでは、これは空です。例えば、Got VAST response などです。
eventTimestamp	string	true	リクエストの日付と時刻。
eventType	string	true	このログメッセージをトリガーしたイベントのコード。例えば、VAST_RESPONSE などです。

プロパティ	タイプ	必須	説明
errorType	string	false	構造化エラー分類。PRE_ADS_REQUEST_HO OK_ERROR およ び PRE_ADS_REQUEST_FU NCTION_ERROR イベントに存在します。値: INTERNAL_ERROR、SYNTAX_ERROR、RESOURCE_LIMIT_ERROR、RESTRICTION_ERROR、TIMEOUT_ERROR、VALIDATION_ERROR。
cause	string	false	特定の障害を説明するエラー詳細文字列。PRE_ADS_REQUEST_HO OK_ERROR およ び PRE_ADS_REQUEST_FU NCTION_ERROR イベントに存在します。

プロパティ	タイプ	必須	説明
eventId	string	false	フック呼び出しあたりの一意の識別子。このフィールドを使用して、同じ実行のフックレベルイベントと関数レベルのイベントを関連付けます。Functions イベントにのみ存在します。
executionTimeMs	整数	false	ミリ秒単位の実行時間。Functions イベントにのみ存在します。
functionId	string	false	関数の識別子。フックイベントの場合、これはフックで設定された関数です。関数イベントでは、これはエグゼキューター内の個々の関数です。Functions イベントにのみ存在します。
functionType	string	false	関数のタイプ: CUSTOM_OUTPUT 、 HTTP_REQUEST 、 SEQUENTIAL_EXECUTOR 、 または CONCURRENT_EXECUTOR 。関数レベルのイベントにのみ存在します。

プロパティ	タイプ	必須	説明
http	オブジェクト	false	HTTP リクエストとレスポンスの詳細。関数のみのHTTP_REQUEST 関数レベルのイベントに存在します。request (url、メソッド、ヘッダー、本文)responseTimeMs 、 、 response (statusCode、ヘッダー、本文)、および truncated (ログイベントサイズ制限内に収まるようにリクエストまたはレスポンス本文が切り捨てられた場合、true) が含まれます。
input	オブジェクト	false	出力式の対象となる変数の前値。関数レベルのイベントにのみ存在します。
output	オブジェクト	false	関数によって生成される値。PRE_AD_REQUEST_FUNCTION_COMPLETED イベントにのみ存在します。

プロパティ	タイプ	必須	説明
status	string	false	フック実行の結果: SUCCESSまたは ERROR。PRE_ADS_REQUEST_H0 OK_SUMMARY イベントにのみ存在します。
originId	string	true	MediaTailor 設定からの設定名。これは、同じく設定の一部である動画コンテンツソースとは異なります。
prefetchScheduleName	string	false	この広告イベントに関連付けられたプリフェッチスケジュールの名前。
requestHeaders	requestheaders 型の配列	false	MediaTailor が ADS リクエストに含めたヘッダー。通常、ADS へのリクエストが失敗するとログに含まれるもので、トラブルシューティングに役立ちます。
requestId	string	true	MediaTailor リクエスト ID。同じリクエストの複数のログエントリに関連付けるために使用できます。

プロパティ	タイプ	必須	説明
sessionId	string	true	MediaTailor がプレイヤーセッションに割り当てた一意の数値識別子。プレイヤーがセッションに対して行うすべてのリクエストは、同じセッション ID を持ちます。例えば、e039fd39-09f0-46b2-aca9-9871cc116cde などです。
sessionType	string (有効な値: [DASH, HLS])	true	プレイヤーのストリームタイプ。
vastAd	vastAd 型のオブジェクト	false	VAST レスポンスから解析された 1 つの広告に関する情報。
vastResponse	vastResponse 型のオブジェクト	false	MediaTailor が ADS から受け取った VAST レスポンスに関する情報。
vodCreativeOffsets	vodCreativeOffsets 型のオブジェクト	false	VMAP レスポンスに基づいて、MediaTailor が avail を挿入するマニフェスト内の時間オフセットを示すマップ。

プロパティ	タイプ	必須	説明
<code>vodVastResponseTimeOffset</code>	数値	false	VOD 広告挿入の VMAP 固有の時間オフセット。

adContent

このセクションでは、ADS ログ adContent のプロパティについて説明します。

ADS ログ adContent プロパティ

プロパティ	タイプ	必須	説明
<code>adPlaylistUri</code>	adPlaylistUri 型のオブジェクト	false	バリエントのオリジンマニフェストからバリエントの広告マニフェストへのマッピング。DASH の場合、これには 1 つのエントリが含まれます。すべてのバリエントが 1 つの DASH マニフェストで表されるためです。

adPlaylistUri

このセクションでは、ADS ログ adPlaylistUri のプロパティについて説明します。

ADS ログ adPlaylistUri プロパティ

プロパティ	タイプ	必須	説明
<code><any string></code>	string	false	特定のバリエントの広告マニフェストの URL。

avail

このセクションでは、ADS ログ avail のプロパティについて説明します。

ADS ログ avail プロパティ

プロパティ	タイプ	必須	説明
availId	string	true	この avail の一意の識別子。HLS の場合、これは avail が開始される位置のメディアシーケンス番号です。DASH の場合、これは期間 ID です。
creativeAds	creativeAd 型の配列	true	MediaTailor が avail に挿入した広告。
fillRate	数値	true	広告が avail 継続時間を埋めるレート。0.0 (0% の場合) から 1.0 (100% の場合) です。
filledDuration	数値	true	Avail に挿入されたすべての広告の継続時間の合計。
numAds	数値	true	MediaTailor が avail に挿入した広告の数。
originAvailDuration	数値	true	オリジンからのコンテンツストリームで指定された avail の継続時間 (CUE_OUT または SCTE)。

プロパティ	タイプ	必須	説明
skippedAds	skippedAd 型の配列	false	MediaTailor が挿入しなかった広告 TRANSCODE_IN_PROGRESS やなど TRANSCODE_ERROR 。 スキップされた広告理由のリストについては、「」を参照してください 広告スキップのトラブルシューティング 。
slateAd	slateAd 型のオブジェクト	true	MediaTailor が avail の未入力セグメントを埋めるために使用するスレート広告に関する情報。

creativeAd

このセクションでは、ADS ログ creativeAd のプロパティについて説明します。

ADS ログ creativeAd プロパティ

プロパティ	タイプ	必須	説明
adContent	adContent 型のオブジェクト	true	挿入された広告のコンテンツに関する情報。
creativeUniqueId	string	true	トランスコーディングのキーとして使用される、広告の一意的識別子。使用可能

プロパティ	タイプ	必須	説明
			な場合、これは、VAST レスponse内のクリエイティブの ID フィールドです。それ以外の場合は、広告のメザニン URL です。
trackingEvents	trackingEvents 型のオブジェクト	true	広告のさまざまな追跡イベントの追跡ビーコン URL。キーはイベント名であり、値はビーコン URL のリストです。
transcodedAdDuration	数値	true	トランスコードされたアセットから計算された広告の継続時間。
uri	string	true	トランスコーダーへの入力であるメザニンバージョンの広告の URL。
vastDuration	数値	true	VAST レスponseから解析された広告の継続時間。

requestheaders

このセクションでは、ADS ログの requestheaders のプロパティについて説明します。

ADS ログの requestheaders のプロパティ

プロパティ	タイプ	必須	説明
name	string	true	ヘッダーの名前。
value	string	true	ヘッダーの値

skippedAd

このセクションでは、ADS ログ skippedAd のプロパティについて説明します。

ADS ログ skippedAd のプロパティ

プロパティ	タイプ	必須	説明
adMezzanineUrl	string	true	スキップされた広告のメザニン URL。
creativeUniqueId	string	true	トランスコーディングのキーとして使用される、広告の一意的識別子。使用可能な場合、これは、VAST レスponse内のクリエイティブの ID フィールドです。それ以外の場合は、広告のメザニン URL です。
skippedReason	string	true	広告が挿入されなかった理由を示すコード。例えば、TRANSCODE_IN_PROGRESS などです。

プロパティ	タイプ	必須	説明
			スキップされた広告理由のリストについては、「」を参照してください 広告スキップのトラブルシューティング 。
transcodeAdDuration	数値	false	トランスコードされたアセットから計算された広告の継続時間。
vastDuration	数値	true	VAST レスポンスから解析された広告の継続時間。

slateAd

このセクションでは、ADS ログ slateAd のプロパティについて説明します。

ADS ログ slateAd のプロパティ

プロパティ	タイプ	必須	説明
adContent	adContent 型のオブジェクト	true	挿入された広告のコンテンツに関する情報。
creativeUniqueId	string	true	トランスコーディングのキーとして使用される、広告の一意の識別子。使用可能な場合、これは、VAST レスポンス内のクリエイティブの ID フィールドです。それ以外の場合は、広

プロパティ	タイプ	必須	説明
			告のメザニン URL です。
transcode dAdDuration	数値	true	トランスコードされたアセットから計算された広告の継続時間。
uri	string	true	トランスコーダーへの入力であるメザニンバージョンの広告の URL。

trackingEvents

このセクションでは、ADS ログ trackingEvents のプロパティについて説明します。

ADS ログ trackingEvents のプロパティ

プロパティ	タイプ	必須	説明
<any string>	文字列型の配列	false	指定された追跡イベントのビーコン URL のリスト (インプレッション、完了など)

vastAd

このセクションでは、ADS ログ vastAd のプロパティについて説明します。

ADS ログ vastAd のプロパティ

プロパティ	タイプ	必須	説明
adSystem	string	true	VAST レスポンスの AdSystem タグの値。

プロパティ	タイプ	必須	説明
adTitle	string	true	VAST レスポンスで 広告に使用できるメ ディアファイル。
creativeAdId	string	true	VAST レスポンスの Creative タグの adId 属性の値。
creativeId	string	true	VAST レスポンスの Creative タグの id 属性の値。
duration	数値	true	VAST レスポンスの duration 要素の linear タグに基づ く、広告のおおよそ の継続時間。
trackingEvents	trackingEvents 型のオ ブジェクト	true	広告のさまざまな追 跡イベントの追跡ビ ーコン URL。キーは イベント名であり、 値はビーコン URL の リストです。
vastAdId	string	true	VAST レスポンスの Ad タグの id 属性の値
vastAdTagUri	string	false	広告の VMAP 固有の リダイレクト URI。
vastMediaFiles	vastMediaFile 型の配 列	true	VAST レスポンスの 広告に使用可能なメ ディアファイルのリ スト。

vastMediaFile

このセクションでは、ADS ログ vastMediaFile のプロパティについて説明します。

ADS ログ vastMediaFile のプロパティ

プロパティ	タイプ	必須	説明
apiFramework	string	true	メディアファイルの管理に必要な API フレームワーク。例えば、VPAID などです。
bitrate	数値	true	メディアファイルのビットレート。
delivery	string	true	メディアファイルに使用されるプロトコル。プログレッシブまたはストリーミングのいずれかに設定されます。
height	数値	true	メディアファイルのピクセルの高さ。
id	string	true	MediaFile タグの id 属性の値。
type	string	true	MediaFile タグの type 属性から取得されたメディアファイルの MIME タイプ。
uri	string	true	トランスコーダーへの入力であるメザンバージョンの広告の URL。

プロパティ	タイプ	必須	説明
width	数値	true	メディアファイルのピクセル幅。

vastResponse

このセクションでは、ADS ログ vastResponse のプロパティについて説明します。

ADS ログ vastResponse のプロパティ

プロパティ	タイプ	必須	説明
errors	文字列型の配列	true	VAST レスポンスの Error タグから解析されたエラー URL。
vastAds	vastAd 型の配列	true	VAST レスポンスから解析された広告。
version	string	true	レスポンスの VAST タグの version 属性から解析された VAST 仕様バージョン。

vodCreativeOffsets

このセクションでは、ADS ログ vodCreativeOffsets のプロパティについて説明します。

ADS ログ vodCreativeOffsets のプロパティ

プロパティ	タイプ	必須	説明
<any string>	vodCreativeOffset 型の配列	false	マニフェストの時間オフセットから、この時点で挿入する広告のリストへのマッピング。

vodCreativeOffset

このセクションでは、ADS ログ vodCreativeOffset のプロパティについて説明します。

ADS ログ vodCreativeOffset のプロパティ

プロパティ	タイプ	必須	説明
adContent	adContent 型のオブジェクト	true	挿入された広告のコンテンツに関する情報。
creativeUniqueId	string	true	トランスコーディングのキーとして使用される、広告の一意の識別子。使用可能な場合、これは、VAST レスポンス内のクリエイティブの ID フィールドです。それ以外の場合は、広告のメザニン URL です。
trackingEvents	trackingEvents 型のオブジェクト	true	広告のさまざまな追跡イベントの追跡ビーコン URL。キーはイベント名であり、値はビーコン URL のリストです。
transcodedAdDuration	数値	true	トランスコードされたアセットから計算された広告の継続時間。
uri	string	true	トランスコーダーへの入力であるメザニ

プロパティ	タイプ	必須	説明
			ンバージョンの広告の URL。
vastDuration	数値	true	VAST レスポンスから解析された広告の継続時間。

ADS ログの JSON スキーマ

ADS ログの JSON AWS Elemental MediaTailor スキーマを以下に示します。

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "http://amazon.com/elemental/midas/mms/adsLogSchema.json",
  "type": "object",
  "title": "AWS Elemental MediaTailor ADS Log JSON Schema",
  "required": [
    "eventType",
    "eventTimestamp",
    "requestId",
    "sessionType",
    "eventDescription",
    "awsAccountId",
    "customerId",
    "originId",
    "sessionId"
  ],
  "additionalProperties": false,
  "properties": {
    "eventType": {
      "$id": "#/properties/eventType",
      "type": "string",
      "description": "The code for the event that triggered this log message. Example: <code>VAST_RESPONSE</code>.",
      "examples": [
        "FILLED_AVAIL"
      ]
    },
    "eventTimestamp": {
      "$id": "#/properties/eventTimestamp",
```

```

    "type": "string",
    "description": "The date and time of the event.",
    "examples": [
        "1970-01-01T00:00:00Z"
    ],
    "format": "date-time"
},
"requestId": {
    "$id": "#/properties/requestId",
    "type": "string",
    "description": "The MediaTailor request ID, which you can use to correlate
multiple log entries for the same request.",
    "examples": [
        "c7c7ae8c-a61e-44e0-8efd-7723995337a1"
    ],
    "pattern": "^(.*)$"
},
"sessionType": {
    "$id": "#/properties/sessionType",
    "type": "string",
    "enum": [
        "HLS",
        "DASH"
    ],
    "description": "The player's stream type."
},
"eventDescription": {
    "$id": "#/properties/eventDescription",
    "type": "string",
    "description": "A short description of the event that triggered this log message,
provided by the MediaTailor service. By default, this is empty. Example: <code>Got
VAST response</code>.",
    "default": "",
    "examples": [
        "Got VAST response"
    ],
    "pattern": "^(.*)$"
},
"awsAccountId": {
    "$id": "#/properties/awsAccountId",
    "type": "string",
    "description": "The AWS account ID for the MediaTailor configuration that was
used for the session."
},

```



```

"customerId": {
  "$id": "#/properties/customerId",
  "type": "string",
  "description": "The hashed version of the AWS account ID, which you can use to
correlate multiple log entries.",
  "pattern": "^(.*)$"
},
"originId": {
  "$id": "#/properties/originId",
  "type": "string",
  "description": "The configuration name from the MediaTailor configuration. This
is different from the video content source, which is also part of the configuration.",
  "examples": [
    "external-canary-dash-serverside-reporting-onebox"
  ],
  "pattern": "^(.*)$"
},
"sessionId": {
  "$id": "#/properties/sessionId",
  "type": "string",
  "description": "The unique numeric identifier that MediaTailor assigned to the
player session. All requests that a player makes for a session have the same session
ID. Example: <code>e039fd39-09f0-46b2-aca9-9871cc116cde</code>.",
  "examples": [
    "120b9873-c007-40c8-b3db-0f1bd194970b"
  ],
  "pattern": "^(.*)$"
},
"avail": {
  "$id": "#/properties/avail",
  "type": "object",
  "title": "avail",
  "description": "Information about an avail that MediaTailor fills with ads.
Currently, for the <code>FILLED_AVAIL</code> event type, this is the plan created by
MediaTailor when it first encounters the avail. How the avail is eventually filled may
vary from this plan, depending on how the content plays out. ",
  "required": [
    "creativeAds",
    "originAvailDuration",
    "filledDuration",
    "fillRate",
    "driftMillisecondsAtAvailStart",
    "numAds",
    "slateAd",

```

```

    "availId"
  ],
  "additionalProperties": false,
  "properties": {
    "originAvailDuration": {
      "$id": "#/properties/avail/originAvailDuration",
      "type": "number",
      "description": "The duration of the avail as specified in the content stream
from the origin (<code>CUE_OUT</code> or <code>SCTE</code>).",
    },
    "filledDuration": {
      "$id": "#/properties/avail/filledDuration",
      "type": "number",
      "description": "The sum of the durations of all the ads inserted into the
avail."
    },
    "fillRate": {
      "$id": "#/properties/avail/fillRate",
      "type": "number",
      "description": "The rate at which the ads fill the avail duration, from 0.0
(for 0%) to 1.0 (for 100%).",
    },
    "driftMillisecondsAtAvailStart": {
      "$id": "#/properties/avail/driftMillisecondsAtAvailStart",
      "type": "number",
      "description": "The cumulative drift at the beginning of this avail. A
positive value implies that we are moving away from the live edge, a negative value
implies that we are moving towards the live edge."
    },
    "creativeAds": {
      "$id": "#/properties/avail/creativeAds",
      "type": "array",
      "description": "The ads that MediaTailor inserted into the avail.",
      "items": {
        "type": "object",
        "title": "creativeAd",
        "description": "Information about a single inserted ad.",
        "required": [
          "uri",
          "creativeUniqueId",
          "adSystem",
          "adContent",
          "trackingEvents",
          "vastDuration",

```

```

        "transcodedAdDuration"
    ],
    "additionalProperties": false,
    "properties": {
        "uri": { "$ref": "#/definitions/adMezzanineUri" },
        "creativeUniqueId": { "$ref": "#/definitions/creativeUniqueId" },
        "adSystem": { "$ref": "#/definitions/adSystem" },
        "adContent": { "$ref": "#/definitions/adContent" },
        "trackingEvents": { "$ref": "#/definitions/trackingEvents" },
        "vastDuration": { "$ref": "#/definitions/vastDuration" },
        "transcodedAdDuration": { "$ref": "#/definitions/transcodedAdDuration" }
    }
},
"numAds": {
    "$id": "#/properties/avail/numAds",
    "type": "number",
    "description": "The number of ads that MediaTailor inserted into the avail."
},
"slateAd": {
    "$id": "#/properties/avail/slateAd",
    "type": ["object", "null"],
    "title": "slateAd",
    "description": "Information about the slate ad, which MediaTailor uses to
fill any unfilled segments in the avail.",
    "additionalProperties": false,
    "required": [
        "uri",
        "creativeUniqueId",
        "adContent",
        "transcodedAdDuration"
    ],
    "properties": {
        "uri": { "$ref": "#/definitions/adMezzanineUri" },
        "creativeUniqueId": { "$ref": "#/definitions/creativeUniqueId" },
        "adContent": { "$ref": "#/definitions/adContent" },
        "transcodedAdDuration": { "$ref": "#/definitions/transcodedAdDuration" }
    }
},
"availId": {
    "$id": "#/properties/avail/availId",
    "type": "string",
    "description": "The unique identifier for this avail. For HLS, this is the
media sequence number where the avail begins. For DASH, this is the period ID."

```

```

    },
    "skippedAds": {
      "$id": "#/properties/avail/skippedAds",
      "type": "array",
      "description": "The ads that MediaTailor didn't insert, for reasons like
<code>TRANSCODE_IN_PROGRESS</code> and <code>TRANSCODE_ERROR</code>.",
      "items": {
        "type": "object",
        "title": "skippedAd",
        "description": "Information about a single skipped ad.",
        "required": [
          "creativeUniqueId",
          "adMezzanineUrl",
          "skippedReason",
          "vastDuration"
        ],
        "additionalProperties": false,
        "properties": {
          "creativeUniqueId": { "$ref": "#/definitions/creativeUniqueId" },
          "adMezzanineUrl": {
            "type": "string",
            "description": "The mezzanine URL of the skipped ad."
          },
          "skippedReason": {
            "type": "string",
            "description": "The code that indicates why the ad wasn't inserted.
Example: <code>TRANSCODE_IN_PROGRESS</code>."
          },
          "vastDuration": { "$ref": "#/definitions/vastDuration" },
          "transcodedAdDuration": { "$ref": "#/definitions/transcodedAdDuration" },
          "targetVariant": {
            "type": "object",
            "title": "targetVariant",
            "description": "The target variant of the source content. This key is
present when an ad wasn't inserted due to the source content containing a variant that
could not match to any variants present in this ad.",
            "required": [
              "mediaProtocol",
              "mediaType",
              "bitrate",
              "mediaResolution",
              "codecs"
            ],
            "additionalProperties": false,

```

```
"properties": {
  "mediaProtocol": {
    "type": "string",
    "description": "The media protocol of this variant, such as HLS.",
    "enum": [
      "HLS",
      "DASH"
    ]
  },
  "mediaType": {
    "type": "array",
    "description": "The media type of this variant, such as VIDEO.",
    "items": {
      "type": "string",
      "enum": [
        "VIDEO",
        "AUDIO",
        "SUBTITLES",
        "TRICK_PLAY"
      ]
    },
    "description": "Media type, such as VIDEO."
  }
},
"bitrate": {
  "$ref": "#/definitions/bitrate"
},
"mediaResolution": {
  "type": "object",
  "title": "mediaResolution",
  "description": "The media resolution of this variant.",
  "required": [
    "width",
    "height"
  ],
  "additionalProperties": false,
  "properties": {
    "width": {
      "$ref": "#/definitions/width"
    },
    "height": {
      "$ref": "#/definitions/height"
    }
  }
},
},
```

```

        "codecs": {
            "type": "array",
            "description": "The codecs of this variant.",
            "items": {
                "type": "string",
                "description": "Codec, such as avc1."
            }
        }
    },
    "adBreakTrackingEvents": {
        "$id": "#properties/avail/adBreakTrackingEvents",
        "type": "object",
        "title": "adBreakTrackingEvents",
        "description": "VMAP/ad break tracking events and corresponding URL",
        "properties": {
            "items": {
                "$ref": "#/definitions/adBreakTrackingEvents"
            }
        }
    },
    "vastResponse": {
        "$id": "#/properties/vastResponse",
        "type": "object",
        "title": "vastResponse",
        "description": "Information about the VAST response that MediaTailor received from the ADS.",
        "required": [
            "version",
            "vastAds",
            "errors",
            "nonLinearAdsList"
        ],
        "additionalProperties": false,
        "properties": {
            "version": {
                "$id": "#/properties/vastResponse/version",
                "type": "string",

```

```

      "description": "The VAST specification version, parsed from the
<code>version</code> attribute of the <code>VAST</code> tag in the response.",
      "examples": [
        "3.0"
      ],
      "pattern": "^(.*)$"
    },
    "vastAds": {
      "$id": "#/properties/vastResponse/vastAds",
      "type": "array",
      "description": "The ads parsed from the VAST response.",
      "items": {
        "$ref": "#/definitions/vastAd"
      }
    },
    "errors": {
      "$id": "#/properties/vastResponse/errors",
      "type": "array",
      "description": "The error URLs parsed from the <code>Error</code> tags in the
VAST response.",
      "items": {
        "type": "string",
        "description": "A single error URL."
      }
    },
    "nonLinearAdsList": {
      "$id": "#/properties/vastResponse/nonLinearAds",
      "type": "array",
      "description": "A list of NonLinearAds as they are read from the VAST
response.",
      "items": {
        "$ref": "#/definitions/nonLinearAds"
      }
    }
  },
  "vastAd": {
    "$ref": "#/definitions/vastAd"
  },
  "vodVastResponseTimeOffset": {
    "$id": "#/properties/vodVastResponseTimeOffset",
    "type": "number",

```

```

    "description": "The VMAP specific time offset for VOD ad insertion.",
    "examples": [
      5.0
    ]
  },

  "vodCreativeOffsets": {
    "$id": "#/properties/vodCreativeOffsets",
    "type": "object",
    "title": "vodCreativeOffsets",
    "description": "A map that indicates the time offsets in the manifest where
MediaTailor will insert avails, based on the VMAP response.",
    "additionalProperties": {
      "type": "array",
      "$id": "#/properties/vodCreativeOffsets/entry",
      "description": "A mapping from a time offset in the manifest to a list of ads
to insert at this time.",
      "items": {
        "type": "object",
        "$id": "#/properties/vodCreativeOffsets/entry/items",
        "title": "vodCreativeOffset",
        "description": "The list of ads to insert at the specified time offset.",
        "additionalProperties": false,
        "required": [
          "uri",
          "creativeUniqueId",
          "vastDuration",
          "transcodedAdDuration",
          "adContent",
          "trackingEvents"
        ],
        "properties": {
          "uri": { "$ref": "#/definitions/adMezzanineUri" },
          "creativeUniqueId": { "$ref": "#/definitions/creativeUniqueId" },
          "vastDuration": { "$ref": "#/definitions/vastDuration" },
          "transcodedAdDuration": { "$ref": "#/definitions/transcodedAdDuration" },
          "adContent": { "$ref": "#/definitions/adContent" },
          "trackingEvents": { "$ref": "#/definitions/trackingEvents" }
        }
      }
    ]
  },

  "adsRequestUrl": {

```



```

    "$id": "#/properties/adsRequestUrl",
    "type": "string",
    "description": "The full URL of the ADS request made by MediaTailor."
  },
  "adMarkers": {
    "$id": "#/properties/adMarkers",
    "type": "string",
    "description": "Found Ad Marker in the Manifest."
  },
  "segmentationUpid": {
    "$id": "#/properties/segmentationUpid",
    "type": "string",
    "description": "Value of segmentation upid parsed from ad markers in manifest."
  },
  "segmentationTypeId": {
    "$id": "#/properties/segmentationTypeId",
    "type": "integer",
    "description": "Value of segmentation typeId parsed from ad markers in manifest."
  },
  "requestHeaders": {
    "$id": "#/properties/requestHeaders",
    "type": "array",
    "description": "The headers that MediaTailor included with the ADS request.
Typically, the logs include these when a request to the ADS fails, to help with
troubleshooting.",
    "items": {
      "type": "object",
      "title": "requestheaders",
      "description": "The name and value for a single header included in the ADS
request.",
      "required": [
        "name",
        "value"
      ],
      "additionalProperties": false,
      "properties": {
        "name": {
          "type": "string",
          "description": "The name of the header."
        },
        "value": {
          "type": "string",
          "description": "The value of the header."
        }
      }
    }
  }
}

```

```

    }
  }
},

"originalTargetUrl": {
  "$id": "#/properties/originalTargetUrl",
  "type": "string",
  "description": "The old URL to which MediaTailor was going to make a request."
},
"prefetchScheduleName": {
  "$id": "#/properties/prefetchScheduleName",
  "type": "string",
  "description": "The name of the prefetch schedule associated with this ad
event.",
  "examples": [
    "PrefetchScheduleName"
  ],
  "pattern": "^(.*)$"
},
"updatedTargetUrl": {
  "$id": "#/properties/updatedTargetUrl",
  "type": "string",
  "description": "The new URL to which MediaTailor is making a request."
},

"rawAdsResponse": {
  "$id": "#/properties/rawAdsResponse",
  "type": "string",
  "description": "Paginated ADS response as it's exactly returned to MediaTailor."
},
"rawAdsResponseIndex": {
  "$id": "#/properties/rawAdsResponseIndex",
  "type": "integer",
  "description": "Integer value denoting this rawAdsResponse's index into the
full ADS response. This value is used to order the paginated messages for this ADS
response."
},

"eventId": {
  "$id": "#/properties/eventId",
  "type": "string",
  "description": "Unique identifier per hook invocation. Use this field to
correlate hook-level and function-level events for the same execution.",
  "examples": [

```

```

    "5dc6f040-0f72-4e8c-a64e-25eeef62708c"
  ],
},
"functionId": {
  "$id": "#/properties/functionId",
  "type": "string",
  "description": "The identifier of the function. On hook events, this is the
function configured on the hook. On function events, this is the individual function
within the executor.",
  "examples": [
    "my_first_function"
  ]
},
"functionType": {
  "$id": "#/properties/functionType",
  "type": "string",
  "enum": [
    "CUSTOM_OUTPUT",
    "HTTP_REQUEST",
    "SEQUENTIAL_EXECUTOR",
    "CONCURRENT_EXECUTOR"
  ],
  "description": "The type of function."
},
"executionTimeMs": {
  "$id": "#/properties/executionTimeMs",
  "type": "integer",
  "description": "Execution time in milliseconds."
},
"status": {
  "$id": "#/properties/status",
  "type": "string",
  "enum": [
    "SUCCESS",
    "ERROR"
  ],
  "description": "Hook execution outcome. Present on HOOK_SUMMARY events only."
},
"errorType": {
  "$id": "#/properties/errorType",
  "type": "string",
  "enum": [
    "INTERNAL_ERROR",
    "SYNTAX_ERROR",

```

```

        "RESOURCE_LIMIT_ERROR",
        "RESTRICTION_ERROR",
        "TIMEOUT_ERROR",
        "VALIDATION_ERROR"
    ],
    "description": "Structured error classification. Present on error events only."
},
"cause": {
    "$id": "#/properties/cause",
    "type": "string",
    "description": "Error detail string describing the specific failure."
},
"input": {
    "$id": "#/properties/input",
    "type": "object",
    "description": "Before-values of variables targeted by output expressions."
},
"output": {
    "$id": "#/properties/output",
    "type": "object",
    "description": "Values produced by the function. Present on FUNCTION_COMPLETED
events only."
},
"http": {
    "$id": "#/properties/http",
    "type": "object",
    "description": "HTTP request and response details for HTTP_REQUEST functions.",
    "properties": {
        "request": {
            "type": "object",
            "description": "The HTTP request details.",
            "properties": {
                "url": { "type": "string", "description": "The request URL." },
                "method": { "type": "string", "description": "The HTTP method." },
                "headers": {
                    "type": "array",
                    "description": "The request headers.",
                    "items": {
                        "type": "object",
                        "properties": {
                            "name": { "type": "string" },
                            "value": { "type": "string" }
                        }
                    }
                }
            }
        }
    }
}

```

```

    },
    "body": { "type": ["string", "null"], "description": "The request body." }
  },
  "responseTimeMs": { "type": ["integer", "null"], "description": "HTTP response
time in milliseconds." },
  "response": {
    "type": ["object", "null"],
    "description": "The HTTP response details. Null when the server was never
reached.",
    "properties": {
      "statusCode": { "type": ["integer", "null"], "description": "The HTTP
status code. Null when the server was never reached." },
      "headers": {
        "type": ["array", "null"],
        "description": "The response headers.",
        "items": {
          "type": "object",
          "properties": {
            "name": { "type": "string" },
            "value": { "type": "string" }
          }
        }
      }
    },
    "body": { "type": ["string", "null"], "description": "The response body." }
  },
  "truncated": { "type": "boolean", "description": "True when request or response
body was truncated to fit within log event size limits." }
}
},

```

"__COMMENT_oneOf": "The oneOf section defines subtypes for our events. Subtypes can have different rules, including which fields are required. For more information, see <https://json-schema.org/understanding-json-schema/reference/combining.html#oneof> ",

```

"oneOf": [
  { "$ref": "#/definitions/eventAdMarkersFound" },
  { "$ref": "#/definitions/eventNonAdMarkerFound" },
  { "$ref": "#/definitions/eventMakingAdsRequest" },
  { "$ref": "#/definitions/eventModifiedTargetUrl" },
  { "$ref": "#/definitions/eventRawAdsResponse" },
  { "$ref": "#/definitions/eventVastResponse" },

```

```

{ "$ref": "#/definitions/eventFilledAvail" },
{ "$ref": "#/definitions/eventFilledOverlayAvail" },
{ "$ref": "#/definitions/eventErrorFiringBeaconFailed" },
{ "$ref": "#/definitions/eventWarningNoAdvertisements" },
{ "$ref": "#/definitions/eventUnknownHost" },
{ "$ref": "#/definitions/eventErrorAdsTimeout" },
{ "$ref": "#/definitions/eventErrorVastMissingOverlays" },
{ "$ref": "#/definitions/eventPlannedAvail" },
{ "$ref": "#/definitions/eventEmptyVastResponse" },
{ "$ref": "#/definitions/eventEmptyVmapResponse" },
{ "$ref": "#/definitions/eventErrorUnknown" },
{ "$ref": "#/definitions/eventVastRedirect" },
{ "$ref": "#/definitions/eventRedirectedVastResponse" },
{ "$ref": "#/definitions/eventErrorAdsMissingImpression" },
{ "$ref": "#/definitions/eventErrorAdsResponseParse" },
{ "$ref": "#/definitions/eventErrorAdsInvalidResponse" },
{ "$ref": "#/definitions/eventErrorDisallowedHost" },
{ "$ref": "#/definitions/eventPersonalizationDisabled" },
{ "$ref": "#/definitions/eventWarningDynamicVariableSubFailed" },
{ "$ref": "#/definitions/eventVodTimeBasedAvailPlanVastResponseForOffset" },
{ "$ref": "#/definitions/eventVodTimeBasedAvailPlanSuccess" },
{ "$ref": "#/definitions/eventPreAdsRequestHookSummary" },
{ "$ref": "#/definitions/eventPreAdsRequestHookError" },
{ "$ref": "#/definitions/eventPreAdsRequestFunctionCompleted" },
{ "$ref": "#/definitions/eventPreAdsRequestFunctionError" }
],

"definitions": {
  "eventAdMarkersFound": {
    "$id": "#/definitions/eventAdMarkersFound",
    "required": [
      "eventType",
      "adMarkers"
    ],
    "properties": {
      "eventType": {
        "type": "string",
        "const": "AD_MARKER_FOUND"
      }
    }
  },
  "eventNonAdMarkerFound": {
    "$id": "#/definitions/eventNonAdMarkerFound",
    "required": [

```

```
    "eventType",
    "adMarkers"
  ],
  "properties": {
    "eventType": {
      "type": "string",
      "const": "NON_AD_MARKER_FOUND"
    }
  }
},
"eventMakingAdsRequest": {
  "$id": "#/definitions/eventMakingAdsRequest",
  "required": [
    "eventType",
    "adsRequestUrl"
  ],
  "properties": {
    "eventType": {
      "type": "string",
      "const": "MAKING_ADS_REQUEST"
    }
  }
},

"eventModifiedTargetUrl": {
  "$id": "#/definitions/eventModifiedTargetUrl",
  "required": [
    "eventType",
    "originalTargetUrl",
    "updatedTargetUrl"
  ],
  "properties": {
    "eventType": {
      "type": "string",
      "const": "MODIFIED_TARGET_URL"
    }
  }
},

"eventRawAdsResponse": {
  "$id": "#/definitions/eventRawAdsResponse",
  "required": [
    "eventType",
    "rawAdsResponse",
```

```

    "rawAdsResponseIndex"
  ],
  "properties": {
    "eventType": {
      "type": "string",
      "const": "RAW_ADS_RESPONSE"
    }
  }
},

```

"eventVastResponse": {
 "\$id": "#/definitions/eventVastResponse",
 "_comment": "NOTE: the vastResponse property should ideally be marked as a required field for this event, but unfortunately, in the case of an empty vast response, we currently emit an EMPTY_VAST_RESPONSE followed by a VAST_RESPONSE, and the vastResponse property is not present in the latter. We need to fix this so that we don't emit both of those events in the empty response case, and update this schema to flag vastResponse as required for VAST_RESPONSE.",

```

  "required": [
    "eventType"
  ],
  "properties": {
    "eventType": {
      "type": "string",
      "const": "VAST_RESPONSE"
    }
  }
},

```

```

"eventFilledAvail": {
  "$id": "#/definitions/eventFilledAvail",
  "required": [
    "eventType",
    "avail"
  ],
  "properties": {
    "eventType": {
      "type": "string",
      "const": "FILLED_AVAIL"
    }
  }
},

```

```

"eventFilledOverlayAvail": {

```



```
    "$id": "#/definitions/eventFilledOverlayAvail",
    "required": [
      "eventType",
      "avail"
    ],
    "properties": {
      "eventType": {
        "type": "string",
        "const": "FILLED_OVERLAY_AVAIL"
      }
    }
  },

  "eventErrorVastMissingOverlays": {
    "$id": "#/definitions/eventErrorVastMissingOverlays",
    "required": [
      "eventType",
      "adsRequestUrl",
      "requestHeaders"
    ],
    "properties": {
      "eventType": {
        "type": "string",
        "const": "ERROR_VAST_MISSING_OVERLAYS"
      }
    }
  },

  "eventErrorFiringBeaconFailed": {
    "$id": "#/definitions/eventErrorFiringBeaconFailed",
    "required": [
      "eventType",
      "error",
      "beaconInfo"
    ],
    "properties": {
      "eventType": {
        "type": "string",
        "const": "ERROR_FIRING_BEACON_FAILED"
      }
    }
  },

  "eventWarningNoAdvertisements": {
```

```
    "$id": "#/definitions/eventWarningNoAdvertisements",
    "required": [
      "eventType"
    ],
    "_comment": "We should really have a more descriptive error field for these
events",
    "properties": {
      "eventType": {
        "type": "string",
        "const": "WARNING_NO_ADVERTISEMENTS"
      }
    }
  },
  "eventUnknownHost": {
    "$id": "#/definitions/eventUnknownHost",
    "required": [
      "eventType",
      "requestHeaders"
    ],
    "properties": {
      "eventType": {
        "type": "string",
        "const": "ERROR_UNKNOWN_HOST"
      }
    }
  },
  "eventErrorAdsTimeout": {
    "$id": "#/definitions/eventErrorAdsTimeout",
    "required": [
      "eventType",
      "adsRequestUrl",
      "requestHeaders"
    ],
    "properties": {
      "eventType": {
        "type": "string",
        "const": "ERROR_ADS_TIMEOUT"
      }
    }
  },
  "eventPlannedAvail": {
```

```
    "$id": "#/definitions/eventPlannedAvail",
    "required": [
      "eventType"
    ],
    "_comment": "TODO: Flesh this out as we implement it",
    "properties": {
      "eventType": {
        "type": "string",
        "const": "PLANNED_AVAIL"
      }
    }
  },

  "eventEmptyVastResponse": {
    "$id": "#/definitions/eventEmptyVastResponse",
    "required": [
      "eventType"
    ],
    "properties": {
      "eventType": {
        "type": "string",
        "const": "EMPTY_VAST_RESPONSE"
      }
    }
  },

  "eventEmptyVmapResponse": {
    "$id": "#/definitions/eventEmptyVmapResponse",
    "required": [
      "eventType"
    ],
    "properties": {
      "eventType": {
        "type": "string",
        "const": "EMPTY_VMAP_RESPONSE"
      }
    }
  },

  "eventErrorUnknown": {
    "$id": "#/definitions/eventErrorUnknown",
    "required": [
      "eventType"
    ],
    "properties": {
      "eventType": {
        "type": "string",
        "const": "UNKNOWN_ERROR"
      }
    }
  }
}
```

```

    "_comment": "TODO: we should have a field for the exception message or
something",
    "properties": {
      "eventType": {
        "type": "string",
        "const": "ERROR_UNKNOWN"
      }
    }
  },

  "eventVastRedirect": {
    "$id": "#/definitions/eventVastRedirect",
    "required": [
      "eventType"
    ],
    "properties": {
      "eventType": {
        "type": "string",
        "const": "VAST_REDIRECT"
      }
    }
  },

  "eventRedirectedVastResponse": {
    "$id": "#/definitions/eventRedirectedVastResponse",
    "required": [
      "eventType"
    ],
    "properties": {
      "eventType": {
        "type": "string",
        "const": "REDIRECTED_VAST_RESPONSE"
      }
    }
  },
  "_comment": "NOTE that the property vastResponse is not required because empty
vast responses do not contain a vastResponse."
},

  "eventErrorAdsResponseParse": {
    "$id": "#/definitions/eventErrorAdsResponseParse",
    "required": [
      "eventType"
    ],
    "_comment": "We should have a field with an error message here",

```

```
    "properties": {
      "eventType": {
        "type": "string",
        "const": "ERROR_ADS_RESPONSE_PARSE"
      }
    }
  },

  "eventErrorAdsInvalidResponse": {
    "$id": "#/definitions/eventErrorAdsInvalidResponse",
    "required": [
      "eventType",
      "additionalInfo"
    ],
    "properties": {
      "eventType": {
        "type": "string",
        "const": "ERROR_ADS_INVALID_RESPONSE"
      }
    }
  },

  "eventErrorAdsMissingImpression": {
    "$id": "#/definitions/eventErrorAdsMissingImpression",
    "required": [
      "eventType"
    ],
    "properties": {
      "eventType": {
        "type": "string",
        "const": "ERROR_VAST_MISSING_IMPRESSION"
      }
    }
  },

  "eventErrorDisallowedHost": {
    "$id": "#/definitions/eventErrorDisallowedHost",
    "required": [
      "eventType"
    ],
    "properties": {
      "eventType": {
        "type": "string",
        "const": "ERROR_DISALLOWED_HOST"
      }
    }
  }
}
```

```
    }
  }
},

"eventPersonalizationDisabled": {
  "$id": "#/definitions/eventPersonalizationDisabled",
  "required": [
    "eventType"
  ],
  "properties": {
    "eventType": {
      "type": "string",
      "const": "ERROR_PERSONALIZATION_DISABLED"
    }
  }
},

"eventWarningDynamicVariableSubFailed": {
  "$id": "#/definitions/eventWarningDynamicVariableSubFailed",
  "required": [
    "eventType",
    "adsRequestUrl"
  ],
  "properties": {
    "eventType": {
      "type": "string",
      "const": "WARNING_URL_VARIABLE_SUBSTITUTION_FAILED"
    }
  }
},

"eventVodTimeBasedAvailPlanVastResponseForOffset": {
  "$id": "#/definitions/eventVodTimeBasedAvailPlanVastResponseForOffset",
  "required": [
    "eventType",
    "vastResponse"
  ],
  "properties": {
    "eventType": {
      "type": "string",
      "const": "VOD_TIME_BASED_AVAIL_PLAN_VAST_RESPONSE_FOR_OFFSET"
    }
  }
},
```

```

"eventVodTimeBasedAvailPlanSuccess": {
  "$id": "#/definitions/eventVodTimeBasedAvailPlanSuccess",
  "required": [
    "eventType",
    "vodCreativeOffsets"
  ],
  "properties": {
    "eventType": {
      "type": "string",
      "const": "VOD_TIME_BASED_AVAIL_PLAN_SUCCESS"
    }
  }
},

"creativeUniqueId": {
  "type": "string",
  "description": "The unique identifier for the ad, used as a key for transcoding.
This is the ID field for the creative in the VAST response, if available. Otherwise,
it's the mezzanine URL of the ad. "
},

"adSystem": {
  "type": "string",
  "description": "The value of the <code>AdSystem</code> tag in the VAST response.
"
},

"vastDuration": {
  "type": "number",
  "description": "The duration of the ad, as parsed from the VAST response."
},

"transcodedAdDuration": {
  "type": "number",
  "description": "The duration of the ad, calculated from the transcoded asset."
},

"adContent": {
  "$id": "#/properties/adContent",
  "type": ["object", "null"],
  "title": "adContent",
  "description": "Information about the content of the inserted ad.",
  "additionalProperties": false,

```

```
    "properties": {
      "adPlaylistUri": {
        "$id": "#/properties/adContent/adPlaylistUri",
        "type": "object",
        "title": "adPlaylistUri",
        "description": "The mapping from the origin manifest for a variant to the ad manifest for the variant. For DASH, this contains a single entry, because all variants are represented in a single DASH manifest. ",
        "additionalProperties": {
          "$id": "#/properties/adContent/adPlaylistUri/adPlaylistUri",
          "type": "string",
          "description": "The URL of the ad manifest for the specific variant."
        }
      }
    }
  },

  "adMezzanineUri": {
    "type": "string",
    "description": "The URL of the mezzanine version of the ad, which is the input to the transcoder."
  },

  "bitrate": {
    "type": "integer",
    "examples": [
      533
    ],
    "description": "The bitrate."
  },

  "width": {
    "type": "integer",
    "examples": [
      1280
    ],
    "description": "Width in pixels."
  },

  "height": {
    "type": "integer",
    "examples": [
      720
    ],
    "description": "Height in pixels."
  },
}
```



```

"trackingEvents": {
  "type": "object",
  "title": "trackingEvents",
  "description": "The tracking beacon URLs for the various tracking events for the
ad. The keys are the event names, and the values are a list of beacon URLs.",

  "additionalProperties": {
    "type": "array",
    "description": "The list of beacon URLs for the specified tracking event
(impression, complete, and so on)",
    "items": {
      "type": "string",
      "description": "The beacon URLs for this tracking event."
    }
  }
},

"nonLinearAds": {
  "$id": "#/properties/nonLinearAds",
  "type": "object",
  "title": "nonLinearAds",
  "description": "A NonLinearAds as it appears in the VAST response.",
  "required": [
    "nonLinearAdList",
    "nonLinearTrackingEvents"
  ],
  "properties": {
    "nonLinearAdList": {
      "type": "array",
      "description": "List of non linear ads as they exist within one
NonLinearAds.",
      "items": {
        "type": "object",
        "title": "nonLinearAdList",
        "description": "List of NonLinearAd as they are parsed from its parent
NonLinearAds.",
        "properties": {
          "nonLinearAdId": {
            "type": "string",
            "description": "Ad ID of this non linear ad."
          },
          "nonLinearAdSystem": {
            "type": "string",

```

```
    "description": "Ad system of this non linear ad's parent Inline ad."
  },
  "nonLinearAdTitle": {
    "type": "string",
    "description": "Ad title of this non linear ad's parent Inline ad."
  },
  "nonLinearCreativeId": {
    "type": "string",
    "description": "Creative ID of this non linear ad's parent Creative
ad."
  },
  "nonLinearCreativeAdId": {
    "type": "string",
    "description": "Creative ad ID of this non linear ad."
  },
  "nonLinearCreativeSequence": {
    "type": "string",
    "description": "Creative sequence of this non linear ad."
  },
  "nonLinearWidth": {
    "type": "string",
    "description": "Width of this non linear ad."
  },
  "nonLinearHeight": {
    "type": "string",
    "description": "Height of this non linear ad."
  },
  "nonLinearExpandedWidth": {
    "type": "string",
    "description": "Expanded width of this non linear ad."
  },
  "nonLinearExpandedHeight": {
    "type": "string",
    "description": "Expanded height of this non linear ad."
  },
  "nonLinearScalable": {
    "type": "boolean",
    "description": "Boolean denoting if this non linear ad is scalable."
  },
  "nonLinearMaintainAspectRatio": {
    "type": "boolean",
    "description": "Boolean denoting if aspect ratio should be maintained
for this non linear ad."
  },
  },
```

```

    "nonLinearMinSuggestedDuration": {
      "type": "string",
      "description": "Min suggested duration for this non linear ad."
    },
    "nonLinearApiFramework": {
      "type": "string",
      "description": "API framework for this non linear ad's parent Inline
ad."
    },
    "nonLinearStaticResource": {
      "type": "string",
      "description": "Static resource for this non linear ad."
    },
    "nonLinearStaticResourceCreativeType": {
      "type": "string",
      "description": "Static Resource creative type for this non linear ad."
    },
    "nonLinearIframeResource": {
      "type": "string",
      "description": "I-Frame resource for this non linear ad."
    },
    "nonLinearHtmlResource": {
      "type": "string",
      "description": "HTML resource for this non linear ad."
    },
    "nonLinearAdParameters": {
      "type": "string",
      "description": "Ad parameters for this non linear ad."
    },
    "nonLinearClickThrough": {
      "type": "string",
      "description": "Click Through data for this non linear ad."
    },
    "nonLinearClickTracking": {
      "type": "string",
      "description": "Click Tracking data for this non linear ad."
    },
    "nonLinearClickTrackingId": {
      "type": "string",
      "description": "Click Tracking ID for this non linear ad."
    }
  }
}
},

```

```

    "nonLinearTrackingEvents": { "$ref": "#/definitions/trackingEvents" },
    "extensions": {
      "$id": "#/properties/nonLinearAds/extensions",
      "type": "array",
      "description": "Extensions that exist for this NonLinearAds.",
      "items": {
        "$id": "#/properties/nonLinearAds/extensions/items",
        "type": "object",
        "title": "Extensions",
        "description": "Extensions found in non linear ads",
        "additionalProperties": false,
        "properties": {
          "extensionType": {
            "$id": "#/properties/nonLinearAds/extensions/extensionType",
            "type": "string",
            "description": "The value of the extension type attribute of the
<code>Extensions</code> tag.",
            "examples": [
              "FreeWheel"
            ]
          },
          "extensionContent": {
            "$id": "#/properties/nonLinearAds/extensions/extensionContent",
            "type": "string",
            "description": "The extension content attribute of the
<code>Extensions</code> tag.",
            "examples": [
              "progressive"
            ]
          }
        }
      }
    },
    "adBreakTrackingEvents": {
      "$id": "#/properties/adBreakTrackingEvents",
      "type": "object",
      "title": "adBreakTrackingEvents",
      "description": "These are all VMAP ad break tracking events.",
      "additionalProperties": {
        "type": "array",
        "description": "VMAP/ad break tracking events and corresponding URL",
        "items": {

```

```

        "type": "string",
        "description": "The beacon URLs for this tracking event."
    }
},
"vastAd": {
    "$id": "#/properties/vastAd",
    "type": "object",
    "title": "vastAd",
    "description": "Information about a single ad parsed from the VAST response.",
    "required": [
        "vastAdId",
        "adSystem",
        "adTitle",
        "creativeId",
        "creativeAdId",
        "duration",
        "vastMediaFiles",
        "trackingEvents"
    ],
    "additionalProperties": false,
    "properties": {
        "vastAdId": {
            "$id": "#/properties/vastAd/vastAdId",
            "type": "string",
            "description": "The value of the id attribute of the <code>Ad</code> tag in
the VAST response",
            "examples": [
                "ad1"
            ]
        },
        "adSystem": {"$ref": "#/definitions/adSystem" } ,
        "adTitle": {
            "$id": "#/properties/vastAd/adTitle",
            "type": "string",
            "description": "The media files that are available for the ad in the VAST
response.",
            "examples": [
                "External NCA1C1L1 LinearInlineSkippable"
            ]
        },
        "creativeId": {
            "$id": "#/properties/vastAd/creativeId",
            "type": "string",

```

```

    "description": "The value of the id attribute of the <code>Creative</code>
tag in the VAST response.",
    "examples": [
        "creative1"
    ]
},
"creativeAdId": {
    "$id": "#/properties/vastAd/creativeAdId",
    "type": "string",
    "description": "The value of the adId attribute of the <code>Creative</code>
tag in the VAST response."
},
"duration": {
    "$id": "#/properties/vastAd/duration",
    "type": "number",
    "description": "The approximate duration of the ad, based on the
<code>duration</code> tag in the <code>linear</code> element of the VAST response.",
    "examples": [
        30,
        30.0
    ]
},
"vastMediaFiles": {
    "$id": "#/properties/vastAd/vastMediaFiles",
    "type": "array",
    "description": "The list of available media files for the ad in the VAST
response.",
    "items": {
        "$id": "#/properties/vastAd/vastMediaFiles/items",
        "type": "object",
        "title": "vastMediaFile",
        "description": "Information about a media file for the ad.",
        "required": [
            "uri",
            "id",
            "delivery",
            "type",
            "apiFramework",
            "width",
            "height",
            "bitrate"
        ],
        "additionalProperties": false,
        "properties": {

```

```

    "uri": { "$ref": "#/definitions/adMezzanineUri" },
    "id": {
      "$id": "#/properties/vastAd/vastMediaFiles/items/properties/id",
      "type": "string",
      "description": "The value of the id attribute of the <code>MediaFile</code> tag.",
      "examples": [
        "GDFP"
      ]
    },
    "delivery": {
      "$id": "#/properties/vastAd/vastMediaFiles/items/properties/delivery",
      "type": "string",
      "description": "The protocol used for the media file, set to either progressive or streaming.",
      "examples": [
        "progressive"
      ]
    },
    "type": {
      "$id": "#/properties/vastAd/vastMediaFiles/items/properties/type",
      "type": "string",
      "description": "The MIME type of the media file, taken from the type attribute of the <code>MediaFile</code> tag.",
      "examples": [
        "video/mp4"
      ]
    },
    "apiFramework": {
      "$id": "#/properties/vastAd/vastMediaFiles/items/properties/apiFramework",
      "type": "string",
      "description": "The API framework needed to manage the media file. Example: <code>VPAID</code>."
    },
    "width": {
      "$ref": "#/definitions/width"
    },
    "height": {
      "$ref": "#/definitions/height"
    },
    "bitrate": {
      "$ref": "#/definitions/bitrate"
    }
  }

```

```

    }
  }
},
"trackingEvents": { "$ref": "#/definitions/trackingEvents" },
"vastAdTagUri": {
  "$id": "#/properties/vastAd/vastAdTagUri",
  "type": "string",
  "description": "The VMAP-specific redirect URI for an ad.",
  "examples": [
    "https://ads.redirect.com/redirect1"
  ]
}
}
},
"eventPreAdsRequestHookSummary": {
  "$id": "#/definitions/eventPreAdsRequestHookSummary",
  "required": [
    "eventType",
    "eventId",
    "functionId",
    "executionTimeMs",
    "status"
  ],
  "properties": {
    "eventType": {
      "type": "string",
      "const": "PRE_ADS_REQUEST_HOOK_SUMMARY"
    }
  }
}
},
"eventPreAdsRequestHookError": {
  "$id": "#/definitions/eventPreAdsRequestHookError",
  "required": [
    "eventType",
    "eventId",
    "functionId",
    "executionTimeMs",
    "errorType",
    "cause"
  ],
  "properties": {
    "eventType": {

```



```

        "type": "string",
        "const": "PRE_ADS_REQUEST_HOOK_ERROR"
    }
}
},

"eventPreAdsRequestFunctionCompleted": {
    "$id": "#/definitions/eventPreAdsRequestFunctionCompleted",
    "required": [
        "eventType",
        "eventId",
        "functionId",
        "functionType",
        "executionTimeMs"
    ],
    "properties": {
        "eventType": {
            "type": "string",
            "const": "PRE_ADS_REQUEST_FUNCTION_COMPLETED"
        }
    }
},

"eventPreAdsRequestFunctionError": {
    "$id": "#/definitions/eventPreAdsRequestFunctionError",
    "required": [
        "eventType",
        "eventId",
        "functionId",
        "functionType",
        "executionTimeMs",
        "errorType",
        "cause"
    ],
    "properties": {
        "eventType": {
            "type": "string",
            "const": "PRE_ADS_REQUEST_FUNCTION_ERROR"
        }
    }
}
}
}
}

```

AWS Elemental MediaTailor マニフェストログの説明とイベントタイプ

以下のセクションでは、MediaTailor がマニフェストのリクエストと受信時にオリジンサーバーとのイベントを記述するために出力するログについて説明します。これらはManifestServiceログです。

トピック

- [ManifestService のイベント](#)
- [マニフェストログのプロパティ](#)

ManifestService のイベント

次のイベントは、MediaTailor とオリジンとのやり取り中に出力されます。

Log	説明
CONFIG_SECURITY_ERROR	MediaTailor 設定にセキュリティ上の問題があります。
CONFIG_SYNTAX_ERROR	オリジンパスとアセットパスの URL の形式が正しくありません。
CONNECTION_ERROR	オリジンへの MediaTailor 接続が拒否または失敗しました。
GENERATED_MANIFEST	MediaTailor がマニフェストを生成しました。 これらのログを受信するには、デバッグモードが有効になっている必要があります。デバッグログモードを有効にする方法などの詳細については、「」を参照してください デバッグログの生成 。
HOST_DISALLOWED	MediaTailor は、このホストへの HTTP リクエストを許可しません。

Log	説明
INCOMPATIBLE_HLS_VERSION	マニフェストは互換性のない HLS バージョンを使用します。MediaTailor にはバージョン 3 以降が必要です。
INVALID_SINGLE_PERIOD_DASH_MANIFEST	単一期間 DASH マニフェストが無効です。MediaTailor は単一期間 DASH マニフェストを通過しています。
IO_ERROR	MediaTailor がオリジンとの通信中に IO エラーが発生しました。
LAST_PERIOD_MISSING_AUDIO	オリジンオーディオまたはビデオのずれ AdaptationSets により、DASH マニフェストの最後の期間にすべてのオーディオがありません。再生の問題を回避するには、少なくとも次のリクエストまで最後の期間の発行を遅らせます。
LAST_PERIOD_MISSING_AUDIO_WARNING	オリジンオーディオまたはビデオのずれ AdaptationSets により、DASH マニフェストの最後の期間にすべてのオーディオがありません。最後の期間を公開する (遅延しない) ことを選択します。オーディオがないと、再生の問題が発生する可能性があります。
MANIFEST_ERROR	MediaTailor マニフェストリクエストに失敗しました。
NO_MASTER_OR_MEDIA_PLAYLIST	オリジンレスポンスには、プライマリプレイリストまたはメディアプレイリストは含まれません。
NO_MASTER_PLAYLIST	オリジンレスポンスには、予想されるプライマリプレイリストが含まれていません。
NO_MEDIA_PLAYLIST	オリジンレスポンスには、予想されるメディアプレイリストが含まれていません。

Log	説明
ORIGIN_MANIFEST	MediaTailor がオリジンマニフェストを取得しました。 これらのログを受信するには、デバッグモードが有効になっている必要があります。デバッグログモードを有効にする方法などの詳細については、「」を参照してくださいデバッグログの生成。
PARSING_ERROR	オリジンはマニフェストリクエストを解析できません。
PRE_SESSION_INIT_FUNCTION_COMPLETED	セッション前の初期化フックの個々の関数が完了しました。これはオプトインイベントタイプです。
PRE_SESSION_INIT_FUNCTION_ERROR	セッション前の初期化フックの個々の関数が失敗しました。
PRE_SESSION_INIT_HOOK_ERROR	セッション前の初期化フックの実行に失敗しました。
PRE_SESSION_INIT_HOOK_SUMMARY	成功またはエラーステータスを含む、セッション前の初期化フック実行の概要。これはオプトインイベントタイプです。
SCTE35_PARSING_ERROR	MediaTailor はマニフェスト内のSignal Binary要素を解析できません。
SESSION_INITIALIZED	セッションが初期化されました。 これらのログを受信するには、デバッグモードが有効になっている必要があります。デバッグログモードを有効にする方法などの詳細については、「」を参照してくださいデバッグログの生成。

Log	説明
TIMEOUT_ERROR	MediaTailor マニフェストリクエストがタイムアウトしました。
TRACKING_RESPONSE	MediaTailor は追跡レスポンスを提供しました。 これらのログを受信するには、デバッグモードが有効になっている必要があります。デバッグログモードを有効にする方法などの詳細については、「」を参照してください デバッグログの生成 。
UNKNOWN_ERROR	MediaTailor で不明なエラーが発生しました。
UNKNOWN_HOST	ホストが不明です。
UNSUPPORTED_SINGLE_PERIOD_DASH_MANIFEST	単一期間 DASH マニフェストはサポートされていません。MediaTailor は単一期間 DASH マニフェストを通過しています。

マニフェストログのプロパティ

このセクションでは、マニフェストログのプロパティについて説明します。

プロパティ	タイプ	必須
awsAccountId	string	true
eventTimestamp	string	true
originId	string	true
customerId	string	false
eventType	string	false
sessionId	string	false

プロパティ	タイプ	必須
originRequestUrl	string	false
mediaTailorPath	string	false
requestId	string	false
responseBody	string	false
sessionType	string (有効な値: [DASH, HLS])	false
requestNextToken	string	false
eventDescription	string	false
errorType	string	false
eventId	string	false
executionTimeMs	整数	false
functionId	string	false
functionType	string	false
http	オブジェクト	false
input	オブジェクト	false
output	オブジェクト	false
status	string	false
assetPath	string	false
originFullUrl	string	false
originPrefixUrl	string	false
additionalInfo	string	false

プロパティ	タイプ	必須
cause	string	false
response	string	false
httpCode	string	false
errorMessage	string	false
adAdsResponse	string	false
adAdsRawResponse	string	false
adAdsRequest	string	false
adNumNewAvails	string	false
generatedMediaPlay list	string	false
requestStartTime	string	false
requestEndTime	string	false
requestStartTimeEp ochMillis	string	false
requestEndTimeEpoc hMillis	string	false

AWS Elemental MediaTailor トランスコードログの説明とイベントタイプ

以下のセクションでは、広告ステッチング用のクリエイティブを準備する際にMediaTailor がトランスコードサービスでイベントを記述するために出力するログについて説明します。これらはTranscodeServiceログです。

トピック

- [TranscodeService のイベント](#)

- [トランスコードログのプロパティ](#)

TranscodeService のイベント

以下のイベントは、広告のトランスコード中に MediaTailor とのやり取り中に出力されます。

Log	説明
IMPORT_ERROR	MediaTailor はインポートジョブ中に内部エラーが発生しました (前提条件付き広告の場合)。空の広告セットを使用する。
INITIALIZED	MediaTailor は、トランスコードジョブまたはインポートジョブ (前提条件付き広告の場合) のいずれかを初期化しました。
INTERNAL_ERROR	MediaTailor で内部エラーが発生しました。空の広告セットを使用する。
MISSING_VARIANTS	バリエーションがないため、MediaTailor は広告をトランスコードできませんでした。空の広告セットを使用する。
PROFILE_NOT_FOUND	トランスコードするプロファイルがないため、MediaTailor は広告をトランスコードできませんでした。空の広告セットを使用する。
TRANSCODE_COMPLETED	動画のトランスコードが完了しました。広告は広告挿入に使用できます。
TRANSCODE_ERROR	MediaTailor でトランスコードジョブ中に内部エラーが発生しました。空の広告セットを使用する。
TRANSCODE_IN_PROGRESS	ビデオトランスコードが進行中です。トランスコードされたビデオは準備ができていません。空の広告セットを使用する。

トランスコードログのプロパティ

このセクションでは、トランスコードログのプロパティについて説明します。

プロパティ	タイプ	必須	説明
awsAccountId	string	true	セッションに使用された MediaTailor 設定の AWS アカウント ID。
eventTimestamp	string	true	リクエストの日付と時刻。
originId	string	true	MediaTailor 設定からの設定名。これは、同じく設定の一部である動画コンテンツソースとは異なります。
eventType	string	false	このログメッセージをトリガーしたイベントのコード。例えば、TRANSCODE_ERROR などです。
eventDescription	string	false	MediaTailor サービスによって提供される、このログメッセージをトリガーしたイベントの簡単な説明。デフォルトでは、これは空です。
sessionId	string	false	MediaTailor がプレイヤーセッションに割り当てた一意

プロパティ	タイプ	必須	説明
			の数値識別子。プレイヤーがセッションに対して行うすべてのリクエストは、同じセッション ID を持ちます。例えば、e039fd39-09f0-46b2-aca9-9871cc116cde などです。
creativeUniqueId	string	false	トランスコードされている広告クリエイティブの一意の識別子。
profileName	string	false	
adUri	string	false	広告クリエイティブの URI。
transcodeRequestId	string	false	このトランスコードリクエストの一意の識別子。
cacheStatus	string	false	MediaTailor がトランスコードされた広告をキャッシュしたかどうかを示します。

提供されたログを使用して AWS Elemental MediaTailor ログを送信する

提供されたログを使用して柔軟性を高め、MediaTailor が再生設定から出力するログを配信する場所を制御できます。

提供されたログでは、MediaTailor は設定に関連付けられたすべてのログアクティビティを Amazon CloudWatch Logs に送信します。その後、CloudWatch Logs は、指定したログの割合を選択した送信先に送信します。サポートされている送信先は、Amazon CloudWatch Logs ロググループ、Amazon S3 バケット、または Amazon Data Firehose ストリームです。

販売されたログはボリューム割引料金で利用できるため、CloudWatch Logs に直接ログを送信するよりもコストを節約できます。料金については、「Amazon CloudWatch 料金表」の「ログ」タブの「Vended Logs」を参照してください。 [Amazon CloudWatch](#)

販売ログを使用するには、以下を実行する必要があります。

1. [アクセス許可を追加する](#)。
2. [ログ配信先を作成する](#)。
3. [CloudWatch Logs でログ配信を設定する](#)。
4. [MediaTailor で販売ログを有効にする](#)。

提供されたログの詳細については、CloudWatch Logs [ユーザーガイド](#) の AWS 「サービスからのログ記録を有効にする」を参照してください。MediaTailor は、提供されたログの V2 をサポートしています。

ステップ 1: MediaTailor ログ配信のアクセス許可を追加する

販売ログを設定するユーザーには、MediaTailor で配信先の作成、ログ配信の設定、販売ログの有効化を行うアクセス許可が必要です。次のポリシーを使用して、提供されたログをセットアップするための適切なアクセス許可があることを確認します。

CloudWatch Logs と配信先のポリシー

Amazon CloudWatch Logs ユーザーガイドの以下のセクションでは、CloudWatch Logs と配信先でログを操作できるようにするポリシーについて説明します。複数の場所にログを送信する場合は、複数のポリシーを作成する代わりに、ポリシーステートメントを 1 つのポリシーに結合できます。

- [CloudWatch Logs に送信されたログ](#)
- [Amazon S3 に送信されたログ](#)
- [Firehose に送信されたログ](#)

コンソールからセットアップするためのポリシー

API または [コンソール](#) ではなく [コンソール](#) を使用して、販売ログの配信を設定する場合は AWS CLI、ポリシーに次の追加のアクセス許可が必要です。

MediaTailor で提供されたログのポリシー

MediaTailor で販売されたログ配信を作成、表示、または変更するには、ポリシーに次のアクセス許可が必要です。

アクセス許可の追加とポリシーの操作については、「」を参照してください [の Identity and Access Management AWS Elemental MediaTailor](#)。

ステップ 2: MediaTailor ログの配信先を作成する

ログが送信されるリソースを作成します。後のステップでログ配信の設定に使用するリソースの ARN を記録します。

CloudWatch Logs ロググループの配信先

ロググループの作成には、次のいずれかを使用します。

- コンソールについては、「Amazon [CloudWatch Logs ユーザーガイド](#)」の「[CloudWatch Logs でロググループを作成する](#)」を参照してください。Amazon CloudWatch
- API については、「Amazon CloudWatch Logs API リファレンス」の「[CreateLogGroup](#)」を参照してください。Amazon CloudWatch
- SDKs 「Amazon CloudWatch Logs [ユーザーガイドCreateLogGroup](#)」の AWS 「[SDK で使用する](#)」または [AWS CLI](#) 「」を参照してください。Amazon CloudWatch

Amazon S3 バケット配信先

バケットの作成には、次のいずれかを使用します。

- コンソール、SDKs 「Amazon Simple Storage Service ユーザーガイド」の「[バケットの作成](#)」を参照してください。
- API については、Amazon Simple Storage Service API リファレンスの[CreateBucket](#)」を参照してください。

Firehose ストリーム配信先

ストリームの作成については、「Amazon Data [Firehose デベロッパーガイド](#)」の「[コンソールから Firehose ストリームを作成する](#)」を参照してください。

ステップ 3: MediaTailor 再生設定で提供されたログを有効にする

前のステップで作成した配信先にログを送信する再生設定を作成または更新します。後のステップでログ配信の設定に使用する設定の名前を記録します。

- コンソールで、[設定の作成](#)または 設定[設定の編集](#)を編集してログ記録設定にアクセスすることで、提供されたログを有効にするには。ログ記録戦略では、「Vended logs」を選択します。
- API を通じて販売ログを有効にするには、既存の設定が必要です。ConfigureLogsForPlaybackConfiguration を使用してログ記録戦略 を追加しますVended logs。

CloudWatch Logs に直接ログを送信するレガシー MediaTailor ログ記録戦略を使用していて、提供されたログに移行する場合は、「」を参照してください[ログ記録戦略の移行](#)。

Important

ログ戦略をレガシー CloudWatch から販売済みログに変更した場合、MediaTailor は更新を保存するとすぐにこの変更を行います。提供されたログ記録が完全に設定されるまで、ログの受信は停止します。

ステップ 4: CloudWatch Logs でログ配信を設定する

CloudWatch Logs では、ログ配信の部分を表す 3 つの要素を作成する必要があります。これらの要素は、Amazon CloudWatch Logs API リファレンスの [CreateDelivery](#) で詳しく説明されています。CloudWatch Logs API を使用して配信を設定する大まかな手順は次のとおりです。

CloudWatch Logs でログ配信を設定するには (API)

1. [PutDeliverySource](#) を使用してログのソースを追加します。

は、ログを生成している再生設定DeliverySourceを表します。を作成するには、再生設定の名前が必要ですDeliverySource。

2. を使用して[PutDeliveryDestination](#)、ログが書き込まれる送信先を追加します。

は配信先DeliveryDestinationを表します。を作成するには、ロググループ、バケット、またはストリームの ARN が必要ですDeliveryDestination。

3. アカウント間でログを配信[PutDeliveryDestinationPolicy](#)する場合は、 を使用します。

配信先が再生設定とは異なるアカウントにある場合は、 `DeliveryDestinationPolicy`。このポリシーにより、CloudWatch Logs は ログを配信できます `DeliveryDestination`。

4. [CreateDelivery](#) を使用して `DeliverySource`を にリンクします `DeliveryDestination`。

は、 `DeliverySource`と 間の接続 `Delivery`を表します `DeliveryDestination`。

AWS Elemental MediaTailor ログ記録戦略の移行

ログ戦略をレガシー CloudWatch から販売済みログに変更した場合、MediaTailor は更新を保存するとすぐにこの変更を行います。ログ記録ワークフローの中断を回避するには、次の手順を使用してログ記録戦略を移行します。

1. 「[販売ログの使用](#)」に説明されている手順に従います。では [MediaTailor で販売ログを有効にする](#)、両方のログ記録戦略 (提供されたログとレガシー CloudWatch) を選択します。

MediaTailor は、提供されたログと CloudWatch Logs の両方を通じてログを直接送信します。

2. ロギング戦略と配信先に応じて、ワークフローに必要な変更を加えます。
3. ログ記録戦略からレガシー CloudWatch を再検討 [MediaTailor で販売ログを有効にする](#)して削除します。

Amazon CloudWatch Logs に直接 AWS Elemental MediaTailor ログを書き込む

MediaTailor は、セッションアクティビティ、および広告決定サーバーとのやり取りに関する詳しい情報が含まれたログを生成し、それらを Amazon CloudWatch に書き込みます。ログは、セッション中に発生したアクティビティの順次的な説明を提供します。

MediaTailor は、提供されたログを使用して、ログ配信とボリューム割引の料金に柔軟性を持たせることもできます。提供されたログの詳細については、「」を参照してください [販売ログの使用](#)。

トピック

- [Amazon CloudWatch Logs のアクセス許可](#)
- [AWS Elemental MediaTailor チャンネルアセンブリの「As Run」ログ](#)
- [AWS Elemental MediaTailor Amazon CloudWatch Logs Insights での ADS ログ分析](#)

Amazon CloudWatch Logs のアクセス許可

AWS Identity and Access Management (IAM) を使用して、Amazon CloudWatch AWS Elemental MediaTailor へのアクセスを許可するロールを作成します。アカウントに対して CloudWatch Logs 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。

CloudWatch へのアクセスを MediaTailor に許可する

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、[Roles] (ロール)、[Create role] (ロールを作成) の順に選択します。
3. 別の AWS アカウントロールタイプを選択します。
4. アカウント ID には、AWS アカウント ID を入力します。
5. [Require external ID (外部 ID を必須にする)] を選択し、「**Midas**」と入力します。このオプションにより、リクエストに正しい `sts:ExternalId` が含まれている場合にのみサービスがロールを引き受けることができるという条件が、信頼ポリシーに自動的に追加されます。
6. [Next: Permissions] (次のステップ: 許可) を選択します。
7. このロールが実行できるアクションを指定するアクセス許可ポリシーを追加します。以下のいずれかのオプションを選択した後、[Next: Review (次へ: 確認)] を選択します。
 - Amazon CloudWatch Logs への完全なアクセス権を付与するための `CloudWatchLogsFullAccess`
 - Amazon CloudWatch への完全なアクセス権を付与するための `CloudWatchFullAccess`
8. [ロール名] に「**MediaTailorLogger**」と入力し、[ロールの作成] を選択します。
9. [ロール] ページで、作成したロールを選択します。
10. プリンシパルを更新するには、信頼関係を編集します。
 1. ロールの [Summary (概要)] ページで、[Trust relationship (信頼関係)] タブを選択します。
 2. [Edit trust relationship (信頼関係の編集)] を選択します。
 3. ポリシードキュメントで、プリンシパルを MediaTailor サービスに変更します。プリンシパルは以下ようになります。

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

ポリシー全体は以下のようになります。

4. [Update Trust Policy] (信頼ポリシーの更新) をクリックします。

AWS Elemental MediaTailor チャンネルアセンブリの「As Run」ログ

実行ログは、CloudWatch MediaTailor/Channel/AsRunLogロググループで、再生中のプログラムと広告時間枠に関する情報を表示します。

チャンネルを作成すると、実行時のログはデフォルトで無効になります。コンソールまたは AWS Command Line Interface (AWS CLI) を使用して、アカウントのチャンネルごとに実行時のログ状態を有効または無効にできます。

As Run ログを有効にすると、MediaTailor はサービスにリンクされたロールを自動的に作成し、MediaTailor が CloudWatch Logs アカウントで As Run ログを書き込んで管理できるようにします。サービスにリンクされたロールの詳細については、「[MediaTailor 向けのサービスリンクロールの使用](#)」を参照してください。

Note

As Run Log は現在、デフォルトのプログラムのみをサポートしています。現時点では、プログラムルールによって作成された alternateMedia はサポートされていません。つまり、現在 alternateMedia の実行中ログは生成されません。

トピック

- [As Run ログの有効化](#)
- [As Run ログの無効化](#)

As Run ログの有効化

As Run ログを有効にするには、チャンネル名を指定し、そのチャンネルの As Run ログタイプを有効にします。

Console

チャンネルの作成時に As Run ログを有効にするには

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/mediatailor/> で MediaTailor コンソールを開きます。
2. ナビゲーションペインで、[Channel assembly] (チャンネルアセンブリ) > [Channels] (チャンネル) とクリックします。
3. ナビゲーションバーで、[Create channel] (チャンネルを作成) をクリックします。
4. チャンネルの詳細の設定、出力の設定、アクセスコントロールペインで、必要に応じてチャンネルを設定します。
5. アクセスコントロールペインで、次へを選択します。
6. ログペインの ログタイプ で、実行として有効化 を選択して実行ログを有効にします。

チャンネルの更新時に As Run ログを有効にするには

Note

チャンネルが現在実行中の場合は、更新する前にそのチャンネルを停止する必要があります。チャンネルを停止したら、アクション > 編集を選択してチャンネルの更新を開始できます。

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/mediatailor/> で MediaTailor コンソールを開きます。
2. ナビゲーションペインで、[Channel assembly] (チャンネルアセンブリ) > [Channels] (チャンネル) とクリックします。
3. 更新するチャンネルを選択して、As Run ログを有効にします。
4. [Actions] (アクション)、[Edit] (編集) の順に選択します。
5. 「チャンネルの詳細の設定」、「出力の設定」、および「アクセスコントロール」ペインで、必要に応じてチャンネル設定を更新します。
6. アクセスコントロールペインで、次へを選択します。
7. ログペインの ログタイプ で、実行として有効化 を選択して実行ログを有効にします。

ログ記録タブから As Run ログを有効にするには

Note

チャンネルが現在実行中の場合は、Actions > Edit を選択して As Run ログを有効にする代わりに、Logging タブを使用する必要があります。

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/mediatailor/> で MediaTailor コンソールを開きます。
2. ナビゲーションペインで、[Channel assembly] (チャンネルアセンブリ) > [Channels] (チャンネル) とクリックします。
3. As Run ログを有効にするチャンネルを選択します。
4. チャンネルの名前の下にあるナビゲーションバーで、ログ記録を選択します。
5. ログ記録 > ログタイプで、As run を選択して As Run ログを有効にします。

AWS Command Line Interface (AWS CLI)

As Run ログを有効にするには

[configure-logs-for-channel](#) コマンドを実行し、必要なパラメータに適切な値を指定します。

この例は Linux、macOS、または Unix 用にフォーマットされており、読みやすさを向上させるためにバックスラッシュ (\) の行継続文字を使用しています。

```
$ aws mediatailor configure-logs-for-channel \  
--channel-name MyChannel \  
--log-types AS_RUN
```

この例は Microsoft Windows 用にフォーマットされており、読みやすさを向上させるためにキャレット (^) の行継続文字を使用しています。

```
C:\> aws mediatailor configure-logs-for-channel ^  
--channel-name MyChannel ^  
--log-types AS_RUN
```

上記の中で、

- **MyChannel** は、所有していて As Run ログを有効にするチャンネルの名前です。

コマンドが正常に実行された場合は、以下のような出力が表示されます。

```
{
  "ChannelName": "MyChannel",
  "LogTypes": [
    "AS_RUN"
  ]
}
```

As Run ログの無効化

有効になっているチャンネルの As Run ログを無効にするには、チャンネル名を指定し、そのチャンネルの As Run ログタイプを無効にします。

Console

チャンネルの更新時に As Run ログを無効にするには

Note

チャンネルが現在実行中の場合は、更新する前にそのチャンネルを停止する必要があります。チャンネルを停止したら、アクション > 編集を選択してチャンネルの更新を開始できます。

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/mediatailor/> で MediaTailor コンソールを開きます。
2. ナビゲーションペインで、[Channel assembly] (チャンネルアセンブリ) > [Channels] (チャンネル) とクリックします。
3. 更新するチャンネルを選択して、As Run ログを有効にします。
4. [Actions] (アクション)、[Edit] (編集) の順に選択します。
5. 「チャンネルの詳細の設定」、「出力の設定」、および「アクセスコントロール」ペインで、必要に応じてチャンネル設定を更新します。
6. アクセスコントロールペインで、次へを選択します。
7. ログペインのログタイプで、実行として有効にして実行ログを無効にします。

ログ記録タブから As Run ログを無効にするには

Note

チャンネルが現在実行中の場合は、Actions > Edit を選択して As Run ログを無効にする代わりに、Logging タブを使用する必要があります。

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/mediatailor/> で MediaTailor コンソールを開きます。
2. ナビゲーションペインで、[Channel assembly] (チャンネルアセンブリ) > [Channels] (チャンネル) とクリックします。
3. As Run ログを無効にするチャンネルを選択します。
4. チャンネルの名前の下にあるナビゲーションバーで、ログ記録を選択します。
5. ログ記録 > ログタイプで、As run をクリアして As Run ログを無効にします。

AWS Command Line Interface (AWS CLI)

As Run ログを無効にするには

[configure-logs-for-channel](#) コマンドを実行し、必要なパラメータに適切な値を指定します。

この例は Linux、macOS、または Unix 用にフォーマットされており、読みやすさを向上させるためにバックスラッシュ (\) の行継続文字を使用しています。

```
$ aws mediatailor configure-logs-for-channel \  
--channel-name MyChannel \  
--log-types
```

この例は Microsoft Windows 用にフォーマットされており、読みやすさを向上させるためにキャレット (^) の行継続文字を使用しています。

```
C:\> aws mediatailor configure-logs-for-channel ^  
--channel-name MyChannel ^  
--log-types
```

上記の中で、

- **MyChannel** は、所有していて As Run ログを無効にするチャンネルの名前です。

コマンドが正常に実行された場合は、以下のような出力が表示されます。

```
{
  "ChannelName": "MyChannel",
  "LogTypes": []
}
```

AWS Elemental MediaTailor Amazon CloudWatch Logs Insights での ADS ログ分析

Amazon CloudWatch Logs Insights を使用して、AWS Elemental MediaTailor 広告決定サーバー (ADS) ログを表示およびクエリできます。MediaTailor は、通常の処理とエラー状態に関するイベントログを CloudWatch に送信します。ログは JSON スキーマに従っています。CloudWatch Logs Insights を使用して、時間枠ごとにログを選択してから、それらに対してクエリを実行できます。

一般情報については、「[CloudWatch Logs Insights を使用したログデータの分析](#)」を参照してください。

Note

ログにアクセスするには、Amazon CloudWatch にアクセスするための許可が必要です。手順については、「[Amazon CloudWatch Logs のアクセス許可](#)」を参照してください。

CloudWatch コンソールを使用して ADS ログを表示してクエリする

1. CloudWatch コンソール (<https://console.aws.amazon.com/cloudwatch/>) を開きます。
2. ナビゲーションペインの [Logs] (ログ) で [Insights] (インサイト) をクリックします。
3. 検索バーにと入力し **AdDec**、ドロップダウンリストから **MediaTailor/AdDecisionServerInteractions** を選択します。
4. (オプション) 調査する期間を調整します。
5. (オプション) ダイアログボックスでクエリを変更します。一般的なガイダンスについては、「[CloudWatch Logs Insights のクエリ構文](#)」を参照してください。MediaTailor ADS に関するクエリの例については、「[ADS ログのクエリ](#)」を参照してください。
6. [Run query] (クエリの実行) を選択します。クエリには数秒かかる場合があります、その間は [Run query] (クエリの実行) の代わりに [Cancel] (キャンセル) が表示されます。

7. (オプション) 結果を CSV ファイルとしてエクスポートするには、[アクション]、[クエリ結果のダウンロード (CSV)] の順に選択します。

Note

コンソールは、クエリ結果で返されるレコードとエクスポートされるレコードの数を制限するため、バルクデータには API、AWS Command Line Interface (AWS CLI)、または SDK を使用します。

トピック

• [ADS ログのクエリ](#)

ADS ログのクエリ

CloudWatch Logs は、ログをクエリするための豊富なオプションを提供します。クエリ構文に関する詳細情報については、「[CloudWatch Logs Insights のクエリ構文](#)」を参照してください。このセクションには、ADS ログクエリの使用を開始するための一般的なクエリの例が記載されています。すべてのクエリは、現在の時間範囲設定のログに対して実行されます。

以下のクエリは、ADS ログからすべての情報を取得します。

```
fields @timestamp, eventType, sessionId, requestId, @message
| sort sessionId, @timestamp asc
```

以下のクエリは、ADS に対するすべてのリクエストを取得します。このクエリは、MediaTailor ログのリクエストヘッダーコンテンツを取得する方法を示しています。

```
fields @timestamp, adsRequestUrl, requestHeaders.0.value as @userAgent,
requestHeaders.1.value as @xForwardedFor, sessionId, requestId
| filter eventType = "MAKING_ADS_REQUEST"
| sort @timestamp asc
```

以下のクエリは、MediaTailor が所定のセッションに挿入した広告を取得します。

```
fields @timestamp, sessionId, requestId, @message
| filter eventType = "FILLED_AVAIL"
| sort @timestamp asc
```

以下のクエリは、MediaTailor がプレイヤーに代わって呼び出した追跡 URL を取得します。

```
fields @timestamp, beaconInfo.trackingEvent, beaconInfo.beaconUri,  
  beaconInfo.headers.0.value as @userAgent, beaconInfo.headers.1.value as  
  @xForwardedFor, sessionId, requestId  
| filter eventType = "BEACON_FIRED"  
| sort @timestamp asc
```

以下のクエリは、sessionId で結果をフィルタリングすることによって、特定の再生セッションの情報を取得します。

```
fields @timestamp, eventType, sessionId, requestId, @message  
| filter sessionId = "0aaf6507-c6f9-4884-bfe7-f2f841cb8195"  
| sort @timestamp asc
```

以下のクエリは、requestId で結果をフィルタリングすることによって、単一のリクエストの情報を取得します。

```
fields @timestamp, eventType, sessionId, requestId, @message  
| filter requestId = "f5d3cf39-6258-4cf1-b3f6-a34ff8bf641d"  
| sort @timestamp asc
```

以下のクエリは、ログに記録された各イベントタイプのログエントリの数を取得します。

```
fields eventType  
| stats count() as @eventCount by eventType
```

以下のクエリは、広告をスキップしたすべての avail に対する avail ID とスキップされた広告のリストを取得します。

```
fields avail.availId  
| parse @message '"skippedAds":[*]' as @skippedAdsList  
| filter ispresent(@skippedAdsList)
```

AWS Elemental MediaTailor ログのボリュームの制御

MediaTailor 広告挿入セッションログは、冗長になる場合があります。ログコストを削減するため、MediaTailor が Amazon CloudWatch Logs に送信するセッションログの割合を定義すること

ができます。例えば、再生設定に 1,000 個の広告挿入セッションがあり、有効な割合の値を 60 に設定する場合、MediaTailor は 600 個のセッションに関するログを CloudWatch Logs に送信します。MediaTailor は、ログを送信するセッションをランダムに決定します。特定のセッションに関するログを表示するには、[デバッグログモード](#)を使用できます。

ログの割合を設定すると、MediaTailor は、アカウントに CloudWatch のログを書き込むために必要な許可を MediaTailor に付与するサービスリンクロールを自動的に作成します。MediaTailor がサービスリンクロールを使用する方法については、「[MediaTailor 向けのサービスリンクロールの使用](#)」を参照してください。

ログ設定の作成

MediaTailor が CloudWatch Logs に書き込むセッションログの割合を制御するには、再生設定のログ設定を作成します。ログ設定を作成するときは、再生設定名と有効な割合の値を指定します。

Console

既存の再生設定のログ設定を作成するには

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/mediatailor/> で MediaTailor コンソールを開きます。
2. [Playback configuration] (再生設定) ペインで、ログ設定を作成する再生設定を選択します。
3. [Edit] (編集) をクリックします。
4. [Log configuration] (ログ設定) で、[percent enabled] (有効な割合) 値を指定します。

新しい再生設定のログ設定を作成するには

- [ログ設定](#) の手順を実行します。

AWS Command Line Interface (AWS CLI)

既存の再生設定のログ設定を作成する

を使用してログ設定を作成するには AWS CLI、[configure-logs-for-playback-configuration](#) コマンドを実行し、必要なパラメータに適切な値を指定します。

この例は Linux、macOS、または Unix 用にフォーマットされており、読みやすさを向上させるためにバックスラッシュ (\) の行継続文字を使用しています。


```
$ aws mediatailor configure-logs-for-playback-configuration \  
--percent-enabled 10 \  
--playback-configuration-name MyPlaybackConfiguration
```

この例は Microsoft Windows 用にフォーマットされており、読みやすさを向上させるためにキャレット (^) の行継続文字を使用しています。

```
C:\> aws mediatailor configure-logs-for-playback-configuration ^  
--percent-enabled 10 ^  
--playback-configuration-name MyPlaybackConfiguration
```

上記の中で、

- *percent-enabled* は、MediaTailor が CloudWatch Logs に送信する再生設定セッションログの割合です。
- *playback-configuration-name* は、ログ設定を作成している再生設定の名前です。

コマンドが正常に実行された場合は、以下のような出力が表示されます。

```
{  
  "PercentEnabled": 10,  
  "PlaybackConfigurationName": "MyPlaybackConfiguration"  
}
```

新しい再生設定のログ設定を作成するには

- [put-playback-configuration](#) コマンドには `configure-logs-for-playback-configuration` オプションを使用します。

ログ設定の非アクティブ化

ログ設定を作成した後は、ログ設定を削除することはできません。無効にすることしかできません。ログ設定を無効にするには、MediaTailor コンソールまたは API を使用して、有効率の値を 0 に設定します。そうすることで、その再生設定のすべてのセッションログがオフになります。

アカウント内のログ設定用に MediaTailor が使用するサービスリンクロールを削除する場合は、まずすべてのログ設定を非アクティブにする必要があります。サービスリンクロールの削除方法については、「[MediaTailor 向けのサービスリンクロールの使用](#)」を参照してください。

Console

再生設定のログ設定を非アクティブにする

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/mediatailor/> で MediaTailor コンソールを開きます。
2. [Playback configuration] (再生設定) ペインで、ログ設定を非アクティブにする再生設定を選択します。
3. [編集] を選択します。
4. [Log configuration] (ログ設定) で、[percent enabled] (有効な割合) 値を 0 に設定します。そうすることで、この再生設定のすべてのセッションログがオフになります。
5. [Save] (保存) をクリックします。

AWS Command Line Interface (AWS CLI)

ログ設定を非アクティブにする

- [configure-logs-for-playback-configuration](#) コマンドを使用して、percent-enabled 値を 0 に設定します。

AWS Elemental MediaTailor ログとイベントのフィルタリング

MediaTailor の再生設定から出力されるログには、再生セッション中に発生するさまざまなアクティビティに関する情報が含まれます。これらのアクティビティは、ログのイベントタイプで識別されます。多くのイベントはデフォルトでログに記録されます。Amazon CloudWatch のログのコストを制御するために、MediaTailor が出力するログを指定できます。

MediaTailor では、ログフィルタリングを制御できるため、以下を実行できます。

- ログから除外するログイベントを指定する
- 広告決定サーバー (ADS) からの raw レスポンスのログ記録を有効にする

これらのログフィルタリング設定は、再生セッションごとに個別に設定することも、再生設定のすべての再生セッションのデフォルトとして設定することもできます。

- セッションごとにログをフィルタリングするには、再生セッションの初期化リクエストにクエリパラメータを含めます。

- 再生設定ごとにログをフィルタリングするには、MediaTailor コンソールまたは API を使用して再生設定の設定を指定します。

以下のセクションでは、セッションと再生設定でログフィルタリングを有効にする手順について説明します。

セッションごとのログフィルター

各セッションのカスタマイズされたログ詳細レベルを定義するには、最初のサーバー側またはクライアント側の再生セッションリクエストに次のパラメータを追加します。パラメータに値を追加して、含めるイベントまたは除外するイベントをカンマ区切り形式で表します。

- `aws.adsInteractionLogPublishOptInEventTypes` は、特定の広告決定サーバー (ADS) インタラクションのログを受信します。
- `aws.adsInteractionLogExcludeEventTypes` は、特定の ADS インタラクションのログの受信を停止します。
- `aws.manifestServiceLogExcludeEventTypes` は、特定のマニフェストサービスインタラクションのログの受信を停止します。

MediaTailor が出力するログとイベントタイプのリストについては、「[ADS ログ](#)」「[マニフェストログ](#)」を参照してください[トランスコードログ](#)。

ログフィルタリングのクエリパラメータを渡さない場合、MediaTailor はすべてのログを配信先に書き込みます。

Exampleログフィルターを使用したサーバー側のセッションの初期化

マニフェストログと ADS ログMAKING_ADS_REQUESTから GENERATED_MANIFESTおよび PARSING_ERRORイベントを除外するには、セッション初期化リクエストは次のようになります。

```
GET <mediatailorURL>/v1/master/<hashed-account-id>/<origin-id>/index.m3u8?  
aws.logMode=DEBUG&aws.manifestServiceLogExcludeEventTypes=GENERATED_MANIFEST,PARSING_ERROR&aws.
```

ADS から raw ログを有効にするには、`AdsInteractionPublishOptInEventType` パラメータ `RAW_ADS_RESPONSE` の値を含めます。

```
GET <mediatailorURL>/v1/master/<hashed-account-id>/<origin-id>/index.m3u8?  
aws.adsInteractionPublishOptInEventType=RAW_ADS_RESPONSE
```

Example ログフィルターを使用したクライアント側のセッションの初期化

クライアント側のセッション初期化中にログイベントを除外するには、MediaTailor へのクライアントの POST リクエストに `availSuppression` および `ログタイプパラメータ` を含めます。クライアント側の再生セッションリクエストを作成する方法の詳細については、「[クライアント側の広告追跡](#)」を参照してください。次の例では、マニフェストログ `CONFIG_SECURITY_ERROR` と ADS ログ `MAKING_ADS_REQUEST` から および `PARSING_ERROR` イベントを除外します。

```
POST parent.m3u8
{
  "adsInteractionLog": {
    ...
    "excludeEventTypes": [
      "MAKING_ADS_REQUEST"
    ]
  },
  "manifestServiceLog": {
    ...
    "excludeEventTypes": [
      "GENERATED_MANIFEST",
      "PARSING_ERROR"
    ]
  },
  "logMode": "DEBUG"
}
```

ADS から raw ログを有効にするには、`publishOptInEventTypes` パラメータ `RAW_ADS_RESPONSE` の値を含めます。

```
POST parent.m3u8
{
  "adsInteractionLog": {
    "publishOptInEventTypes": ["RAW_ADS_RESPONSE"],
    "excludeEventTypes": [
      "MAKING_ADS_REQUEST"
    ]
  },
  "manifestServiceLog": {
    ...
    "excludeEventTypes": [
      "GENERATED_MANIFEST",
      "PARSING_ERROR"
    ]
  }
}
```

```
    ],  
    },  
    "logMode": "DEBUG"  
  }  
}
```

再生ごとの設定ログフィルター

再生設定の設定を使用して、MediaTailor がこの再生設定からデフォルトとして出力するログイベントタイプを定義します。MediaTailor は、セッション初期化リクエストのクエリパラメータのフィルタリングを含まないすべてのセッションに対して、これらのデフォルトのログフィルタリング設定を使用します。

以下を実行できます。

- 特定の広告決定サーバー (ADS) インタラクションのログを受信します。
- 特定の ADS インタラクションのログを除外します。
- 特定のマニフェストサービスインタラクションのログを除外します。

MediaTailor コンソールからこれらの設定を行うには、「」を参照してください[設定の作成](#)。MediaTailor API については、API AWS Elemental MediaTailor リファレンス[PutPlaybackConfiguration](#)の「」を参照してください。

MediaTailor が出力するログとイベントタイプのリストについては、「」、[ADS ログ](#)「」、[マニフェストログ](#)「」を参照してください[トランスコードログ](#)。

AWS Elemental MediaTailor デバッグログの生成

デバッグログを使用して、MediaTailor 広告挿入の再生セッション問題をトラブルシューティングします。デバッグログを生成するには、MediaTailor に対するプレイヤーのリクエストのログモードをデバッグに設定します。サーバー側のレポートの場合は、再生リクエストでログモードを設定します。クライアント側のレポートの場合は、セッション開始リクエストでログモードを設定します。

ログモードがデバッグに設定されている場合、MediaTailor はすべてのログイベントタイプを CloudWatch Logs に書き込みます。ログは、以下のイベントに関する情報を提供します。デバッグログで生成されるデータの完全なリストについては、「[デバッグログのフィールド](#)」を参照してください。

- オリジンインタラクション – MediaTailor とオリジンサーバーとのやり取りに関する詳細。例えば、オリジンマニフェストレスポンス、マニフェストタイプ、およびオリジン URL などです。

- 生成されたマニフェスト - MediaTailor からの再生セッションレスポンスに関する詳細情報。例えば、MediaTailor が生成するマニフェストなどです。
- 開始されたセッション - セッション ID などのセッション開始に関する詳細情報。

セッションごとに受け取るロギイベントタイプをカスタマイズするには、「」を参照してください [ログとイベントのフィルタリング](#)。

前提条件

ログモードをデバッグに設定するには、まず CloudWatch にログを送信するための許可を MediaTailor に付与する必要があります (まだ付与していない場合)。CloudWatch にアクセスするための許可を MediaTailor に付与したら、デバッグログモードを有効にする準備が整います。CloudWatch にアクセスするための許可を MediaTailor に付与する方法については、「[Amazon CloudWatch の許可のセットアップ](#)」を参照してください。

ログモードをデバッグに設定する方法

このセクションでは、サーバー側のレポートとクライアント側のレポートのログモードをデバッグに設定する方法を説明します。

サーバー側のレポート

サーバー側のレポートの場合、HLS または DASH MediaTailor エンドポイントに対するプレイヤーの GET HTTP 再生リクエストに `?aws.logMode=DEBUG` クエリのパラメータと値を含めます。サーバー側のレポートに関する一般情報については、「[サーバー側のレポート](#)」を参照してください。

Important

DEBUG 値では、大文字と小文字が区別されます。

`?aws.logMode=DEBUG` が含まれる再生リクエストは、以下のようになります。

Example HLS エンドポイントに対する再生リクエスト

```
GET <mediatailorURL>/v1/master/<hashed-account-id>/<origin-id>/<asset-id>?  
aws.logMode=DEBUG
```

ログモードをデバッグに設定した後は、デバッグログセッションがアクティブであることを確認することが推奨されます。デバッグセッションがアクティブであることを確認するには、その

セッション ID に関する CloudWatch のログが存在するかどうかをチェックします。セッション ID は、MediaTailor が提供する再生エンドポイントに含まれています。詳細については、「[Verify that the debug log mode is active for your playback session](#)」を参照してください。

クライアント側のレポート

クライアント側のレポートの場合は、MediaTailor の /v1/session エンドポイントに対するクライアントの POST HTTP セッション開始リクエストボディに logMode キーと DEBUG 値を含めます。クライアント側のレポートに関する一般情報については、「[クライアント側のレポート](#)」を参照してください。

Important

DEBUG 値では、大文字と小文字が区別されます。

ログモードをデバッグに設定した後は、デバッグセッションがアクティブであることを確認することが推奨されます。デバッグセッションがアクティブであることを確認するには、そのセッション ID に関連付けられた SESSION_INITIALIZED イベントが CloudWatch のログにあるかどうかを確認します。セッション ID は、MediaTailor が提供する再生エンドポイントに含まれています。詳細については、「[Verify that the debug log mode is active for your playback session](#)」を参照してください。

アクティブなデバッグセッションの最大数

アクティブなデバッグログセッションは、最大 10 個まで設定できます。プレイヤーがそのセッション開始または再生リクエストを MediaTailor に送信すると、MediaTailor がこの制限に到達しているかどうかをチェックします。到達している場合、MediaTailor は古いセッションがあるかどうかをチェックします。セッションは、一定期間内にアクセスされなかった場合に古いと見なされます。この期間は、ライブストリームの場合は 10 分、VOD ストリームの場合は 30 分です。

アクティブなデバッグログセッション数の上限に到達した場合、セッションに関するデバッグログは CloudWatch Logs に書き込まれません。セッションに関するデバッグログが CloudWatch Logs がない場合は、この上限に到達している可能性があります。制限に到達したかどうかを確認するには、「[Verify that the debug log mode is active for your playback session](#)」を参照してください。

デバッグログのフィールド

以下の表は、MediaTailor が CloudWatch に書き込むデバッグログのフィールドのリストです。

フィールド	説明
awsAccountId	AWS アカウント ID。
customerId	MediaTailor カスタマー ID。
eventTimestamp	デバッグログイベントに関連付けられた ISO 8601 タイムスタンプ。
eventType	デバッグログイベントのタイプ。 値: • ORIGIN_INTERACTION – MediaTailor とオリジンサーバーとのやり取りに関する詳細。例えば、オリジンマニフェストレスポンス、マニフェストタイプ、およびオリジン URL などです。 • GENERATED_MANIFEST – MediaTailor からの再生セッションレスポンスに関する詳細情報。例えば、MediaTailor が生成するマニフェストなどです。 • SESSION_INITIALIZED – セッション ID などのセッション開始に関する詳細情報。
originRequestUrl	このリクエスト用に取得されたオリジンサーバーの URL。
mediaTailorPath	呼び出された MediaTailor エンドポイント。最初のマニフェストリクエストで MediaTailor に渡されたパラメータが含まれます。
requestId	MediaTailor に対する特定の HTTP リクエストの ID。
responseBody	MediaTailor からのレスポンスボディ内のマニフェスト。これは、raw オリジンマニフェストか、MediaTailor によって生成されたマニフェストのどちらかになります。
sessionId	再生セッション ID。
sessionType	再生セッションのタイプ。 値: HLS、DASH

デバッグログを読み取る

MediaTailor は、Amazon CloudWatch Logs にデバッグログを書き込みます。これには、一般的な CloudWatch Logs 料金が適用されます。デバッグログの読み取りには CloudWatch Insights を使用してください。CloudWatch Logs Insights の使用方法については、AWS CloudWatch Logs ユーザーガイドの「[CloudWatch Logs Insights を使用したログデータの分析](#)」を参照してください。

Note

デバッグログが CloudWatch に表示されるには、数分かかる場合があります。ログが表示されない場合は、数分待ってから再試行してください。それでもログが表示されない場合は、アクティブなデバッグログセッションの最大数に到達している可能性があります。到達しているかどうかを確認するには、CloudWatch クエリを実行して、再生セッション向けに開始されたデバッグセッションがあったかどうかを調べます。詳細については、「[Verify that the debug log mode is active for your playback session](#)」を参照してください。

例

このセクションには、MediaTailor のデバッグログデータの読み取りに使用できるサンプルクエリが記載されています。

Example 1: 再生セッションのデバッグログモードがアクティブであることを確認する

```
fields @timestamp, @message
| filter sessionId = "32002de2-837c-4e3e-9660-f3075e8dfd90"
| filter eventType = "SESSION_INITIALIZED" # client-side reporting
or mediaTailorPath like "/v1/master" # server-side reporting HLS
or mediaTailorPath like "/v1/dash" # server-side reporting DASH
```

Example 2: オリジンからのレスポンスを表示する

```
fields @timestamp, responseBody, @message, mediaTailorPath
| filter eventType = "ORIGIN_MANIFEST" and sessionId = "32002de2-837c-4e3e-9660-f3075e8dfd90"
```

Example 3: 所定のセッションについて MediaTailor が生成したマニフェストを表示する

```
fields @timestamp, responseBody, @message
```

```
| filter mediaTailorPath like "/v1/master/" and eventType = "GENERATED_MANIFEST" and  
sessionId = "32002de2-837c-4e3e-9660-f3075e8dfd90"
```

Example 4: 所定の **requestId** に関するすべてのイベントを表示する

このクエリを使用して、オリジンマニフェストと MediaTailor によって生成されたマニフェストを表示します。

```
fields @timestamp, responseBody, @message, mediaTailorPath  
| filter requestId = "e5ba82a5-f8ac-4efb-88a0-55bed21c45b4"
```

Amazon CloudWatch メトリクス AWS Elemental MediaTailor によるモニタリング

CloudWatch を使用して AWS Elemental MediaTailor メトリクスをモニタリングできます。CloudWatch は、サービスのパフォーマンスに関する raw データを収集し、そのデータを読み取り可能なほぼリアルタイムのメトリクスに処理します。これらの統計は 15 か月間保持されるため、履歴情報にアクセスし、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できます。また、特定のしきい値を監視するアラームを設定し、これらのしきい値に達したときに通知を送信したりアクションを実行したりできます。詳細については、「[Amazon CloudWatch ユーザーガイド](#)」を参照してください。

メトリクスは、古いマニフェストを調査するときに便利です。詳細については、「[メトリクスを使用して古いマニフェストを診断する](#)」を参照してください。

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。

CloudWatch コンソールを使用してメトリクスを表示するには

1. CloudWatch コンソール (<https://console.aws.amazon.com/cloudwatch/>) を開きます。
2. ナビゲーションペインで [Metrics (メトリクス)] を選択してください。
3. [すべてのメトリクス] で、[MediaTailor] 名前空間を選択します。
4. メトリクスを表示するメトリクスディメンション (originID など) を選択します。
5. 表示する期間を指定します。

AWS Command Line Interface (AWS CLI) を使用し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/MediaTailor"
```

AWS Elemental MediaTailor CloudWatch メトリクス

AWS Elemental MediaTailor 名前空間には、次のメトリクスが含まれます。これらのメトリクスはデフォルトでユーザーのアカウントに発行されます。

チャネルアセンブリ (CA) メトリクス

次の表では、すべてのメトリクスをチャネルまたはチャネル出力で使用できます。

メトリクス	説明
4xxErrorCount	4xx エラーの数。
5xxErrorCount	5xx エラーの数。
RequestCount	リクエストの総数。トランザクション数は、プレイヤーが更新されたマニフェストをリクエストする頻度とプレイヤーの数に大きく依存します。プレイヤーリクエストのそれぞれがトランザクションとして計上されます。
TotalTime	アプリケーションサーバーがリクエストの処理にかかった時間。クライアントとネットワークとの間でバイトの受信と書き込みにかかった時間が含まれます。

サーバー側の広告挿入 (SSAI) メトリクス

次の表に、サーバー側の広告挿入メトリクスを示します。

メトリクス	説明
AdDecisionServer.Ads	ユーザー指定の CloudWatch 期間内において、広告決定サーバー (ADS) レスポンスに含まれていた広告の数。 ディメンション: • ConfigurationName • アカウントレベル (ディメンションなし)
AdDecisionServer.Duration	ユーザー指定の CloudWatch 期間内において、MediaTailor が ADS から受信したすべての広告の合計継続時間 (ミリ秒単位)。この期間は、Avail.Duration 指定したよりも長くすることができます。 ディメンション: ConfigurationName
AdDecisionServer.Errors	ユーザー指定の CloudWatch 期間内において、MediaTailor が ADS から受信した HTTP 200 以外のステータスコードを持つレスポンス、空のレスポンス、およびタイムアウトレスポンスの数。 ディメンション: ConfigurationName
AdDecisionServer.FillRate	ユーザー指定の期間中に、ADS からのレスポンスが対応する個々の ad avail を埋めた率の単純平均。 加重平均を取得するには、AdDecisionServer.Duration を Avail.Duration の割合 (%) で計算します。単純平均および加重平均の詳細については、「単純平均と加重平均」を参照してください。 ディメンション: ConfigurationName

メトリクス	説明
AdDecisionServer.Latency	MediaTailor から ADS へのリクエストの応答時間をミリ秒単位で表します。 ディメンション: ConfigurationName
AdDecisionServer.Timeouts	ユーザー指定の CloudWatch 期間中の ADS に対するタイムアウトリクエストの数。 ディメンション: ConfigurationName
AdNotReady	ユーザー指定の期間中に、内部トランスコーダーサービスによるトランスコードが未完了の広告を ADS が参照した回数。 このメトリクスの値が大きいと、全体的な Avail.FillRate が低くなる可能性があります。 ディメンション: ConfigurationName
AdsBilled	MediaTailor が挿入に基づいて顧客に請求する広告の数。 ディメンション: • ConfigurationName • アカウントレベル (ディメンションなし)
Avail.Duration	CloudWatch 期間内における、ad avail の計画された合計ミリ秒数。計画された合計は、オリジンマニフェスト内の ad avail 継続時間に基づきます。 ディメンション: • ConfigurationName • アカウントレベル (ディメンションなし)

メトリクス	説明
Avail.FilledDuration	CloudWatch 期間内に MediaTailor が広告で埋める ad avail 時間の計画された合計ミリ秒数。 ディメンション: • ConfigurationName • アカウントレベル (ディメンションなし)
Avail.FillRate	CloudWatch 期間内に MediaTailor が個々の ad avail を埋める率の計画された単純平均。 加重平均を取得するには、Avail.FilledDuration を Avail.Duration の割合 (%) で計算します。単純平均および加重平均の詳細については、「単純平均と加重平均」を参照してください。 MediaTailor が達成できる最大 Avail.FillRate は AdDecisionServer.FillRate によって制限されます。Avail.FillRate が低い場合は、AdDecisionServer.FillRate と比較してください。AdDecisionServer.FillRate が低い場合は、avail の継続時間に対して ADS が十分な広告を返していない可能性があります。 ディメンション: ConfigurationName

メトリクス	説明
Avail.Impression	MediaTailor がサーバー側のレポート中にインプレッションビーコンを発した広告の数。MediaTailor は、最終的にビーコンが配信されたかどうかにかかわらず、少なくとも 1 つのインプレッション URI を持つ広告ごとに 1 つのカウントを出力します。このメトリクスを分母として使用して、ビデオ完了率 (VCR) を計算します (「」を参照)Avail.Complete 。 ディメンション: • ConfigurationName • アカウントレベル (ディメンションなし)
Avail.Complete	MediaTailor がサーバー側のレポート中に完全なビーコンを発射した広告の数。MediaTailor は、ビーコンが最終的に配信されたかどうかにかかわらず、少なくとも 1 つの完全な追跡 URI を持つ広告ごとに 1 つのカウントを出力します。動画完成率 (VCR) を計算するには、の合計を の合計Avail.Complete で割りますAvail.Impression 。 ディメンション: • ConfigurationName • アカウントレベル (ディメンションなし)
Avail.ObservedDuration	CloudWatch 期間内に発生した、ad avail の観測された合計ミリ秒数。Avail.ObservedDuration は ad avail の終了時点で発行され、ad avail 間のマニフェストで報告されたセグメントの継続時間に基づきます。 ディメンション: ConfigurationName

メトリクス	説明
Avail.ObservedFilledDuration	CloudWatch 期間内に MediaTailor が広告で埋めた ad avail 時間の観測されたミリ秒数。 ディメンション: ConfigurationName
Avail.ObservedFillRate	CloudWatch 期間内に MediaTailor が個々の ad avail を埋めた率の観測された単純平均。 最初のCUE-INタグで HLS マニフェストに対してのみ出力されます。CUE-IN タグがない場合、MediaTailor はこのメトリクスを出力しません。 ディメンション: ConfigurationName
Avail.ObservedSlateDuration	CloudWatch 期間内に挿入されたスレートの観測された合計ミリ秒数。 ディメンション: ConfigurationName
GetManifest.Age	マニフェストの合計経過時間をミリ秒単位で表します。オリジンがマニフェストを作成するときから、MediaTailor がパーソナライズされたマニフェストを送信するときまで測定されます。 マニフェストの経過時間を測定するためのメトリクスの詳細については、「」を参照してください メトリクスを使用して古いマニフェストを診断する。 ディメンション: ConfigurationName
GetManifest.Errors	ユーザー指定の CloudWatch 期間中、MediaTailor がマニフェストを生成する間に受信されたエラーの数。 ディメンション: ConfigurationName

メトリクス	説明
GetManifest.Latency	マニフェストを生成するリクエストの MediaTailor 応答時間をミリ秒単位で表します。 マニフェストの経過時間を測定するためのメトリクスの詳細については、「」を参照してください メトリクスを使用して古いマニフェストを診断する。 ディメンション: ConfigurationName
GetManifest.MediaTailorAge	マニフェストが MediaTailor に保存された時間をミリ秒単位で表します。MediaTailor がオリジンレスポンスを受信したときから、MediaTailor がパーソナライズされたマニフェストを送信するときまで測定されます。 マニフェストの経過時間を測定するためのメトリクスの詳細については、「」を参照してください メトリクスを使用して古いマニフェストを診断する。 ディメンション: ConfigurationName

メトリクス	説明
Origin.Age	オリジンにマニフェストがあるミリ秒単位の時間。オリジンがマニフェストを作成するときから MediaTailor がオリジンリクエストを送信するときまで測定されます。 すべてのorigin.*メトリクスは、オリジンから直接実行されるリクエストに対して出力されます。キャッシュされたオリジンレスポンスに対しては出力されません。 マニフェストの経過時間を測定するためのメトリクスの詳細については、「」を参照してください メトリクスを使用して古いマニフェストを診断する。 ディメンション: ConfigurationName
Origin.Errors	ユーザー指定の CloudWatch 期間中、Media Tailor がオリジンサーバーから受信した HTTP 200 以外のステータスコードを持つレスポンスとタイムアウトレスポンスの数。 すべてのorigin.*メトリクスは、オリジンから直接実行されるリクエストに対して出力されます。キャッシュされたオリジンレスポンスに対しては出力されません。 ディメンション: ConfigurationName

メトリクス	説明
Origin.ManifestFileSizeBytes	HLS と DASH の両方のオリジンマニフェストのファイルサイズをバイト単位で指定します。通常、このメトリクスは と組み合わせて使用されますOrigin.ManifestFileSizeTooLarge 。 すべてのorigin.*メトリクスは、オリジンから直接実行されるリクエストに対して出力されます。キャッシュされたオリジンレスポンスに対しては出力されません。 ディメンション: ConfigurationName
Origin.ManifestFileSizeTooLarge	マニフェストサイズが設定された量より大きいオリジンからのレスポンスの数。通常、このメトリクスは と組み合わせて使用されますOrigin.ManifestFileSizeBytes 。 すべてのorigin.*メトリクスは、オリジンから直接実行されるリクエストに対して出力されます。キャッシュされたオリジンレスポンスに対しては出力されません。 ディメンション: ConfigurationName
Origin.Timeouts	ユーザー指定の CloudWatch 期間中のオリジンサーバーに対するタイムアウトリクエストの数。 すべてのorigin.*メトリクスは、オリジンから直接実行されるリクエストに対して出力されます。キャッシュされたオリジンレスポンスに対しては出力されません。 ディメンション: ConfigurationName

メトリクス	説明
Requests	すべてのリクエストタイプにおける 1 秒あたりの同時トランザクションの数。トランザクション数は、主にプレイヤーの数と、プレイヤーが更新されたマニフェストをリクエストする頻度によって異なります。プレイヤーリクエストのそれぞれがトランザクションとして計上されます。 ディメンション: • ConfigurationName • アカウントレベル (ディメンションなし)
SkippedReason.DurationExceeded	ADS が指定された表示期間よりも長い広告期間を返したために表示に挿入されなかった広告の数。このメトリクスの値が大きいと、AdsBilled メトリクスと AdDecisionServer.Ads メトリクスの間に不一致が生じる可能性があります。広告がスキップされた理由の詳細については、「」を参照してください広告スキップのトラブルシューティング。 ディメンション: • ConfigurationName • アカウントレベル (ディメンションなし)
SkippedReason.EarlyCueIn	早期のためにスキップされた広告の数CUE-IN。 ディメンション: • ConfigurationName • アカウントレベル (ディメンションなし)

メトリクス	説明
<code>SkippedReason.ImportError</code>	インポートジョブでエラーが発生したためにスキップされた広告の数。 ディメンション: • <code>ConfigurationName</code> • アカウントレベル (ディメンションなし)
<code>SkippedReason.ImportInProgress</code>	既存のアクティブなインポートジョブが原因でスキップされた広告の数。 ディメンション: • <code>ConfigurationName</code> • アカウントレベル (ディメンションなし)
<code>SkippedReason.InternalError</code>	MediaTailor の内部エラーによりスキップされた広告の数。 ディメンション: • <code>ConfigurationName</code> • アカウントレベル (ディメンションなし)
<code>SkippedReason.NewCreative</code>	アセットがクライアントによって初めてリクエストされたために <code>avail</code> に挿入されなかった広告の数。このメトリクスの値が高いと、アセットが正常にトランスコードされるまで <code>Avail.FillRate</code>、全体が一時的に低くなる可能性があります。 ディメンション: • <code>ConfigurationName</code> • アカウントレベル (ディメンションなし)

メトリクス	説明
SkippedReason.NoVariantMatch	広告とコンテンツの間にバリエーションが一致しないためにスキップされた広告の数。 ディメンション: • ConfigurationName • アカウントレベル (ディメンションなし)
SkippedReason.PersonalizationThresholdExceeded	この設定でパーソナライゼーションしきい値設定を超える広告の期間。 ディメンション: • ConfigurationName • アカウントレベル (ディメンションなし)
SkippedReason.ProfileNotFound	トランスコードプロファイルが見つからないためにスキップされた広告の数。 ディメンション: • ConfigurationName • アカウントレベル (ディメンションなし)
SkippedReason.TranscodeError	トランスコードエラーによりスキップされた広告の数。 ディメンション: • ConfigurationName • アカウントレベル (ディメンションなし)

メトリクス	説明
SkippedReason.TranscodeInProgress	広告がまだトランスコードされていないために表示に挿入されなかった広告の数。このメトリクスの値が高いと、アセットが正常にトランスコードされるまでAvail.FillRate、一時的に全体的な が低くなる可能性があります。 ディメンション: • ConfigurationName • アカウントレベル (ディメンションなし)

Note

HLS Interstitials セッションでは、広告決定の遅延バインド性により、一部のメトリクスの動作が異なります。

- Avail.ObservedFilledDuration MediaTailor は実際のクライアント側の再生動作を観測できないAvail.FilledDurationため、 は一致します。
- Avail.ObservedSlateDuration は、再生が観察されるのではなく、アセットリストレスポンスから計画されたスレート期間を報告します。
- 「Observed」というプレフィックスが付いたメトリクスは、HLS Interstitials セッションの推定値を提供します。

関数メトリクス

再生設定で [Functions](#) を使用すると、次のメトリクスが発行されます。これらのメトリクスは、関数がライフサイクルフックに設定されているときに常に出力されます。オプトインやログの設定は必要ありません。

フックレベルのメトリクス — ライフサイクルフックの実行ごとに 1 つのデータポイント:

メトリクス	説明
PreSessionInitHook.Invocations	セッション前の初期化フックが呼び出された回数。 ディメンション: ConfigurationName
PreSessionInitHook.Errors	エラーが発生したセッション前の初期化フック実行の数。 ディメンション: ConfigurationName
PreSessionInitHook.Latency	セッション前の初期化フックの実行時間をミリ秒単位で表します。 ディメンション: ConfigurationName
PreAdsRequestHook.Invocations	pre-ads リクエストフックが呼び出された回数。 ディメンション: ConfigurationName
PreAdsRequestHook.Errors	エラーが発生した事前広告リクエストフック実行の数。 ディメンション: ConfigurationName
PreAdsRequestHook.Latency	pre-ads リクエストフックの実行時間をミリ秒単位で表します。 ディメンション: ConfigurationName

関数レベルのメトリクス — 関数の実行ごとに 1 つのデータポイント。これらのメトリクスには追加のディメンション (FunctionId、FunctionType、HookType) が含まれているため、どの関数が遅いか失敗しているかを特定できます。

メトリクス	説明
Function.Invocations	個々の関数が実行された回数。

メトリクス	説明
	ディメンション: • ConfigurationName • ConfigurationName , FunctionId , FunctionType , HookType
Function.Errors	エラーが発生した個々の関数実行の数。 ディメンション: • ConfigurationName • ConfigurationName , FunctionId , FunctionType , HookType
Function.Latency	個々の関数の実行時間をミリ秒単位で表します。 ディメンション: • ConfigurationName • ConfigurationName , FunctionId , FunctionType , HookType

単純平均と加重平均

MediaTailor からの広告リクエストに対する ADS からのレスポンスと、MediaTailor が ad avail を埋める方法に関する単純平均と加重平均を取得できます。

- 単純平均は、AdDecisionServer.FillRate および Avail.FillRate で提供されます。これらは、対象期間の個々の avail に対するフィルレート割合 (%) の平均です。単純平均では、個々の avail の継続時間の違いは考慮されません。
- 加重平均は、すべての avail 継続時間の合計に対するフィルレート (%) です。これらの値は、 $(\text{AdDecisionServer.Duration} \times 100) \div \text{Avail.Duration}$ および $(\text{Avail.FilledDuration} \times 100) \div \text{Avail.Duration}$ として計算されます。これらの平均値には、各 ad avail の継続時間における違いが反映されており、継続時間が長いものほど重みが大きくなります。

単一の ad avail のみを含む期間の場合、AdDecisionServer.FillRate によって算出される単純平均は、 $(\text{AdDecisionServer.Duration} \times 100) \div \text{Avail.Duration}$ で算出される加重平均と同等です。Avail.FillRate で算出される単純平均は、 $(\text{Avail.FilledDuration} \times 100) \div \text{Avail.Duration}$ で算出される加重平均と同等です。

例

指定した期間に次の 2 つの ad avail が含まれるとします。

- 最初の ad avail の継続時間は 90 秒です。
 - この avail に対する ADS レスポンスは、45 秒の広告を提供します (50% 満稿)。
 - MediaTailor は、利用可能な広告時間の 45 秒分を埋めます (50% 満稿)
- 2 つめの ad avail の継続時間は 120 秒です。
 - この avail に対する ADS レスポンスは、120 秒の広告を提供します (100% 満稿)。
 - MediaTailor は、利用可能な広告時間の 90 秒分を埋めます (75% 満稿)

メトリクスは以下のとおりです。

- Avail.Duration は 210 (2 つの ad avail 継続時間の合計: $90 + 120$) です。
- AdDecisionServer.Duration は 165 (2 つのレスポンス継続時間の合計: $45 + 120$) です。
- Avail.FilledDuration は 135 (2 つの埋められた継続時間の合計: $45 + 90$) です。
- AdDecisionServer.FillRate は 75% (avail ごとの満稿割合 (%) の平均: $(50\% + 100\%) \div 2$) です。これは単純平均です。
- ADS フィルレートの加重平均は 78.57% (Avail.Duration の割合としての AdDecisionServer.Duration: $(165 \times 100) \div 210$) です。この計算では、継続時間の違いが考慮されます。
- Avail.FillRate は 62.5% (avail ごとの満稿割合 (%) の平均: $(50\% + 75\%) \div 2$) です。これは単純平均です。
- MediaTailor の avail フィルレートの加重平均は 64.29% (Avail.Duration の割合としての Avail.FilledDuration: $(135 \times 100) \div 210$) です。この計算では、継続時間の違いが考慮されます。

MediaTailor が ad avail に対して達成できる最高の Avail.FillRate は 100% です。ADS は、avail で利用可能な時間よりも長い広告時間を返す場合がありますが、MediaTailor が埋めることができるのは利用可能な時間のみです。

AWS Elemental MediaTailor CloudWatch デイメンション

次のデイメンションを使用して AWS Elemental MediaTailor データをフィルタリングできます。

デイメンション	説明
ConfigurationName	メトリクスが属する設定を示します。すべてのメトリクスで使用できます。
FunctionId	関数の識別子。Function.* メトリクスでのみ使用できます。
FunctionType	関数のタイプ: CUSTOM_OUTPUT、HTTP_REQUEST、SEQUENTIAL_EXECUTOR、または CONCURRENT_EXECUTOR。Function.* メトリクスでのみ使用できます。
HookType	関数をトリガーしたライフサイクルフック: PRE_SESSION_INITIALIZATION または PRE_ADS_REQUEST。Function.* メトリクスでのみ使用できます。

メトリクスを使用して から古いマニフェストを診断する AWS Elemental MediaTailor

古いマニフェストは、最近更新されていないマニフェストです。さまざまな広告挿入ワークフローでは、さまざまな要因 (ダウストリームシステムの要件など) に基づいて、マニフェストが古くなったと見なされるまでに が経過する時間に対する耐性が異なる場合があります。Amazon CloudWatch メトリクスを使用して、ワークフローの古さの許容度を超えるマニフェストを特定し、マニフェスト更新の遅延の原因を特定できます。

次のメトリクスは、古いマニフェストとその原因を特定するのに役立ちます。

MediaTailor が出力するすべてのメトリクスについては、「」を参照してください [AWS Elemental MediaTailor CloudWatch メトリクス](#)。

メトリクス	定義	使用アイテム
GetManifest.Age	この設定の GetManifest.MediaTailorAge と の両方を含む、マニフェストの合計経過時間Origin.Age を測定します。	このメトリクスを使用して、更新しやすい値を過ぎていて古いマニフェストを特定できます。 このメトリクスにアラームを設定して、古いマニフェストが提供されているときにアラートを受け取るようにします。アラームの詳細については、「Amazon CloudWatch ユーザーガイド」の「メトリクスのアラーム」を参照してください。Amazon CloudWatch アラームを受け取ったら、Origin.Age と GetManifest.MediaTailorAge を使用して、MediaTailor またはオリジンが古さを引き起こしているかどうかを確認します。
Origin.Age	この設定で MediaTailor に送信する前に、オリジンにマニフェストがある期間を測定します。 このメトリクスは、レスポンスがコンテンツ配信ネットワーク (CDN) から送信されたときに出力されません。レスポンスは、が出力Origin.Age するオリジンから送信する必要があります。	で古いマニフェストを特定するとGetManifest.Age 、 Origin.Age メトリクスと GetManifest.MediaTailorAge メトリクスを分析して、マニフェストの古さの原因となっている を特定できます。 Origin.Age がオリジンでの一般的な処理時間よりも長い場合は、アップストリームシステムが問題の原因となっている可能性があるため、そこで診断に焦点を当てる必要があります。
GetManifest.MediaTailorAge	MediaTailor がこの設定に対してこのマニフェストを保存した時間を測定します。	で古いマニフェストを特定するとGetManifest.Age 、 GetManifest.MediaTailorAge メトリクスと Origin.Age メトリクスを分析し

メトリクス	定義	使用アイテム
		て、マニフェストの古さの原因となっている を特定できます。 GetManifest.MediaTailorAge が MediaTailor の一般的なマニフェストのパーソナライゼーション時間よりも長い場合、MediaTailor が問題を引き起こしている可能性があるため、そこで診断に焦点を当てる必要があります。 GetManifest.Latency は、MediaTailor がパーソナライズされたマニフェストを作成するのにかかる時間をさらに特定できます。
GetManifest.Latency	MediaTailor がリクエストを処理し、この設定用にパーソナライズされたマニフェストを作成するのにかかる時間を測定します。	Origin.Age と を比較GetManifest.MediaTailorAge し、MediaTailor がマニフェスト配信の遅延の原因であると判断した場合、GetManifest.Latency メトリクスを分析して、マニフェストのパーソナライゼーションプロセスがマニフェストの古さに寄与しているかどうかを判断できます。 GetManifest.MediaTailorAge は、マニフェストが MediaTailor に保存される合計時間を測定します。は、MediaTailor がリクエストに応じてマニフェストをパーソナライズするストレージ時間GetManifest.Latency を測定します。

AWS Elemental MediaTailor API コールの記録

AWS Elemental MediaTailor は AWS CloudTrail、MediaTailor のユーザー、ロール、または のサービスによって実行されたアクションを記録する AWS サービスであると統合されています。CloudTrail は、MediaTailor のすべての API コールをイベントとしてキャプチャします。キャプチャされたコールには、MediaTailor コンソールからのコールと、MediaTailor API オペレーションへのコードコールが含まれます。証跡を作成する場合は、Amazon S3 バケットへの CloudTrail イベント (MediaTailor のイベントを含む) の継続的な配信を有効にすることができます。証跡を設定しない場合でも、CloudTrail コンソールの [イベント履歴] で最新のイベントを表示できます。CloudTrail が収集した情報を使用して、MediaTailor に対して行われたリクエスト、リクエスト元の IP アドレス、リクエスト者、リクエスト日時、および追加の詳細情報を確認できます。

CloudTrail 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。

AWS Elemental MediaTailor CloudTrail の情報

CloudTrail は、AWS アカウントの作成時にアカウントで有効になります。アクティビティが発生すると AWS Elemental MediaTailor、そのアクティビティはイベント履歴の他の AWS サービスイベントとともに CloudTrail 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 Event 履歴でのイベントの表示](#)」を参照してください。

のイベントなど、AWS アカウントのイベントの継続的な記録については AWS Elemental MediaTailor、証跡を作成します。証跡により、CloudTrail はログファイルを Amazon S3 バケットに配信できます。デフォルトでは、コンソールで証跡を作成すると、証跡はすべての AWS リージョンに適用されます。証跡は、AWS パーティション内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、CloudTrail ログで収集されたイベントデータをさらに分析して処理するように他の AWS サービスを設定できます。詳細については次を参照してください:

- [AWS アカウントの証跡の作成](#)
- [AWS CloudTrail ログとの サービス統合](#)
- 「[CloudTrail の Amazon SNS 通知の設定](#)」
- [複数のリージョンから CloudTrail ログファイルを受け取る](#) および [複数のアカウントから CloudTrail ログファイルを受け取る](#)

すべての AWS Elemental MediaTailor アクションは CloudTrail によってログに記録され、[AWS Elemental MediaTailor API リファレンス](#)に記載されています。例えば、PutPlaybackConfiguration と ListPlaybackConfigurations オペレーションへのコールは、CloudTrail ログファイルにエントリを生成します。

各イベントまたはログエントリには、誰がリクエストを生成したかという情報が含まれます。アイデンティティ情報は、以下を判別するのに役立ちます。

- リクエストがルートユーザーまたは AWS Identity and Access Management (IAM) 認証情報を使用して行われたかどうか
- リクエストの送信に使用された一時的なセキュリティ認証情報に、ロールとフェデレーテッドユーザーのどちらが使用されたか
- リクエストが別の AWS サービスによって行われたかどうか

詳細については、[CloudTrail userIdentity 要素](#)を参照してください。

AWS Elemental MediaTailor ログファイルエントリについて

「トレイル」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するように設定できます。CloudTrail のログファイルは、単一か複数のログエントリを含みます。イベントは、任意の出典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、リクエストパラメータなどに関する情報が含まれます。CloudTrail ログファイルは、パブリック API 呼び出しの順序付けられたスタックトレースではないため、特定の順序では表示されません。

以下は、PutPlaybackConfiguration アクションを示す CloudTrail ログエントリの例です。

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAEXAMPLE",
    "arn": "arn:aws:iam::111122223333:user/testuser",
    "accountId": "111122223333",
    "accessKeyId": "AIDAEXAMPLE",
    "userName": "testuser"
  },
  "eventTime": "2018-12-28T22:53:46Z",
  "eventSource": "mediatailor.amazonaws.com",
  "eventName": "PutPlaybackConfiguration",
  "awsRegion": "us-west-2",
```

```

"sourceIPAddress": "1.2.3.4",
"userAgent": "PostmanRuntime/7.4.0",
"requestParameters": {
  "VideoContentSourceUrl": "http://examplevideo.com",
  "Name": "examplename",
  "AdDecisionServerUrl": "http://exampleads.com"
},
"responseElements": {
  "SessionInitializationEndpointPrefix": "https://777788889999.mediatailor.us-east-1.amazonaws.com/v1/session/AKIAIOSFODNN7EXAMPLE/examplename/",
  "DashConfiguration": {
    "ManifestEndpointPrefix": "https://777788889999.mediatailor.us-east-1.amazonaws.com/v1/dash/AKIAIOSFODNN7EXAMPLE/examplename/",
    "MpdLocation": "EMT_DEFAULT"
  },
  "AdDecisionServerUrl": "http://exampleads.com",
  "CdnConfiguration": {},
  "PlaybackEndpointPrefix": "https://777788889999.mediatailor.us-east-1.amazonaws.com",
  "HlsConfiguration": {
    "ManifestEndpointPrefix": "https://777788889999.mediatailor.us-east-1.amazonaws.com/v1/master/AKIAIOSFODNN7EXAMPLE/examplename/"
  },
  "VideoContentSourceUrl": "http://examplevideo.com",
  "Name": "examplename"
},
"requestID": "1a2b3c4d-1234-5678-1234-1a2b3c4d5e6f",
"eventID": "987abc65-1a2b-3c4d-5d6e-987abc654def",
"readOnly": false,
"eventType": "AwsApiCall",
"recipientAccountId": "111122223333"
}

```

以下は、GetPlaybackConfiguration アクションを示す CloudTrail ログエントリの例です。

```

{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAEXAMPLE",
    "arn": "arn:aws:iam::111122223333:user/testuser",
    "accountId": "111122223333",
    "accessKeyId": "AIDAEXAMPLE",

```



```
    "userName": "testuser"
  },
  "eventTime": "2018-12-28T22:52:37Z",
  "eventSource": "mediatailor.amazonaws.com",
  "eventName": "GetPlaybackConfiguration",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "1.2.3.4",
  "userAgent": "PostmanRuntime/7.4.0",
  "requestParameters": {
    "Name": "examplename"
  },
  "responseElements": null,
  "requestID": "0z1y2x3w-0123-4567-9876-6q7r8s9t0u1v",
  "eventID": "888ddd77-3322-eeww-uuii-abc123jkl343",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```

AWS Elemental MediaTailor チャネルアセンブリアラートの受信

MediaTailor は、チャネルアセンブリリソースで発生する問題または潜在的な問題に関するアラートを作成します。アラートは、問題、問題の発生日時、影響を受けるリソースを記述します。

MediaTailor [ListAlerts](#) API を使用して AWS マネジメントコンソール、AWS Command Line Interface (AWS CLI)、AWS SDKs またはプログラムでアラートを表示できます。

Important

アラートを利用できるのは、2021 年 7 月 14 日以降に作成されたチャネルアセンブリリソースのみです。

チャネルアセンブリアラート

アラートタイプ	アラートコード	アラートメッセージ	注意事項
VOD ソース	NOT_PROCESSED	MediaTailor はパッケージ設定 <i>configurationPath</i> を処理していません。	

アラートタイプ	アラートコード	アラートメッセージ	注意事項
	UNREACHABLE	URL URL にアクセスできません。	
	UNAUTHORIZED	URL はリクエストを許可していませんでした。	
	TIMEOUT	URL への接続がタイムアウトしました。	
	UNPARSABLE_MANIFEST	MediaTailor で、 URL からのマニフェストの解析中に問題が発生しました。	
	VARIANT_DURATION_MISMATCH	MediaTailor は、 URL からマニフェストを解析中に、合計期間が一致しないバリエーションを検出しました。これにより、再生中に停止する可能性があります。	マニフェストの期間はバリエーション/表現によって異なります。これにより、字幕が欠落または不正確になり、MediaTailor が広告を挿入できなくなる可能性があります。
	SEGMENT_DURATION_TOO_LONG	MediaTailor は、 url からのマニフェストの解析中に、所要時間が 30 秒を超えるセグメントを検出しました。これにより、再生中の停止、キャプションの欠落や誤り、広告の挿入ができなくなる可能性があります。	マニフェストには、30 秒を超えるセグメントが含まれています。

アラートタイプ	アラートコード	アラートメッセージ	注意事項
	TARGET_DURATION_MISMATCH	MediaTailor は、 <i>URL</i> からのマニフェストの解析中に HLS マニフェスト間で EXT-X-TARGETDURATION 値の不一致を検出しました。これにより、再生中に停止する可能性があります。	ターゲット期間は、ソース内のすべてのマニフェストで一致するわけではありません。
ソースの場所	NOT_PROCESSED	MediaTailor は resource <i>resourceName</i> を処理していません。	
プログラム	VOD_SOURCE_ALERT	このプログラムの VOD ソース <i>vodSourceName</i> には、次のアラートがあります。 <i>vodSourceAlertCode</i> : <i>vodSourceAlertMessage</i>	
	SOURCE_LOCATION_ALERT	このプログラムに含まれるソースロケーション <i>sourceLocationName</i> には、 <i>sourceLocationAlertCode</i> : <i>sourceLocationAlertMessage</i> というアラートがあります。	

アラートタイプ	アラートコード	アラートメッセージ	注意事項
	CODEC_MISMATCH	MediaTailor で <i>channelName</i> スケジュールのコーデックが一致しませんでした。 <i>programName1</i> 1 の manifest manifestUrl と <i>programName2</i> の manifest manifestUrl の不一致が <i>sourceGroupName</i> にあります。 <i>manifestUrl programName2 manifestUrl</i>	
	RESOLUTION_MISMATCH	MediaTailor で <i>channelName</i> スケジュールの解像度が一致しませんでした。 <i>programName1</i> 1 の マニフェスト manifestUrl と <i>programName2</i> の マニフェスト manifestUrl の不一致が <i>sourceGroupName</i> にあります。 <i>manifestUrl programName2 manifestUrl</i>	

アラートタイプ	アラートコード	アラートメッセージ	注意事項
	BANDWIDTH_MISMATCH	MediaTailor で <i>channelName</i> スケジュールの帯域幅が一致しませんでした。 <i>programName1</i> 1 のマニフェスト manifestUrl と programName2 のマニフェスト manifestUrl の不一致が <i>sourceGroupName</i> にあります。 <i>manifestUrl programName2 manifestUrl</i>	
	FRAMERATE_MISMATCH	MediaTailor で <i>channelName</i> スケジュールのフレームレートが一致しませんでした。 <i>programName1</i> 1 のマニフェスト manifestUrl と programName2 のマニフェスト manifestUrl の不一致が <i>sourceGroupName</i> にあります。 <i>manifestUrl programName2 manifestUrl</i>	

アラートタイプ	アラートコード	アラートメッセージ	注意事項
	TARGET_DURATION_MISMATCH	MediaTailor で <i>channelName</i> スケジュールの HLS マニフェスト間でEXT-X-TARGETDURATION 値の不一致が発生しました。 <i>programName1</i> 1 のマニフェスト manifestUrl と programName2 のマニフェスト manifestUrl の不一致が <i>sourceGroupName</i> にあります。 <i>manifestUrl programName2 manifestUrl</i>	
	SEGMENT_DURATION_MISMATCH	MediaTailor で <i>channelName</i> スケジュールのマニフェスト間でセグメント継続時間の値が不一致になりました。 <i>programName1</i> 1 のマニフェスト manifestUrl と programName2 のマニフェスト manifestUrl の不一致が <i>sourceGroupName</i> にあります。 <i>manifestUrl programName2 manifestUrl</i>	
	NO_COMMON_SEGMENT_BOUNDARY_FOR_AD_SLATE	MediaTailor は、プログラム <i>programName</i> の <i>offsetMillis</i> で広告スレートを挿入できませんでした。広告スレートの開始時間に共通のセグメント境界はありません。	

アラートタイプ	アラートコード	アラートメッセージ	注意事項
	NOT_PROCESSED	MediaTailor は resource <i>resourceName</i> を処理していません。	
	TOO_MANY_ALERTS	MediaTailor が検出したアラートが多すぎて、 <i>programName</i> にそれ以上アラートを提供しません。 <i>programName</i> のアラートを引き続き受信するには、既存のアラートをクリアします。	
チャンネル	PROGRAM_ALERT	このチャンネルに含まれるプログラム <i>programName</i> には、次のアラートがあります: <i>programAlertCode</i> : <i>programAlertMessage</i>	

マニフェスト関連のアラートのトラブルシューティング

UNPARSABLE_MANIFEST、`VARIANT_DURATION_MISMATCH`などのマニフェスト関連のアラートが発生した場合`TARGET_DURATION_MISMATCH`は、以下を確認してください。

1. ソース要件の検証: コンテンツが「」で説明されている要件を満たしていることを確認します[the section called “コンテンツソースの統合”](#)。
2. マニフェストログを確認する: マニフェストのエラー分析とイベントタイプの詳細については、「」を参照してください[the section called “マニフェストログ”](#)。
3. エラーコードを確認する: 詳細なエラー分析については、「」を参照してください[トラブルシューティング](#)。
4. マニフェスト形式を確認する: マニフェストがストリーミングプロトコル (HLS または DASH) の正しい形式仕様に従っていることを確認します。

アラートの表示

任意の MediaTailor チャンnelアセンブリリソースに関するアラートを表示できます。チャンネルとプログラムのアラートを表示すると、MediaTailor がそのチャンネルまたはプログラムに含まれるすべての関連リソースをアラートに含めます。例えば、特定のプログラムに関するアラートを表示すると、ソースロケーションと、そのプログラムに含まれる VOD ソースに関するアラートも表示されます。

アラートを表示するには、以下の手順を実行します。

Console

コンソールでアラートを表示する

1. MediaTailor コンソール (<https://console.aws.amazon.com/mediatailor/>) を開きます。
2. アラートを表示するリソースを選択します。
3. [Alerts] (アラート) タブを選択してアラートを表示します。

AWS Command Line Interface (AWS CLI)

チャンネルアセンブリリソースのアラートをリストするには、リソースの [Amazon リソースネーム \(ARN\)](#) が必要です。AWS Command Line Interface (AWS CLI) の `describe-resource_type` コマンドを使用して、リソースの ARN を取得できます。たとえば、`describe-channel` コマンドを実行して、特定のチャンネルの ARN を取得します。

```
aws mediatailor describe-channel --channel-name MyChannelName
```

次に、`aws mediatailor list-alerts` コマンドを使用して、リソースに関連付けられたアラートを一覧表示します。

```
aws mediatailor list-alerts --resource-arn arn:aws:mediatailor:region:aws-account-id:resource-type/resource-name
```

API

チャンネルアセンブリリソースのアラートをリストするには、リソースの [Amazon リソースネーム \(ARN\)](#) が必要です。リソースの ARN は、MediaTailor API で `DescribeResource` オペレーションを使用することで取得できます。たとえば、`DescribeChannel` オペレーションを使用して、特定のチャンネルの ARN を取得します。

次に、`ListAlerts` API を使用してリソースのアラートを一覧表示します。

アラートの処理

アラートが発生した場合は、でアラートを表示するか AWS マネジメントコンソール、AWS Command Line Interface (AWS CLI)、AWS SDKs、または MediaTailor アラート API を使用して、問題の考えられる原因を特定します。

問題が解決されると、MediaTailor がアラートをクリアします。

AWS Elemental MediaTailor リソースのタグ付け

タグは、AWS AWS リソースに割り当てるメタデータラベルです。各タグは、キーと値から構成されます。ユーザーが割り当てるタグでは、ユーザーがキーと値を定義します。たとえば、1 つのリソースのキーを stage と定義し、値を test と定義します。

タグは、以下のことに役立ちます。

- AWS リソースを特定して整理します。多くの AWS サービスはタグ付けをサポートしているため、異なるサービスのリソースに同じタグを割り当てて、リソースが関連していることを示すことができます。たとえば、AWS Elemental MediaTailor 設定に割り当てるのと同じタグを AWS Elemental MediaPackage チャネルとエンドポイントに割り当てることができます。
- AWS コストを追跡します。AWS Billing and Cost Management ダッシュボードでこれらのタグをアクティブ化します。はタグ AWS を使用してコストを分類し、毎月のコスト配分レポートを配信します。詳細については、「[AWS Billing ユーザーガイド](#)」の「[コスト配分タグを使用する](#)」を参照してください。
- AWS リソースへのアクセスを制御します。詳細については、[IAM ユーザーガイド](#)の「[タグを使用したアクセス制御](#)」を参照してください。

以下のセクションでは、のタグについて詳しく説明します AWS Elemental MediaTailor。

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

の次のリソースはタグ付け AWS Elemental MediaTailor をサポートしています。

- Channels
- Configurations
- SourceLocations
- VodSources

タグの制限

AWS Elemental MediaTailor リソースのタグには、次の基本的な制限が適用されます。

- リソースに割り当てることができるタグの最大数: 50
- キーの最大長 – 128 文字 (Unicode)
- 値の最大長 – 256 文字 (Unicode)
- キーと値の有効な文字 – a~z、A~Z、0~9、スペース、および特殊文字 (_ . : / = + - @)
- キーと値では大文字と小文字が区別されます
- aws: をキーのプレフィックスとしてを使用しないでください。これは AWS による使用向けに予約されています。

でのタグの管理 AWS Elemental MediaTailor

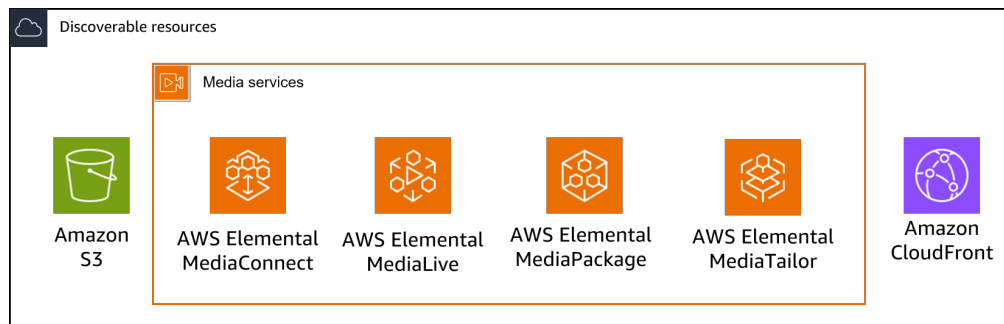
タグは、リソースのプロパティとして設定します。AWS Elemental MediaTailor API または AWS Command Line Interface () を使用してタグを追加、編集、削除できますAWS CLI。詳細については、「[AWS Elemental MediaTailor API リファレンス](#)」を参照してください。

ワークフローモニターを使用した AWS メディアサービスのモニタリング

ワークフローモニターは、AWS メディアワークフローの検出、可視化、モニタリングのためのツールです。ワークフローモニターは、AWS コンソールと API で使用できます。ワークフローモニターを使用して、シグナルマップと呼ばれるワークフローのリソースのビジュアルマッピングを検出および作成できます。Amazon CloudWatch アラームと Amazon EventBridge ルールテンプレートを作成および管理して、マッピングされたリソースをモニタリングできます。作成したモニタリングテンプレートはデプロイ可能な AWS CloudFormation テンプレートに変換され、再現性を確保できます。AWS 推奨されるアラームテンプレートは、事前定義されたベストプラクティスのモニタリングを提供します。

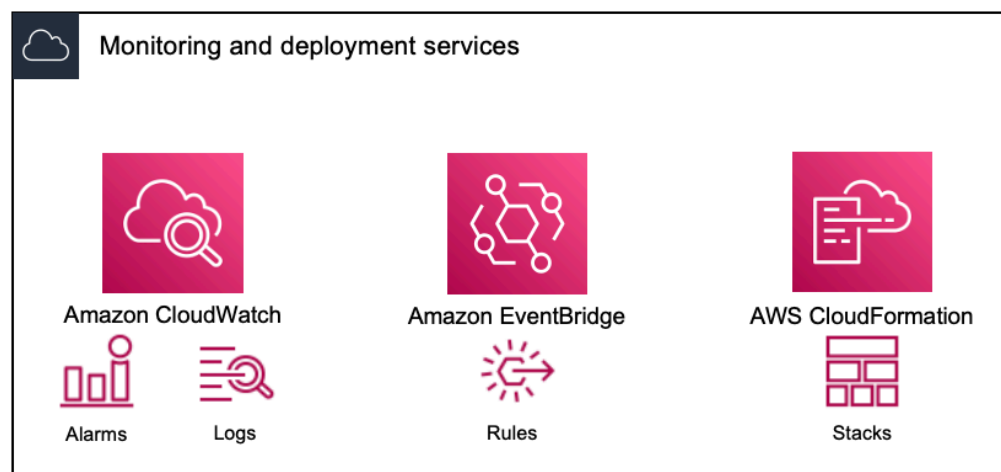
検出

シグナルマップを使用して、メディアワークフローに関連付けられた相互接続された AWS リソースを自動的に検出します。検出は、サポートされている任意のサービスリソースから開始し、ワークフローのエンドツーエンドマッピングを作成できます。シグナルマップは、スタンドアロンの視覚化ツールとして使用することも、モニタリングテンプレートで拡張することもできます。



モニタリング

カスタム CloudWatch アラームと EventBridge ルールテンプレートを作成して、メディアワークフローのヘルスとステータスをモニタリングできます。ベストプラクティスのアラームテンプレートは、ワークフローモニター環境にインポートできます。ベストプラクティスのアラームテンプレートをそのまま使用したり、ワークフローに合わせて編集したりできます。作成したテンプレートはすべて、繰り返し展開できるように「CloudFormation」テンプレートに変換されます。



Note

ワークフローモニターの使用には直接のコストはかかりません。ただし、ワークフローのモニタリングに作成および使用されるリソースにはコストがかかります。

モニタリングがデプロイされると、Amazon CloudWatch および Amazon EventBridge リソースが作成されます。AWS マネジメントコンソールを使用する場合、シグナルマップにモニタリングをデプロイする前に、作成されるリソースの数が通知されます。料金の詳細については、[CloudWatch 料金](#)と[EventBridge 料金](#)を参照してください。

ワークフローモニターは AWS CloudFormation、テンプレートを使用して CloudWatch と EventBridge リソースをデプロイします。これらのテンプレートは、デプロイプロセス中にユーザーに代わってワークフローモニターによって作成される標準クラスの Amazon Simple

Storage Service バケットに保存され、オブジェクトのストレージとリコールの料金が発生します。料金に関する詳細については、「[Amazon S3 の料金](#)」を参照してください。

AWS Elemental MediaPackage チャンネルのワークフローモニタリングマップで生成されたプレビューは、MediaPackage オリジンエンドポイントから配信され、データ転送料金が発生します。料金については、「[MediaPackage の料金](#)」を参照してください。

ワークフローモニターのコンポーネント

ワークフローモニターには 4 つの主要なコンポーネントがあります。

- CloudWatch アラームテンプレート - CloudWatch を使用してモニタリングする条件を定義します。独自のアラームテンプレートを作成するか、によって作成された事前定義されたテンプレートをインポートできます AWS。詳細については、以下を参照してください。[AWS メディアワークフローをモニタリングするための CloudWatch アラームグループとテンプレート](#)
- EventBridge ルールテンプレート - アラームがトリガーされたときに EventBridge が通知を送信する方法を定義します。詳細については、以下を参照してください。[AWS メディアワークフローをモニタリングするための EventBridge ルールグループとテンプレート](#)
- シグナルマップ - 自動プロセスを使用して、既存の AWS リソースを使用して AWS Elemental ワークフローマップを作成します。シグナルマップを使用して、ワークフロー内のリソースを検出し、それらのリソースにモニタリングをデプロイできます。詳細については、「[ワークフローモニターのシグナルマップ](#)」を参照してください
- 概要 - 概要ページでは、1 つの場所から複数のシグナルマップのステータスを直接モニタリングできます。ワークフローのメトリクス、ログ、アラームを確認します。詳細については、「[ワークフローモニターの概要](#)」を参照してください

サポートされるサービス

ワークフローモニターは、次のサービスに関連付けられたリソースの自動検出とシグナルマッピングをサポートします。

- AWS Elemental MediaConnect
- AWS Elemental MediaLive
- AWS Elemental MediaPackage
- AWS Elemental MediaTailor
- Amazon S3

- [Amazon CloudFront](#)

トピック

- [AWS メディアサービスをモニタリングするためのワークフロー 모니터の設定](#)
- [ワークフロー 모니터の使用](#)

AWS メディアサービスをモニタリングするためのワークフロー 모니터の設定

ワークフロー 모니터を初めてセットアップするには、アラームテンプレートとイベントテンプレートを作成し、メディアワークフローのモニタリングに使用されるシグナルマップを検出します。次のガイドでは、管理者レベルとオペレータレベルの IAM ロールの両方をセットアップし、ワークフローモニタリングリソースを作成し、モニタリングをワークフローにデプロイするために必要なステップについて説明します。

トピック

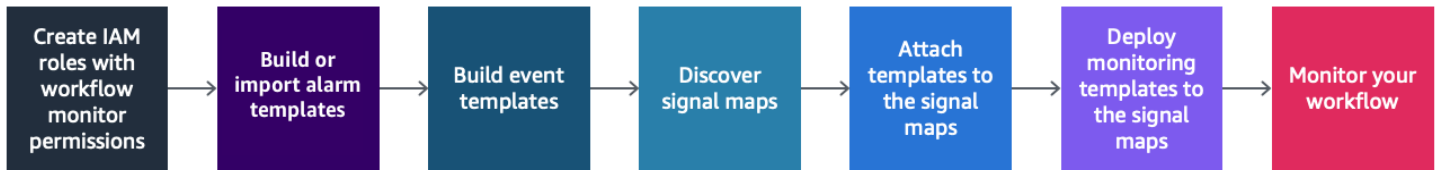
- [ワークフロー 모니터の使用を開始する](#)
- [ワークフローモニターグループとテンプレート](#)
- [ワークフローモニターのシグナルマップ](#)
- [ワークフローモニターのクォータ](#)

ワークフローモニターの使用を開始する

次の手順では、ワークフローモニターを初めて使用方法の概要を説明します。

1. セットアップワークフローは、管理者レベルロールとオペレータレベルロールの IAM アクセス許可をモニタリングします: [ワークフローモニター IAM ポリシー](#)
2. アラームテンプレートを構築するか、以下によって作成された事前定義されたテンプレートをインポートします AWS。 [CloudWatch アラーム](#)
3. EventBridge によって配信されるビルド通知イベント: [EventBridge ルール](#)
4. 既存の Elemental AWS リソースを使用してシグナルマップを検出します。 [シグナルマップ](#)
5. アラームテンプレートと通知ルールをシグナルマップにアタッチします: [テンプレートのアタッチ](#)
6. テンプレートをデプロイして、シグナルマップのモニタリングを開始します: [モニタリングテンプレートのデプロイ](#)

7. AWS コンソールの概要セクションを使用して、ワークフローモニターリソースをモニタリングおよび確認します: [概要](#)



ワークフローモニター IAM ポリシー

ワークフローモニターは複数の AWS サービスとやり取りして、シグナルマップを作成し、CloudWatch と EventBridge のリソースと CloudFormation テンプレートを構築します。ワークフローモニターはさまざまなサービスとやり取りするため、これらのサービスには特定の AWS Identity and Access Management (IAM) ポリシーを割り当てる必要があります。次の例は、管理者とオペレーターの両方の IAM ロールに必要な IAM ポリシーを示しています。

管理者IAM ポリシー

次のポリシー例は、管理者レベルのワークフローモニター IAM ポリシー用です。このロールでは、ワークフローモニターリソースと、ワークフローモニターとやり取りするサポートされているサービスリソースの作成と管理が可能になります。

JSON

```
{

  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:List*",
        "cloudwatch:Describe*",
        "cloudwatch:Get*",
        "cloudwatch:PutAnomalyDetector",
        "cloudwatch:PutMetricData",
        "cloudwatch:PutMetricAlarm",
        "cloudwatch:PutCompositeAlarm",
        "cloudwatch:PutDashboard",
        "cloudwatch:DeleteAlarms",
```

```
        "cloudwatch:DeleteAnomalyDetector",
        "cloudwatch:DeleteDashboards",
        "cloudwatch:TagResource",
        "cloudwatch:UntagResource"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "cloudformation:List*",
        "cloudformation:Describe*",
        "cloudformation:CreateStack",
        "cloudformation:UpdateStack",
        "cloudformation:DeleteStack",
        "cloudformation:TagResource",
        "cloudformation:UntagResource"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "cloudfront:List*",
        "cloudfront:Get*"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:DescribeNetworkInterfaces"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "events:List*",
        "events:Describe*",
        "events:CreateEventBus",
        "events:PutRule",
        "events:PutTargets",
        "events:EnableRule",
```

```
        "events:DisableRule",
        "events>DeleteRule",
        "events:RemoveTargets",
        "events:TagResource",
        "events:UntagResource"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "logs:Describe*",
        "logs:Get*",
        "logs:TagLogGroup",
        "logs:TagResource",
        "logs:UntagLogGroup",
        "logs:UntagResource"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "mediaconnect:List*",
        "mediaconnect:Describe*"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "medialive:*"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "mediapackage:List*",
        "mediapackage:Describe*"
    ],
    "Resource": "*"
},
{
```



```
    "Effect": "Allow",
    "Action": [
        "mediapackagev2:List*",
        "mediapackagev2:Get*"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "mediapackage-vod:List*",
        "mediapackage-vod:Describe*"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "mediatailor:List*",
        "mediatailor:Describe*",
        "mediatailor:Get*"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "resource-groups:ListGroup",
        "resource-groups:GetGroup",
        "resource-groups:GetTags",
        "resource-groups:GetGroupQuery",
        "resource-groups:GetGroupConfiguration",
        "resource-groups:CreateGroup",
        "resource-groups:UngroupResources",
        "resource-groups:GroupResources",
        "resource-groups>DeleteGroup",
        "resource-groups:UpdateGroupQuery",
        "resource-groups:UpdateGroup",
        "resource-groups:Tag",
        "resource-groups:Untag"
    ],
    "Resource": "*"
},
{
```

```
    "Effect": "Allow",
    "Action": [
        "s3:*"
    ],
    "Resource": "arn:aws:s3:::workflow-monitor-templates*"
},
{
    "Effect": "Allow",
    "Action": [
        "sns:TagResource",
        "sns:UntagResource"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "tag:Get*",
        "tag:Describe*",
        "tag:TagResources",
        "tag:UntagResources"
    ],
    "Resource": "*"
}
]
```

オペレーター IAM ポリシー

次のポリシー例は、オペレータレベルのワークフローモニター IAM ポリシー用です。このロールは、ワークフローモニターリソースと、ワークフローモニターとやり取りするサポートされているサービスリソースへの制限付き読み取り専用アクセスを許可します。

JSON

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

```
        "cloudwatch:List*",
        "cloudwatch:Describe*",
        "cloudwatch:Get*"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "cloudformation:List*",
        "cloudformation:Describe*"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "cloudfront:List*",
        "cloudfront:Get*"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:DescribeNetworkInterfaces"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "events:List*",
        "events:Describe*"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "logs:Describe*",
        "logs:Get*"
    ],
    "Resource": "*"
}
```

```
    },
    {
      "Effect": "Allow",
      "Action": [
        "mediaconnect:List*",
        "mediaconnect:Describe*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "medialive:List*",
        "medialive:Get*",
        "medialive:Describe*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "mediapackage:List*",
        "mediapackage:Describe*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "mediapackagev2:List*",
        "mediapackagev2:Get*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "mediapackage-vod:List*",
        "mediapackage-vod:Describe*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
```

```

    "Action": [
      "mediatailor:List*",
      "mediatailor:Describe*",
      "mediatailor:Get*"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:Get*",
      "s3:List*"
    ],
    "Resource": "arn:aws:s3:::workflow-monitor-templates*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "tag:Get*",
      "tag:Describe*"
    ],
    "Resource": "*"
  }
]
}

```

ワークフローモニターグループとテンプレート

ワークフローモニタリングをシグナルマップにデプロイする前に、CloudWatch アラームと EventBridge 通知のグループとテンプレートを作成する必要があります。CloudWatch テンプレートは、アラームのトリガーに使用するシナリオとしきい値を定義します。EventBridge テンプレートは、これらのアラームがどのように報告されるかを決定します。

接続されたリソースのマッピングのみが必要で、ワークフローモニターのモニタリングテンプレート機能を使用しない場合は、CloudWatch および EventBridge テンプレートなしでシグナルマップを使用できます。シグナルマップの使い方については、「[シグナルマップ](#)」を参照してください

トピック

- [AWS メディアワークフローをモニタリングするための CloudWatch アラームグループとテンプレート](#)

- [AWS メディアワークフローをモニタリングするための EventBridge ルールグループとテンプレート](#)

AWS メディアワークフローをモニタリングするための CloudWatch アラームグループとテンプレート

ワークフローモニターアラームを使用すると、シグナルマップのアラームの基盤として既存の CloudWatch メトリクスを使用できます。アラームテンプレートグループを作成して、ワークフローにとって重要なアラームのタイプをソートおよび分類できます。各アラームテンプレートグループ内で、モニタリングする特定の CloudWatch メトリクスとパラメータを含むアラームテンプレートを作成します。独自のアラームテンプレートを作成するか、 によって作成された推奨アラームテンプレートをインポートできます AWS。アラームテンプレートグループとそのグループ内にアラームテンプレートを作成したら、これらのアラームテンプレートグループの 1 つ以上をシグナルマップにアタッチできます。

まずアラームテンプレートグループを作成する必要があります。アラームテンプレートグループを作成したら、独自のテンプレートを作成するか、 によって作成された推奨テンプレートを使用できます AWS。独自のアラームテンプレートを作成する場合は、このページを続行します。推奨テンプレートのインポートの詳細については、[推奨テンプレート](#)を参照してください

このセクションでは、ワークフローモニターを使用した CloudWatch アラームの作成について説明します。CloudWatch サービスがアラームを処理する方法とアラーム コンポーネントの詳細については、[Amazon CloudWatch ユーザー ガイド](#)の CloudWatch アラームの使用を参照してください

アラームテンプレートグループの作成

アラームテンプレートグループを使用すると、ワークフローにとって重要なアラームのタイプをソートおよび分類できます。

アラームテンプレートグループを作成するには

1. ワークフローモニターコンソールのナビゲーションペインから、CloudWatch アラームテンプレート を選択します。
2. アラームテンプレートグループの作成 を選択します。
3. アラームテンプレートグループに一意のグループ名とオプションの説明 を付けます。
4. Create を選択すると、新しく作成されたアラームテンプレートグループの詳細ページが表示されます。

アラームテンプレートの作成

モニタリングする CloudWatch メトリクスとパラメータを使用してアラームテンプレートを作成できます。

アラームテンプレートを作成するには

1. アラームテンプレートグループの詳細ページから、アラームテンプレートの作成 を選択します。
2. アラームテンプレートに一意のテンプレート名とオプションの説明 を付けます。
3. メトリックの選択セクションで次のようにします:
 1. ターゲットリソースタイプ を選択します。ターゲットリソースタイプは、MediaLive と MediaPackage のチャンネルや MediaConnect のフローなど、各サービスのリソースです。
 2. メトリクス名 を選択します。これは、アラームの基盤として機能する CloudWatch メトリクスです。メトリクスのリストは、選択したターゲットリソースタイプ に応じて変わります。
4. アラーム設定セクションで、次の操作を行います。

Note

CloudWatch サービスがアラームを処理する方法とアラーム コンポーネントの詳細については、[Amazon CloudWatch ユーザー ガイド](#)の CloudWatch アラームの使用を参照してください

1. [Statistic(統計)] を選択します。これは、メトリクスのモニタリングに使用される Sum や Average などの値です。
2. [比較演算子] を選択します。このフィールドは、次のステップで設定したしきい値を参照します。
3. しきい値 を設定します。これは、Comparison Operator がステータスより大きい、小さい、または等しいかどうかを判断するために使用される数値です。
4. 期間 を設定します。これは秒単位の時間値です。期間は、統計、比較演算子、しきい値が相互作用してアラームがトリガーされるかどうかを判断する時間の長さです。
5. Datapoints を設定します。この値は、アラームのトリガーに必要なデータポイントの数を決定します。

6. 欠落データを処理する方法を選択します。この選択により、このアラームが欠落データにどのように反応するかが決まります。
5. [作成] を選択して、プロセスを完了します。

完了したアラームテンプレートの例には、次のパラメータがあります。MediaConnect フローのターゲットリソースタイプは、切断メトリクス名 に対してモニタリングされます。統計値は、「より大きい」の比較演算子と 10 のしきい値を持つ Sum に設定されます。期間は 60 秒に設定され、必要なデータポイントは 1 つのうち 1 つのみです。Treat Missing Data は「無視」に設定されます。

これらの設定の結果: ワークフローモニターはフローの切断をモニタリングします。60 秒以内に 10 回以上の切断が発生すると、アラームがトリガーされます。60 秒以内に 10 回以上の切断が発生すると、アラームがトリガーされます。

AWS メディアワークフローをモニタリングするための推奨アラームテンプレート

ワークフローモニターの推奨テンプレートは、メトリクスに適した事前定義されたアラーム設定を持つ Elemental AWS サービスメトリクスの厳選された選択です。カスタマイズされたアラームテンプレートを作成しない場合は、によって作成されたベストプラクティスのモニタリングテンプレートが推奨テンプレートで提供されます AWS。

ワークフローモニターには、サポートされている各サービスに推奨されるテンプレートグループが含まれています。これらのグループは、特定のタイプのワークフローにベストプラクティスのモニタリングを適用するように設計されています。各テンプレートグループには、サービス固有のメトリクスから設定されたアラームの厳選された選択が含まれています。例えば、MediaLive マルチプレックスワークフローに推奨されるテンプレートグループには、MediaConnect CDI ワークフローとは異なる一連の事前設定済みメトリクスがあります。

推奨されるアラームテンプレートを使用するには

1. ステップに従って [アラームテンプレートグループを作成する](#) か、既存のテンプレートグループを選択します。
2. アラームテンプレートセクションで、 のインポート を選択します。AWS 推奨テンプレートをテンプレートグループにインポートする必要があります。
3. CloudWatch アラームテンプレートグループのドロップダウンを使用して、AWS 推奨グループを選択します。これらのグループには、特定のサービスの厳選されたアラームが含まれています。

4. チェックボックスを使用してインポートするテンプレートを選択します。各テンプレートには、メトリクス、事前設定されたモニタリング値、メトリクスの説明が表示されます。テンプレートの選択が完了したら、追加ボタンを選択します。
5. 選択したテンプレートは、インポートするアラームテンプレート (複数可) セクションに移動します。選択を確認し、インポート を選択します。
6. インポートが完了すると、選択したテンプレートがテンプレートグループに追加されます。テンプレートをさらに追加する場合は、インポートプロセスを繰り返します。
7. インポートされたテンプレートは、インポート後にカスタマイズできます。アラーム設定は、アラームのニーズに合わせて変更できます。

AWS メディアワークフローをモニタリングするための EventBridge ルールグループとテンプレート

CloudWatch は Amazon EventBridge ルールを使用して通知を送信します。まず、イベントテンプレートグループを作成します。そのイベントテンプレートグループでは、通知を作成する条件と通知されるユーザーを決定するイベントテンプレートを作成します。

このセクションでは、ワークフローモニターを使用した EventBridge ルールの作成について説明します。Eventbridge サービスがルールを使用する方法の詳細については、[「Amazon EventBridge ユーザーガイド」](#)の「EventBridge ルール」を参照してください

イベントテンプレートグループの作成

イベントテンプレートグループを使用すると、ユースケースに基づいてイベントをソートおよび分類できます。

イベントテンプレートを作成するには

1. ワークフローモニターコンソールのナビゲーションペインから、EventBridge ルールテンプレート を選択します。
2. イベントテンプレートグループの作成 を選択します。
3. アラームテンプレートグループに一意のグループ名とオプションの説明 を付けます。
4. Create を選択すると、新しく作成されたアラームテンプレートグループの詳細ページが表示されます。

イベントテンプレートの作成

作成したイベントテンプレートに基づいて通知を送信できます。

イベントテンプレートグループを作成するには

1. イベントテンプレートグループの詳細ページから、イベントテンプレートの作成 を選択します。
2. イベントテンプレートに一意のテンプレート名とオプションの説明 を付けます。
3. ルール設定セクションで、次の操作を行います。
 1. [イベントタイプ] を選択します。イベントタイプを選択するときは、 によって作成された複数のイベントから選択 AWS するか、シグナルマップアクティブアラームを選択して、アラームテンプレートによって作成されたアラームを使用できます。
 2. ターゲットサービス を選択します。これにより、このイベントの通知方法を決定します。Amazon Simple Notification Service または CloudWatch ログを選択できます。
 3. ターゲットサービスを選択したら、ターゲット を選択します。これは、ターゲットサービスの選択に応じて、Amazon SNS トピックまたは CloudWatch ロググループになります。
4. [作成] を選択して、プロセスを完了します。

ワークフローモニターのシグナルマップ

シグナルマップは、メディアワークフロー内の AWS リソースのビジュアルマッピングです。ワークフローモニターを使用して、サポートされている任意のリソースタイプでシグナルマップ検出を開始できます。検出プロセス中、ワークフローモニターは接続されたすべての AWS リソースを自動的に再帰的にマッピングします。シグナルマップを作成したら、ワークフローモニターコンソールを使用して、モニタリングテンプレートのデプロイ、メトリクスの表示、マッピングされたリソースの詳細の表示などを行うことができます。

トピック

- [AWS メディアワークフローのシグナルマップの作成](#)
- [AWS メディアワークフローのシグナルマップの表示](#)
- [AWS メディアワークフローのシグナルマップにアラームテンプレートとイベントテンプレートのアタッチする](#)
- [AWS メディアワークフローのシグナルマップへのテンプレートのデプロイ](#)
- [AWS メディアワークフローのシグナルマップの更新](#)
- [AWS メディアワークフローのシグナルマップの削除](#)

AWS メディアワークフローのシグナルマップの作成

ワークフロー 모니터のシグナルマップを使用して、メディアワークフローに接続されているすべての AWS リソースのビジュアルマッピングを作成できます。

シグナルマップを作成するには

1. ワークフローモニターコンソールのナビゲーションペインから、シグナルマップ を選択します。
2. シグナルマップの作成 を選択します。
3. シグナルマップに名前と説明 を付けます。
4. 新しいシグナルマップの検出セクションには、現在のアカウントと選択したリージョンのリソースが表示されます。シグナルマップの検出を開始するリソースを選択します。選択したリソースが検出の開始点になります。
5. [作成] を選択します。検出プロセスが完了するまでしばらくお待ちください。プロセスが完了すると、新しいシグナルマップが表示されます。

Note

AWS Elemental MediaPackage チャンネルのワークフローモニターシグナルマップで生成されたプレビューは、MediaPackage オリジンエンドポイントから配信され、データ転送料金が発生します。料金については、「[MediaPackage の料金](#)」を参照してください。

AWS メディアワークフローのシグナルマップの表示

ワークフローモニターのシグナルマップを使用すると、メディアワークフローに接続されているすべての AWS リソースのビジュアルマッピングを表示できます。

シグナルマップビュー

シグナルマップを選択すると、シグナルマップのモニタリングまたは設定に使用できるビューが 2 つあります。シグナルマップのモニタリングとシグナルマップの設定は、シグナルマップコンソールセクションの右上にあるコンテキスト依存ボタンです。

ナビゲーションペインのシグナルマップセクションを使用してシグナルマップを選択すると、シグナルマップが設定ビューに表示されます。設定ビューでは、このシグナルマップにアタッチされたテン

プレートグループを変更したり、アタッチされたテンプレートをデプロイしたり、シグナルマップの基本的な詳細とタグを表示したりできます。

ナビゲーションペインの概要セクションを使用してシグナルマップを選択すると、シグナルマップがモニタリングビューに表示されます。モニタリングビューには、このシグナルマップの CloudWatch アラーム、EventBridge ルール、アラート、ログ、メトリクスが表示されます。

右上のシグナルマップのモニター/設定ボタンを選択すると、いつでもビューを変更できます。設定ビューには管理者レベルの IAM アクセス許可が必要です。必要な IAM アクセス許可は、[ここで確認](#)できます: [ワークフローモニター IAM ポリシー](#)

シグナルマップの操作

シグナルマップには、ワークフローモニターによって検出されたサポートされているすべての AWS リソースのノードが含まれます。MediaLive チャンネルや MediaPackage エンドポイントなどの特定のリソースでは、サムネイルプレビューが利用可能な場合、コンテンツのサムネイルプレビューを表示できます。

リソースノードを選択し、アクションドロップダウンメニューから選択したリソースの詳細を表示するを選択すると、関連するサービスの詳細ページが表示されます。例えば、MediaLive チャンネルを選択し、選択したリソースの詳細を表示するを選択すると、そのチャンネルの MediaLive コンソールの詳細ページが開きます。

リソースノードを選択すると、アクティブなアラームのリストがそのノードのみにフィルタリングされます。アクティブなアラームでリソースのターゲット ARN を選択すると、関連するサービスの詳細ページが表示され、選択したリソースが開かれます。

AWS メディアワークフローのシグナルマップにアラームテンプレートとイベントテンプレートをアタッチする

アラームテンプレートとイベントテンプレートを作成したら、シグナルマップにアタッチする必要があります。作成したアラームテンプレートとイベントテンプレートは、検出されたシグナルマップにアタッチできます。

シグナルマップにアラームテンプレートとイベントテンプレートをアタッチするには

1. ワークフローモニターコンソールのナビゲーションペインから、シグナルマップを選択し、操作するシグナルマップを選択します。
2. シグナルマップページの右上にある CloudWatch アラームテンプレートグループタブで、「CloudWatch アラームテンプレートグループのアタッチ」を選択します。

1. 開いた新しいセクションで、このシグナルマップに適用するすべてのアラームテンプレートグループを選択し、 の追加を選択します。これにより、選択したアラームテンプレートグループが添付された CloudWatch アラームテンプレートグループセクションに移動します。
2. 保存を選択すると、変更が保存され、シグナルマップページに戻ります。
3. シグナルマップページの右側で、EventBridge ルールテンプレートグループタブを選択し、EventBridge ルールテンプレートグループのアタッチ を選択します。
 1. 開いた新しいセクションで、このシグナルマップに適用するすべてのイベントテンプレートグループを選択し、 の追加 を選択します。これにより、選択したルールテンプレートグループが Attached EventBridge ルールテンプレートグループセクションに移動します。
 2. 保存を選択すると、変更が保存され、シグナルマップページに戻ります。
4. CloudWatch アラームと EventBridge ルールテンプレートをシグナルマップに割り当てましたが、モニタリングはまだデプロイされていません。次のセクションでは、モニタリングリソースのデプロイについて説明します。

AWS メディアワークフローのシグナルマップへのテンプレートのデプロイ

アラームテンプレートとイベントテンプレートをシグナルマップにアタッチしたら、モニタリングをデプロイする必要があります。デプロイが完了するまで、シグナルマップのモニタリングはアクティブになりません。

ワークフローモニターは、選択したシグナルマップに関連するアラームのみをデプロイします。例えば、アタッチされたアラームテンプレートグループには、MediaLive、MediaPackage、MediaConnect などの複数のサービスのアラームが含まれている場合があります。選択したシグナルマップに MediaLive リソースのみが含まれている場合、MediaPackage または MediaConnect アラームはデプロイされません。

モニタリングテンプレートをデプロイするには

1. アラームテンプレートグループとイベントテンプレートグループをシグナルマップにアタッチし、変更を保存したら、アクションドロップダウンメニューでモニターをデプロイを選択します。
2. デプロイを確認するよう求められ、作成される CloudWatch リソースと EventBridge リソースの数が表示されます。続行する場合は、デプロイ を選択します。

Note

ワークフローモニターの使用には直接のコストはかかりません。ただし、ワークフローのモニタリングに作成および使用されるリソースにはコストがかかります。

モニタリングがデプロイされると、Amazon CloudWatch および Amazon EventBridge リソースが作成されます。AWS マネジメントコンソールを使用する場合、シグナルマップにモニタリングをデプロイする前に、作成されるリソースの数が通知されます。料金の詳細については、[CloudWatch 料金](#)と[EventBridge 料金](#)を参照してください。

ワークフローモニターは AWS CloudFormation、テンプレートを使用して CloudWatch と EventBridge リソースをデプロイします。これらのテンプレートは、デプロイプロセス中にユーザーに代わってワークフローモニターによって作成される標準クラスの Amazon Simple Storage Service バケットに保存され、オブジェクトのストレージとリコールの料金が発生します。料金に関する詳細については、「[Amazon S3 の料金](#)」を参照してください。

3. デプロイのステータスは、シグナルマップの名前の横に表示されます。デプロイステータスは、CloudFormation コンソールの スタック セクションにも表示されます。リソースの作成とデプロイが数秒後、シグナルマップのモニタリングが開始されます。

AWS メディアワークフローのシグナルマップの更新

ワークフローが変更された場合、シグナルマップを再検出し、モニタリングリソースを再デプロイする必要がある場合があります。ワークフローモニターは、ワークフローを変更できない視覚化およびモニタリングツールです。シグナルマップは、ワークフローのポイントインタイム視覚化を表します。メディアワークフローの一部を追加、削除、または大幅に変更する場合は、シグナルマップを再検出することをお勧めします。シグナルマップにモニタリングリソースがアタッチされている場合は、再検出プロセス後にモニタリングを再デプロイすることをお勧めします。

シグナルマップを再検出するには

1. ワークフローモニターコンソールのナビゲーションペインから、シグナルマップを選択し、操作するシグナルマップを選択します。
2. シグナルマップの設定ビューが表示されていることを確認します。ビューの変更に関する詳細については、「[シグナルマップの表示](#)」を参照してください。
3. シグナルマップページの右上で、アクションドロップダウンメニューを選択します。Rediscover を選択します。

4. 再検出画面が表示されます。再検出するワークフローの一部であるリソースを選択します。再検出ボタンを選択します。
5. シグナルマップは、現在のワークフローに従って再構築されます。モニタリングリソースを再デプロイする必要がある場合は、このシグナルマップのページにとどまります。以前にアタッチされたモニタリングテンプレートはアタッチされたままですが、再デプロイする必要があります。

シグナルマップの再検出後にモニタリングテンプレートを再デプロイするには

1. 再検出後、更新されたシグナルマップに移動します。モニタリングテンプレートを再デプロイするには、アクションドロップダウンメニューからモニターをデプロイを選択します。
2. デプロイを確認するよう求められ、作成される CloudWatch および EventBridge リソースの数が表示されます。続行する場合は、デプロイを選択します。
3. デプロイのステータスは、シグナルマップの名前の横に表示されます。リソースの作成とデプロイが数秒後、シグナルマップのモニタリングが開始されます。

AWS メディアワークフローのシグナルマップの削除

シグナルマップが不要になった場合は、削除できます。シグナルマップにモニタリングテンプレートがデプロイされている場合、削除プロセスにより、このシグナルマップにデプロイされた CloudWatch および EventBridge リソースを削除するように求められます。デプロイされたリソースを削除しても、それらを作成したテンプレートには影響しません。このリソースの削除は、デプロイされているが使用されていない CloudWatch リソースと EventBridge リソースがないことを確認するためのものです。

シグナルマップを削除するには

1. ワークフローモニターコンソールのナビゲーションペインから、シグナルマップを選択し、削除するシグナルマップの横にあるラジオボタンを選択します。
2. [削除] ボタンを選択します。リソースの削除を確認するメッセージが表示されます。削除を選択して、モニタリングリソースの削除プロセスを開始します。
3. Monitor デプロイ列に現在のステータスが表示されます。ステータスが DELETE_COMPLETE に変わったら、削除ボタンを再度選択します。
4. シグナルマップの削除を確認するよう求められます。削除を選択してシグナルマップを続行し、削除します。

ワークフローモニターのクォータ

次のセクションには、ワークフローモニターリソースのクォータが含まれています。各クォータは「アカウントごと」ベースです。アカウントのクォータを引き上げる必要がある場合は、次の表に特に明記されていない限り、[AWS Service Quotas コンソール](#)を使用して引き上げをリクエストできます。

クォータ

リソースタイプ	クォータ
CloudWatch アラームテンプレートグループ	20
CloudWatch アラームテンプレート	200
EventBridge ルールテンプレートグループ	20
EventBridge ルールテンプレート	200
シグナルマップ	30
シグナルマップ: 単一のシグナルマップにアタッチされた CloudWatch アラームテンプレートグループ	5 このクォータを増やすことはできません。
シグナルマップ: 単一のシグナルマップにアタッチされた EventBridge ルールテンプレートグループ	5 このクォータを増やすことはできません。

ワークフローモニターの使用

ワークフローモニターコンソールの概要とシグナルマップセクションを使用して、ワークフローの現在のステータスと関連するアラーム、メトリクス、ログを確認します。

トピック

- [ワークフローモニターの概要](#)
- [ワークフローモニターの概要ログとメトリクス](#)
- [ワークフローモニターのシグナルマップの使用](#)

ワークフローモニターの概要

ワークフローモニターコンソールの概要セクションは、シグナルマップに関する情報を一目で確認できるダッシュボードです。概要セクションでは、各シグナルマップのモニタリングの現在の状態と、CloudWatch メトリクスおよび関連する CloudWatch ログを確認できます。そのシグナルマップコンソールページに移動するシグナルマップを選択できます。

概要フィルタリング

概要セクションの検索バーを使用して、コンテキスト依存制約を使用してシグナルマップのリストをフィルタリングできます。検索バーを選択すると、フィルタリングするプロパティのリストが表示されます。プロパティを選択すると、等価、含む、含まない、含まないなどの演算子が表示されます。演算子を選択すると、選択したプロパティタイプからリソースのリストが作成されます。これらのリソースのいずれかを選択すると、シグナルマップリストには、定義した制約に一致するシグナルマップのみが表示されます。

ワークフローモニターの概要ログとメトリクス

シグナルマップの CloudWatch メトリクスとログを表示するには、シグナルマップの名前の横にあるラジオボタンを選択します。メトリクスとログの両方のタブ付きインターフェイスがシグナルマップリストの下に表示されます。

CloudWatch メトリクス

選択したシグナルマップの CloudWatch メトリクスはコンテキストに依存し、そのシグナルマップワークフローで使用されるサービスに関連付けられたメトリクスのみを表示します。画面上のメトリクスツールを使用して、表示されるメトリクスの期間と期間をカスタマイズできます。

CloudWatch Logs

CloudWatch ロググループをシグナルマップに関連付けると、そのグループがここに表示されます。

ワークフローモニターのシグナルマップの使用

コンソールの概要セクションから、特定のシグナルマップを選択して、そのシグナルマップとそのアタッチされたモニタリングリソースに関する詳細情報を表示できます。

シグナルマップを選択すると、シグナルマップと、詳細情報を含むタブ付きセクションが表示されます。

- CloudWatch アラーム

- EventBridge ルール
- AWS 要素アラート
- メトリクス
- ログ
- 基本的な詳細

シグナルマップの操作

シグナルマップには、ワークフローモニターによって検出されたサポートされているすべての AWS リソースのノードが含まれます。MediaLive チャンネルや MediaPackage エンドポイントなどの特定のリソースでは、サムネイルプレビューが利用可能な場合、コンテンツのサムネイルプレビューを表示できます。

リソースノードを選択し、アクションドロップダウンメニューから選択したリソースの詳細を表示するを選択すると、関連するサービスの詳細ページが表示されます。例えば、MediaLive チャンネルを選択し、選択したリソースの詳細を表示するを選択すると、そのチャンネルの MediaLive コンソールの詳細ページが開きます。

リソースノードを選択すると、アクティブなアラームのリストがそのノードのみにフィルタリングされます。アクティブなアラームでリソースのターゲット ARN を選択すると、関連するサービスの詳細ページが表示され、選択したリソースが開かれます。

デュアルスタック (IPv6) エンドポイント AWS Elemental MediaTailor

AWS Elemental MediaTailor は、IPv4 または IPv6 経由で接続するために使用できるデュアルスタックエンドポイントをサポートしています。これらのエンドポイントは、標準の IPv4-only エンドポイントと機能的に同等です。サービスの動作は同じで、ネットワークトランスポートレイヤーのみが異なります。

デュアルスタックエンドポイントとは

デュアルスタックエンドポイントは、IPv4 アドレスと IPv6 アドレスの両方をアドバタイズします。デュアルスタックエンドポイントを使用すると、DNS はネットワーク設定とクライアント設定に基づいて IPv4 アドレスまたは IPv6 アドレスのいずれかに解決されます。MediaTailor リソース、再生設定、広告挿入動作を変更する必要はありません。

サポートされている機能

次の表に、デュアルスタックエンドポイントの MediaTailor 機能のサポートを示します。

機能	デュアルスタックサポート
コントロールプレーン APIs (すべてのリージョン)	サポート
SSAI データプレーン (セッションの初期化、マニフェスト、セグメント)	サポート
チャネルアセンブリのデータプレーン	サポート対象
AWS マネジメントコンソール	サポート
AWS SDK	近日公開予定

デュアルスタックのエンドポイント形式

MediaTailor デュアルスタックエンドポイントは `.api.aws` ドメインを使用します。

- コントロールプレーン: `mediatailor.region.api.aws`
- SSAI データプレーン: の再生設定の詳細のデュアルスタックエンドポイントセクションに表示されます AWS マネジメントコンソール。
- チャンnelアセンブリデータプレーン: のチャンネルの出力詳細に表示されます AWS マネジメントコンソール。

既存のアプリケーションは、引き続き元の IPv4-only エンドポイント (`api.mediatailor.region.amazonaws.com` コントロールプレーンの場合は、データプレーン `unique-id.mediatailor.region.amazonaws.com` の場合は) で動作します。使用可能なエンドポイントの完全なリストについては、AWS 「全般のリファレンス」の [MediaTailor エンドポイントとクォータ](#) を参照してください。

デュアルスタックのエンドポイントの使用

AWS マネジメントコンソール

デュアルスタックエンドポイントは、デュアルスタックエンドポイントセクションの再生設定の詳細ページに表示されます。これらの URLs は、プレイヤーまたは CDN 設定で使用するために直接コピーできます。

AWS Command Line Interface

API コールを行うときに、デュアルスタックのエンドポイント URL を指定します。

```
aws mediatailor get-playback-configuration \  
  --name my-config \  
  --endpoint-url https://mediatailor.us-east-1.api.aws
```

my-config を再生設定の名前に置き換え、*us-east-1* を AWS リージョンに置き換えます。

AWS SDK

SDK 設定でデュアルスタックを有効にします。

Python (boto3):

```
import boto3
```

```
client = boto3.client(  
    'mediatailor',  
    endpoint_url='https://mediatailor.us-east-1.api.aws'  
)
```

JavaScript (AWS SDK v3):

```
import { MediaTailorClient } from "@aws-sdk/client-mediatailor";  
  
const client = new MediaTailorClient({  
    useDualstackEndpoint: true  
});
```

考慮事項

デュアルスタックエンドポイントを使用する場合は、次の点に注意してください。

- デュアルスタックエンドポイントは、MediaTailor が利用可能なすべての AWS リージョン で使用できます。
- デュアルスタックエンドポイントを使用するには、再生設定、広告決定サーバー設定、CDN 設定を変更する必要はありません。
- CDN が hostname-pattern-basedルーティングを使用している場合は、ルールが .api.awsドメイン形式に対応していることを確認します。
- デュアルスタックエンドポイントの標準 AWS ガイダンスが適用されます。詳細については、AWS 全般のリファレンスの [「デュアルスタックと FIPS エンドポイント」](#) を参照してください。

のクォータ AWS Elemental MediaTailor

MediaTailor のリソースとオペレーションのリクエストは、以下のクォータ (これまで「制限」と呼ばれていたもの) の対象になります。

AWS Service Quotas サービスを使用して、MediaTailor のクォータを表示し、クォータの引き上げをリクエストできます。また、他の多くの AWS サービスも使用できます。詳細については、[Service Quotas ユーザーガイド](#)を参照してください。

広告挿入のクォータ

次の表に、AWS Elemental MediaTailor 広告挿入のクォータを示します。特に明記されていない限り、クォータは調整できません。

名前	デフォルトのクォータ値	説明
広告決定サーバー (ADS) の長さ	25,000	広告決定サーバー (ADS) 仕様の最大文字数。
広告決定サーバー (ADS) のリダイレクト	7	MediaTailor が VAST ラッパータグ内で従うリダイレクトの最大深度。追加のリダイレクトがある場合、MediaTailor は を終了します。
広告決定サーバー (ADS) のタイムアウト	3	MediaTailor が広告決定サーバー (ADS) に対するオープン接続をタイムアウトするまでの最大待機秒数。ADS からのレスポンスがないために接続がタイムアウト

名前	デフォルトのクォータ値	説明	
		する場合、MediaTailor は ad avail を広告で埋めることができません。	
広告挿入リクエスト	10,000	サーバー側の広告挿入を実行するときにパーソナライズされたマニフェストに対して行う 1 秒あたりの最大リクエスト数。広告挿入は、マニフェスト、セッションの初期化、追跡データ、広告セグメントの受信リクエストを処理します。この クォータ は調整可能です。	
Configurations	1,000	MediaTailor が許可する設定の最大数。	
コンテンツオリジンの長さ	512	コンテンツオリジン仕様の最大文字数。	

名前	デフォルトのクォータ値	説明	
コンテンツオリジンサーバーのタイムアウト	2	テンプレートマニフェストのリクエスト時に、MediaTailor がコンテンツオリジンサーバーに対するオープン接続をタイムアウトするまでの最大待機秒数。タイムアウトにより HTTP 504 (GatewayTimeoutException) レスポンスエラーが生成されます。	
マニフェストサイズ	2	オリジン再生マニフェストの最大サイズ。MB 単位。クォータを超過しないことを確実にするため、gzip を使用して入力マニフェストを MediaTailor に圧縮してください。	
パッケージ設定	5	ソースあたりのパッケージ設定の最大数 (ライブまたはビデオオンデマンド)。	

名前	デフォルトのクォータ値	説明	
プリフェッチスケジュール	25	再生設定あたりのアクティブなプリフェッチスケジュールの最大数。この制限に期限切れのプリフェッチスケジュールは含まれません。	
サーバー側のレポートビーコンリクエストのタイムアウト	3 秒	サーバー側のレポート用にビーコンを発射するときにMediaTailor がサーバーへのオープン接続でタイムアウトするまで待機する最大秒数。接続がタイムアウトすると、MediaTailor はビーコンを起動できず、サービスは CloudWatch の MediaTailor/AdDecisionServerInteractions ログにERROR_FIRING_BEACON_FAILED メッセージをログに記録します。	

名前	デフォルトのクォータ値	説明	
PRE_SESSION_INITIALIZATION フックタイムアウト	2 秒	PRE_SESSION_INITIALIZATION フック実行が実行できる最大秒数。すべての関数ステップと HTTP 呼び出しが含まれます。フックがタイムアウトを超えると、MediaTailor はすべての出力を破棄し、関数が実行されなかったかのように続行します。	
PRE_ADS_REQUEST フックタイムアウト	2 秒	PRE_ADS_REQUEST フック実行が実行できる最大秒数。すべての関数ステップと HTTP 呼び出しが含まれます。フックがタイムアウトを超えると、MediaTailor はすべての出力を破棄し、関数が実行されなかったかのように続行します。	
フックあたりの関数の最大数	20	フックごとに許可される関数実行の最大合計数。	

名前	デフォルトのクォータ値	説明	
セッションの期限切れ	マニフェストの継続時間の 10 倍	MediaTailor が非アクティブ状態のセッションを終了するまでの許容最大時間。セッションアクティビティは、プレイヤーのリクエストまたはオリジンサーバーによる事前通知です。セッションの期限が切れると、MediaTailor が HTTP 400 (Bad Request) レスポンスエラーを返します。	

チャネルアセンブリのクォータ

次の表に、AWS Elemental MediaTailor チャネルアセンブリのクォータを示します。特に明記されていない限り、クォータは[調整可能です](#)。

名前	デフォルトのクォータ値	説明	
アカウントあたりのチャネルマニフェストリクエスト	400	アカウントのすべてのチャネルアセンブリチャネルに対する 1 秒あたりのエグレスマニフェストリクエストの最大数。	

名前	デフォルトのクォータ値	説明	
チャンネルごとのチャンネルマニフェストリクエスト	50	1 つのチャンネルアセンブリチャンネルに対する 1 秒あたりのエグレスマニフェストリクエストの最大数。	
チャンネル出力	5	チャンネルあたりの出力の最大数。	
アカウントあたりのチャンネル	100	アカウントあたりのチャンネルの最大数。	
ライブソース	50	ソースロケーションのライブソースの最大数。	
チャンネルあたりのプログラム	400	チャンネルあたりのプログラムの最大数。	
セグメント配信設定	5	ソースロケーションあたりのセグメント配信設定の最大数。	
ソースロケーション	50	アカウントあたりのソースロケーションの最大数。	
VOD ソース	1,000	ソースロケーションのビデオオンデマンド (VOD) ソースの最大数。	

次の表に、AWS Elemental MediaTailor チャンネルアセンブリのスロットリング制限を示します。特に明記されていない限り、クォータは[調整可能です](#)。

名前	1 transactions-per-secondの最大制限	説明	
ConfigureLogsForChannel	1	チャンネルのログを設定します。	
CreateChannel	1	チャンネルを作成します。	
CreateLiveSource	1	ライブソースを作成します。	
CreateProgram	3	プログラムを作成します。	
CreateSourceLocation	1	ソースロケーションを作成します。	
CreateVodSource	1	VOD ソースを作成します。	
DeleteChannel	1	チャンネルを削除します。	
DeleteChannelPolicy	1	チャンネルポリシーを削除します。	
DeleteLiveSource	1	ライブソースを削除します。	
DeleteProgram	3	プログラムを削除します。	
DeleteSourceLocation	1	ソースの場所を削除します。	
DeleteVodSource	1	VOD ソースを削除します。	

名前	1 transactions-per-secondの最大制限	説明	
DescribeChannel	5	チャンネルを記述します。	
DescribeLiveSource	5	ライブソースを記述します。	
DescribeProgram	5	プログラムを記述します。	
DescribeSourceLocation	5	ソースの場所を記述します。	
DescribeVodSource	5	VOD ソースを記述します。	
GetChannelPolicy	5	チャンネルポリシーを取得します。	
GetChannelSchedule	5	チャンネルスケジュールを取得します。	
ListAlerts	5	アラートを一覧表示します。	
ListChannels	5	チャンネルを一覧表示します。	
ListLiveSources	5	ライブソースを一覧表示します。	
ListPrograms	5	プログラムを一覧表示します。	
ListSourceLocations	5	ソースの場所を一覧表示します。	

名前	1 transactions-per-secondの最大制限	説明	
ListTagsForResource	5	リソースのタグを一覧表示します。	
ListVodSources	5	VOD ソースを一覧表示します。	
PutChannelPolicy	3	チャンネルポリシーを配置します。	
StartChannel	1	チャンネルを開始します。	
StopChannel	1	チャンネルを停止します。	
TagResource	1	リソースをタグ付けします。	
UntagResource	1	リソースのタグを削除します。	
UpdateChannel	1	チャンネルを更新します。	
UpdateLiveSource	1	ライブソースを更新します。	
UpdateProgram	1	プログラムを更新します。	
UpdateSourceLocation	1	ソースの場所を更新します。	
UpdateVodSource	1	VOD ソースを更新します。	

MediaTailor のトラブルシューティング

の使用中に再生エラーや同様の問題が発生した場合は AWS Elemental MediaTailor、このセクションのトピックを参照してください。

トピック

- [MediaTailor イベントフローの問題のトラブルシューティング](#)
- [MediaTailor からの再生のトラブルシューティング](#)
- [ビーコン重複排除](#)
- [パラメータのトラブルシューティング](#)
- [MediaTailor 広告スキップのトラブルシューティングガイド](#)

MediaTailor イベントフローの問題のトラブルシューティング

AWS Elemental MediaTailor イベントフローを理解することは、広告挿入の問題をトラブルシューティングするための強力な基盤となります。イベントのシーケンス、タイミング、パターンを分析することで、問題が発生する場所をすばやく特定し、ターゲットを絞ったソリューションを実装できます。

このセクションでは、イベントフロー分析を使用して問題を診断するための実践的なガイダンスを提供します。基本的なイベントフローの概念については、「」を参照してください [広告挿入イベントフロー](#)。

不完全なイベントフローの特定

不完全なイベントフローは、マニフェストのパーソナライゼーション (MediaTailor がパーソナライズされた広告情報をマニフェストに挿入するプロセス) を成功させる前に、予想される一連のイベントが停止したときに発生します。フローの中断箇所を特定すると、広告挿入の失敗の根本原因を特定するのに役立ちます。

不完全な一般的なフローパターン

イベントフローの障害ポイントが異なると、次のような特定のタイプの問題が示されます。

- 広告機会の検出後にフローが停止する: MediaTailor が ADS リクエストを実行できないようにする広告マーカータタはマニフェスト自体に関する問題を示します。ADS 接続、設定、またはタイムアウトの問題は、ADS リクエストの実行後に発生します。

- ADS リクエストの後にフローが停止します。リクエストが行われたときにのみ検出される ADS レスポンスの問題、VAST 解析の問題、クリエイティブ処理の失敗、ADS タイムアウト、接続エラー、無効な ADS URLs などの設定の問題を提案します。
- 追跡ビーコンがない: 追跡設定の問題、サーバー側のレポートの問題、またはクライアント側の実装ギャップを示している可能性があります。

不完全なフロー分析のための CloudWatch クエリ

これらの Amazon CloudWatch Logs Insights クエリを使用して、不完全なイベントフローを特定します。必要な分析のタイプに基づいて、適切なロググループに対してこれらのクエリを実行します。

ロググループの選択:

- MediaTailor/AdDecisionServerInteractions - 広告決定サーバーのインタラクション、広告機会、ADS 関連の障害を分析するクエリに使用します。
- MediaTailor/TranscodeService - トランスコードの問題、クリエイティブ処理の失敗、またはその他の ADS 関連以外の問題が原因で広告が挿入されなかった問題を分析するために使用します。

Example マニフェストのパーソナライゼーションを成功させることなく広告機会を特定する

ロググループ: MediaTailor/AdDecisionServerInteractions

次のクエリは、マニフェストのパーソナライゼーションが成功しなかった広告機会を識別します。

```
fields @timestamp, eventType, avail.availId, sessionId
| filter eventType = "AD_MARKER_FOUND"
| stats count() as total_opportunities by avail.availId
| join (
  fields @timestamp, eventType, avail.availId
  | filter eventType = "FILLED_AVAIL"
  | stats count() as successful_fills by avail.availId
) on avail.availId
| where ispresent(total_opportunities) and not ispresent(successful_fills)
| sort total_opportunities desc
```

Example イベントフローの完了率を分析する

ロググループ: MediaTailor/AdDecisionServerInteractions

次のクエリは、さまざまなイベントタイプの完了率を分析します。

```
fields @timestamp, eventType, avail.availId
| filter eventType in ["AD_MARKER_FOUND", "MAKING_ADS_REQUEST", "VAST_RESPONSE",
  "FILLED_AVAIL", "BEACON_FIRED"]
| stats count() by eventType, avail.availId
| sort avail.availId, eventType
```

Example ビーコンイベントが欠落しているセッションを検索する

ロググループ: MediaTailor/AdDecisionServerInteractions

次のクエリは、avails を満たしているが、対応するビーコンイベントがないセッションを識別します。

```
fields @timestamp, eventType, sessionId, avail.availId
| filter eventType = "FILLED_AVAIL"
| stats count() as filled_avails by sessionId
| join (
  fields @timestamp, eventType, sessionId
  | filter eventType = "BEACON_FIRED"
  | stats count() as beacon_events by sessionId
) on sessionId
| where filled_avails > 0 and (not ispresent(beacon_events) or beacon_events = 0)
| sort filled_avails desc
```

Example トランスコード関連の広告挿入の失敗を特定する

ロググループ: MediaTailor/TranscodeService

次のクエリは、広告挿入の成功を妨げるトランスコードの問題を特定します。

```
fields @timestamp, eventType, sessionId, requestId
| filter eventType in ["TRANSCODE_IN_PROGRESS", "INTERNAL_ERROR", "MISSING_VARIANTS",
  "PROFILE_NOT_FOUND"]
| stats count() as transcode_issues by eventType, sessionId
| sort transcode_issues desc
```

イベントタイミングの問題の分析

イベントタイミング分析は、パフォーマンスのボトルネックを特定し、広告挿入ワークフローを最適化するのに役立ちます。異常なタイミングパターンは、多くの場合、視聴者のエクスペリエンスに影響する根本的な問題を示します。

パフォーマンスタイミングのしきい値

これらのタイミングしきい値を使用して、潜在的なパフォーマンスの問題を特定します。

- 合計フロー時間が 5 秒を超える: ビューワーエクスペリエンスに影響を与える可能性があり、ADS パフォーマンスの問題、オリジンサーバーの問題 (マニフェスト取得タイムアウトなど)、または NAT Gateway、DynamoDB、EC2、またはその他のシステムコンポーネントのインフラストラクチャの問題を含む内部 MediaTailor の問題を示している可能性があります。
- ADS 応答時間が 2 秒を超える場合: ADS パフォーマンスの問題またはネットワークレイテンシーの問題を示します。
- 1 秒を超えるマニフェストパーソナライゼーション: クリエイティブ処理の遅延、オリジンサーバーの問題 (マニフェスト取得タイムアウトなど)、または NAT Gateway、DynamoDB、EC2、またはその他のコンポーネントによるインフラストラクチャの制約を含む内部 MediaTailor システムの問題を示す可能性があります。

タイミング分析クエリ

これらのクエリを使用して、イベントタイミングパターンを分析します。

Example イベントフローの合計期間の測定

次のクエリは、イベントフローの合計期間を測定し、5 秒を超えるイベントフローを識別します。

```
fields @timestamp, eventType, avail.availId
| filter avail.availId = "your-avail-id"
| filter eventType in ["AD_MARKER_FOUND", "FILLED_AVAIL"]
| sort @timestamp asc
| stats min(@timestamp) as start_time, max(@timestamp) as end_time by avail.availId
| eval duration_seconds = (end_time - start_time) / 1000
| where duration_seconds > 5
```

Example ADS レスポンスのタイミングを分析する

次のクエリは、ADS 応答時間を分析し、2 秒を超える応答時間を特定します。

```
fields @timestamp, eventType, avail.availId
| filter avail.availId = "your-avail-id"
| filter eventType in ["MAKING_ADS_REQUEST", "VAST_RESPONSE"]
| sort @timestamp asc
```

```
| stats min(@timestamp) as request_time, max(@timestamp) as response_time by  
avail.availId  
| eval ads_response_seconds = (response_time - request_time) / 1000  
| where ads_response_seconds > 2
```

Exampleスローマニフェストのパーソナライゼーションを特定する

次のクエリは、1 秒以上かかるマニフェストのパーソナライゼーションプロセスを識別します。

```
fields @timestamp, eventType, avail.availId  
| filter avail.availId = "your-avail-id"  
| filter eventType in ["VAST_RESPONSE", "FILLED_AVAIL"]  
| sort @timestamp asc  
| stats min(@timestamp) as response_time, max(@timestamp) as filled_time by  
avail.availId  
| eval personalization_seconds = (filled_time - response_time) / 1000  
| where personalization_seconds > 1
```

一般的なイベントフローの問題と解決策

このセクションでは、頻繁に発生するイベントフローの問題の解決策を、問題のタイプと症状別に整理して提供します。

広告決定サーバーリクエストの失敗

症状: 広告機会の検出後にイベントフローが停止します。ADS リクエストイベントはログに記録されません。

一般的な原因と解決策

- ADS URL 設定エラー: 再生設定の ADS URL が正しくアクセス可能であることを確認します。広告インタラクションログには、ADS リクエストイベント (MAKING_ADS_REQUEST) が表示されますが、対応する VAST レスポンスは表示されません。多くの場合、ERROR_UNKNOWN または同様のエラーイベントが伴います。
- ネットワーク接続の問題: ファイアウォールルールや DNS 解決など、MediaTailor と ADS 間のネットワーク接続を確認します。
- SSL/TLS 証明書の問題: ADS が信頼できる認証機関からの有効な SSL 証明書を使用していることを確認します。特に Google 広告マネージャーの場合は、[AWS サポート](#)に連絡して、Google の SSL 証明書を受け入れる設定フラグを有効にする必要がある場合があります。

診断クエリ

次のクエリは、イベントシーケンスを追跡して ADS リクエストの失敗を診断するのに役立ちます。

```
fields @timestamp, eventType, sessionId
| filter sessionId = "your-session-id"
| filter eventType in ["AD_MARKER_FOUND", "MAKING_ADS_REQUEST", "ERROR_ADS_IO",
"ERROR_UNKNOWN_HOST"]
| sort @timestamp asc
```

広告決定サーバーのレスポンスの失敗

症状: ADS リクエストは成功しましたが、MediaTailor がレスポンスを受信しないか、解析エラーが発生します。

一般的な原因と解決策

- 無効な VAST 形式: VAST 仕様標準に照らして ADS VAST レスポンスを検証します。
- ADS タイムアウトの問題: ADS タイムアウト設定を増やすか、ADS 応答時間を最適化します。
- 空の広告インベントリ: ADS 設定で広告インベントリの可用性とターゲット基準を確認します。

診断クエリ

次のクエリは、リクエストイベントとレスポンスイベントを調べることで、ADS レスポンスの失敗を診断するのに役立ちます。

```
fields @timestamp, eventType, sessionId
| filter sessionId = "your-session-id"
| filter eventType in ["MAKING_ADS_REQUEST", "VAST_RESPONSE", "EMPTY_VAST_RESPONSE",
"ERROR_ADS_RESPONSE_PARSE", "ERROR_ADS_TIMEOUT"]
| sort @timestamp asc
```

マニフェストパーソナライゼーションの失敗

症状: VAST レスポンスは受信されましたが、マニフェストのパーソナライゼーションが失敗するか、広告がスキップされます。

一般的な原因と解決策:

- クリエイティブトランスコーディングの問題: 広告が `NEW_CREATIVE`、挿入前にトランスコーディングする必要があるかどうかを確認します。MediaTailor/TranscodeService ログで

MISSING_VARIANTS, や などのエラーイベントを調べることでINTERNAL_ERROR、トランスコードエラーを確認することもできますPROFILE_NOT_FOUND。

- 期間の不一致の問題: 広告期間が利用可能な広告時間枠期間内に収まることを確認します。
- パーソナライゼーションのしきい値の問題: 再生設定でパーソナライゼーションのしきい値設定を確認します。

診断クエリ

次のクエリは、VAST レスponsと入力済みの表示を調べることで、マニフェストのパーソナライゼーションの失敗を診断するのに役立ちます。

```
fields @timestamp, eventType, sessionId, skippedAds
| filter sessionId = "your-session-id"
| filter eventType in ["VAST_RESPONSE", "FILLED_AVAIL", "WARNING_NO_ADVERTISEMENTS"]
| sort @timestamp asc
```

スキップされた広告理由のクエリ

次のクエリは、広告がスキップされた理由に関する詳細情報を提供します。

```
fields @timestamp, eventType, sessionId, skippedAds.reason, skippedAds.creativeUniqueId
| filter sessionId = "your-session-id"
| filter eventType = "WARNING_NO_ADVERTISEMENTS" or ispresent(skippedAds)
| sort @timestamp asc
```

スキップされた広告理由とクリエイティブな一意の IDs

次のクエリは、各表示の最初の 2 つの広告の理由やクリエイティブな一意の IDs など、スキップされた詳細な広告情報を提供します。

```
fields @timestamp, eventType
| filter sessionId = "your-session-id"
| filter eventType = "FILLED_AVAIL"
| fields avail.skippedAds.0.vastDuration as SkippedDur_Ad0,
  avail.skippedAds.0.skippedReason as Ad0_SkipReason,
  avail.skippedAds.0.creativeUniqueId as SkippedCreative0_UID
| fields avail.skippedAds.1.vastDuration as SkippedDur_Ad1,
  avail.skippedAds.1.skippedReason as Ad1_SkipReason,
  avail.skippedAds.1.creativeUniqueId as SkippedCreative1_UID
```

```
| sort @timestamp desc
```

ビーコンの失敗の追跡

症状: マニフェストのパersonライゼーションは成功しましたが、追跡ビーコンが欠落または失敗しています。

一般的な原因と解決策

- クライアント側の実装の問題: ほとんどの追跡ビーコンの問題は、クライアント側の追跡に十分な頻度で追跡 URLs ポーリングしない、またはプレイヤー固有のビーコン射撃ロジックの問題など、クライアント側の実装の問題に起因します。
- URL アクセシビリティの追跡の問題: VAST レスポンスの追跡 URLs にアクセスできることを確認し、適切なレスポンスを返します。URLs に到達できない場合、または MediaTailor が内部問題を検出してレスポンス配信の追跡が成功しない場合、問題が発生する可能性があります。
- プレイヤーセグメントリクエストの問題: クライアントプレイヤーが実際にセグメントをリクエストしない場合、ビーコンの追跡に見かけの失敗が発生する可能性があります。これにより、ビーコンは送信されません。ビーコンは追跡の失敗として表示されますが、実際にはビーコンの問題ではなくプレイヤーの実装の問題です。

診断クエリ

次のクエリは、満たされた表示とビーコンイベントを調べることで、ビーコン追跡の失敗を診断するのに役立ちます。

```
fields @timestamp, eventType, sessionId
| filter sessionId = "your-session-id"
| filter eventType in ["FILLED_AVAIL", "BEACON_FIRED", "ERROR_FIRING_BEACON_FAILED"]
| sort @timestamp asc
```

イベントフローモニタリングのベストプラクティス

これらのモニタリングプラクティスを実装して、イベントフローの問題を積極的に特定して解決します。

CloudWatch アラームの設定

Amazon CloudWatch アラームを作成して、主要なイベントフローメトリクスをモニタリングします。

- フロー完了率アラーム: 成功したマニフェストのパーソナライゼーションと広告機会の比率が許容可能なしきい値を下回った場合に警告します。
- ADS 応答時間アラーム: 平均 ADS 応答時間をモニタリングし、パフォーマンスしきい値を超えたときに警告します。
- エラー率アラーム: エラーイベントの頻度を追跡し、特定のエラータイプの異常なスパイクを警告します。

定期的なモニタリングクエリ

これらのクエリを定期的に実行して、イベントフローの状態を可視化します。

Example 毎日のイベントフローの成功率

次のクエリは、イベントフローの成功率の毎日の概要をイベントタイプ別に示します。

```
fields @timestamp, eventType
| filter @timestamp > datefloor(@timestamp, 1d)
| stats count() as total_events by eventType
| sort total_events desc
```

Example 時間単位のエラー率の傾向

次のクエリは、エラー率を時間単位で追跡して、傾向のある問題を特定します。

```
fields @timestamp, eventType
| filter eventType like /ERROR_/
| stats count() as error_count by datefloor(@timestamp, 1h) as hour
| sort hour desc
```

パフォーマンス最適化ガイド

イベントフロー分析を使用して広告挿入のパフォーマンスを最適化します。

- ADS 最適化: ADS プロバイダーと協力して応答時間を最適化し、レイテンシーを短縮します。
- クリエイティブ準備: コンテンツプロファイルに合わせて広告クリエイティブを事前にトランスコードし、処理の遅延を軽減します。
- 設定チューニング: イベントフロー分析に基づいて、タイムアウト設定、パーソナライゼーションしきい値、およびその他の設定パラメータを調整します。

その他のトラブルシューティングリソース

イベントフロー分析以外の追加のトラブルシューティングガイダンスについては、以下を参照してください。

- ログ形式の詳細な情報と技術仕様については、「」を参照してください [ログの表示](#)。
- 一般的な広告挿入問題の包括的なトラブルシューティングについては、「」を参照してください [一般的な問題のトラブルシューティング](#)。
- モニタリングとアラートの設定のガイダンスについては、「」を参照してください [Amazon CloudWatch メトリクス AWS Elemental MediaTailor によるモニタリング](#)。
- デバッグログ記録の手順については、「」を参照してください [デバッグログの生成](#)。

MediaTailor からの再生のトラブルシューティング

このセクションでは、プレイヤーソフトウェアのテスト中、およびプレイヤーリクエストの通常の処理中に受け取る可能性がある HTTP エラーコードについて説明します。

Note

また、PutPlaybackConfigurationや などの設定オペレーション中に AWS Elemental MediaTailor API からエラーが表示されることがありますGetPlaybackConfiguration。これらのタイプのエラーの詳細については、[AWS Elemental MediaTailor API リファレンス](#)を参照してください。

プレイヤーが AWS Elemental MediaTailor 直接または CDN を介して リクエストを送信すると、MediaTailor はステータスコードで応答します。MediaTailor がリクエストを正常に処理すると、設定済みのマニフェストと共に、成功を示す HTTP ステータスコード 200 OK を返します。リクエストが失敗した場合、MediaTailor は HTTP ステータスコード、例外名、およびエラーメッセージを返します。

AWS Elemental MediaTailor は 2 つのクラスのエラーを返します。

- クライアントエラー - 通常、不適切にフォーマットされたリクエスト、無効なパラメータ、または不正な URL といったリクエスト自体の問題が原因で発生するエラー。このようなエラーの場合は HTTP 4xx レスポンスコードが表示されます。

- サーバーエラー - 通常、MediaTailor に関する問題、または広告決定サーバー (ADS) やオリジンサーバーといった依存関係のいずれかに関する問題が原因で発生するエラー。このようなエラーの場合は HTTP 5xx レスポンスコードが表示されます。

トピック

- [によって返されるクライアント再生エラー AWS Elemental MediaTailor](#)
- [によって返されるサーバー再生エラー AWS Elemental MediaTailor](#)
- [再生エラーの例](#)

によって返されるクライアント再生エラー AWS Elemental MediaTailor

一般的なガイダンス:

- レスポンスのヘッダーとボディ内にあるほとんどのエラーに関する詳細情報を見つけることができます。
- 一部のエラーについては、設定をチェックする必要があります。再生設定の設定は、 から取得できます AWS Elemental MediaTailor。API の場合、リソースは `GetPlaybackConfiguration/Name` です。詳細については、「[AWS Elemental MediaTailor API リファレンス](#)」を参照してください。

次の表に、 のマニフェスト操作アクティビティによって返されるクライアントエラーコード AWS Elemental MediaTailor、考えられる原因、およびそれらを解決するために実行できるアクションを示します。

コード	例外名	意味	対応方法
40	BadRequestException	MediaTailor は、フォーマットまたはコンテンツに 1 つ、または複数のエラーがあることが原因でリクエストに対応できません。パラメータが不適切にフォーマットされている、または無効な再生設定やセッション ID がリクエストに含まれている可能性があります。	リクエストが適切にフォーマットされており、正確な情報が含まれていることをチェックします。プレイヤーの再生エンドポイント設定が <code>ManifestEndpointPrefix</code> によって返される <code>GetPlaybackConfiguration</code> 設定と一致しているこ

コード	例外名	意味	対応方法
			とを確認します。リクエストを再実行します。
40	AccessDeniedException	リクエストで提供されているホストヘッダーが、MediaTailor の再生 URL で設定されているマニフェストエンドポイントプレフィックスと一致しません。CDN が誤って設定された可能性があります。	CDN 設定をチェックして、MediaTailor に正しいマニフェストエンドポイントプレフィックスを使用していることを確認します。リクエストを再実行します。
40	NotFoundException	MediaTailor が指定された情報を見つけられません。サービス内の何にもマップされていない URL、未定義の設定、利用できないセッションなどの理由が考えられます。	設定とリクエストの有効性を確認してから、セッションを再開してください。
40	ConflictException	プレイヤーが単一のセッションに複数のプレイリストを同時にロードしようとした。その結果、MediaTailor がセッション整合性競合を検出しました。この問題は、HLS プレーヤーで発生します。	プレイヤーがプレイリストを一度に 1 つずつリクエストしていることを確認します。これは HLS 仕様で規定されています。
41	Gone	AWS サポートオペレーターがプレイヤーセッションまたは顧客設定をブロックしました。AWS サポートは、1 つのセッションまたは設定の誤ったトラフィックから発生する非常に大量の 4xx リクエストを検出すると、まれにこれを行います。	リクエストをブロックする必要はないと思われる場合は、 AWS Support までお問い合わせください。適切であれば、状況を調査し、ブロックングフィルターを削除することができます。

追加のサポートが必要な場合は、[AWS Support](#) にお問い合わせください。

によって返されるサーバー再生エラー AWS Elemental MediaTailor

一般的なガイダンス:

- レスポンスのヘッダーとボディ内にあるほとんどのエラーに関する詳細情報を見つけることができます。
- 一部のエラーについては、設定をチェックする必要があります。再生設定の設定は、 から取得できます AWS Elemental MediaTailor。API の場合、リソースは `GetPlaybackConfiguration/Name` です。詳細については、「[AWS Elemental MediaTailor API リファレンス](#)」を参照してください。

次の表に、マニフェスト操作アクティビティによって返されるサーバーエラーコード AWS Elemental MediaTailor、考えられる原因、およびそれらを解決するために実行できるアクションを示します。

コード	例外名	意味	対応方法
50	InternalServerError	処理されない例外	リクエストを再試行します。問題が解決しない場合は、お使いの AWS リージョンの MediaTailor の報告された正常性を確認します https://status.aws.amazon.com/ 。
50	BadGatewayException	オリジンサーバーアドレス、または広告決定サーバー (ADS) アドレスが無効です。無効なアドレスには、プライベート IP アドレスや localhost などがありません。	設定に ADS とオリジンサーバー用の正しい設定があることを確認してから、リクエストを再試行してください。
50	UnsupportedManifestException	オリジンマニフェストが変更されていて MediaTailor がそれをパーソナライズできない、または MediaTailor がオリジンのマニフェスト形式をサポートしません。	これは個々のセッションにのみ影響する可能性があります。セッションを再度開始してください。これは通常、ビューワー内のページを更新することで実

コード	例外名	意味	対応方法
			行できます。問題が解決しない場合は、MediaTailor がオリジンのマニフェスト形式をサポートすることを確認してください。詳細については、「 コンテンツソースの統合 」を参照してください。
50	LoadShed	リクエストへの対応中に、MediaTailor でリソース制限が発生しました。	リクエストを再試行してください。問題が解決しない場合は、お使いの AWS リージョンの MediaTailor の報告された正常性を確認します https://status.aws.amazon.com/ 。
50	ThrottlingException	1 秒あたりのトランザクション数がクォータに到達したため、MediaTailor が使用をスロットリングしています。	リクエストを再試行します。また、で、お使いの AWS リージョンの MediaTailor の報告された正常性を確認することもできます https://status.aws.amazon.com/ 。1 秒あたりのトランザクション数のクォータを引き上げることをお勧めします。詳細については、「 the section called “広告挿入のクォータ” 」を参照してください。

コード	例外名	意味	対応方法
50	GatewayTimeoutException	MediaTailor がオリジンサーバーと通信している間にタイムアウトが発生しました。	リクエストを再試行します。問題が解決しない場合は、オリジンサーバーの状態をチェックし、オリジンサーバーが the section called “広告挿入のクォータ” に記載されているコンテンツオリジンサーバーのタイムアウト内に応答していることを確認します。

追加のサポートが必要な場合は、[AWS Support](#) にお問い合わせください。

再生エラーの例

このセクションでは、コマンドラインの操作に表示される可能性のある再生エラーの例をいくつか示します AWS Elemental MediaTailor。

次の例は、AWS Elemental MediaTailor と広告決定サーバー (ADS) またはオリジンサーバーの間でタイムアウトが発生した場合の結果を示しています。

```
~[ ]> curl -vvv https://111122223333444455556666123456789012.mediatailor.us-west-2.amazonaws.com/v1/master/123456789012/Multiperiod_DASH_Demo/index.mpd
* Trying 54.186.133.224...
* Connected to 111122223333444455556666123456789012.mediatailor.us-west-2.amazonaws.com (11.222.333.444) port 555 (#0)
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: mediatailor.us-west-2.amazonaws.com
* Server certificate: Amazon
* Server certificate: Amazon Root CA 1
* Server certificate: Starfield Services Root Certificate Authority - G2
> GET /v1/master/123456789012/Multiperiod_DASH_Demo/index.mpd HTTP/1.1
> Host: 111122223333444455556666123456789012.mediatailor.us-west-2.amazonaws.com
> User-Agent: curl/7.43.0
> Accept: */*
>
< HTTP/1.1 504 Gateway Timeout
< Date: Thu, 29 Nov 2018 18:43:14 GMT
```

```
< Content-Type: application/json
< Content-Length: 338
< Connection: keep-alive
< x-amzn-RequestId: 123456789012-123456789012
< x-amzn-ErrorType: GatewayTimeoutException:http://internal.amazon.com/coral/
com.amazon.elemental.midas.mms.coral/
<
* Connection #0 to host 111122223333444455556666123456789012.mediatailor.us-
west-2.amazonaws.com left intact
{"message":"failed to generate manifest: Unable to obtain template playlist.
  origin URL:[https://777788889999.mediapackage.us-west-2.amazonaws.com/out/
v1/444455556666111122223333/index.mpd], asset path: [index.mpd], sessionId:
[123456789012123456789012] customerId:[123456789012]}%
```

ビーコン重複排除

このセクションでは、サーバー側の追跡における MediaTailor のビーコン重複排除機能のトラブルシューティングガイドンスを提供します。

症状: ビーコン数が以前に観察された数よりも少ない

原因: ビーコン重複排除により、以前に複数回カウントされた重複イベントが防止されています。

解決策: これは予想される動作であり、より正確な測定を示します。新しいカウントは、テクニカルセグメントリクエストではなく、実際の広告表示イベントを表します。重複排除されたカウントを反映するようにレポートベースラインを更新します。

症状: ビーコン数はプレイヤーの実装によって異なります

原因: プレイヤーによってセグメントリクエストパターンが異なる場合がありますが、重複排除により、これらの違いに関係なく一貫したビーコン数が確保されます。

解決策: これは予想される動作です。重複排除は、さまざまなプレイヤータイプとネットワーク条件にわたって一貫した測定を提供します。

症状: 予想されるビーコン数と実際のビーコン数の不一致

原因: 以前の予想は、ビーコンの重複射撃によるカウントの増加に基づいている可能性があります。

解決策: 履歴データを確認して、重複ビーコンのパターンを特定します。新しい重複排除されたカウントを、測定とレポートの正確なベースラインとして使用します。

パラメータのトラブルシューティング

動的広告変数、マニフェストクエリパラメータ、文字制限、長さ制限、および設定エイリアスに関連する問題のトラブルシューティングについては、[MediaTailor パラメータのトラブルシューティングガイド](#)「」の「」を参照してください[ADS リクエストの MediaTailor 動的広告変数](#)。

MediaTailor 広告スキップのトラブルシューティングガイド

広告スキップは、MediaTailor のお客様が報告する最も一般的な問題の 1 つです。このガイドは、広告スキップの問題を特定し、step-by-stepトラブルシューティング手順を提供するのに役立ちます。

症状と影響

広告スキップが発生すると、次の症状が発生することがあります。

- 予想される広告時間枠中に広告が表示されない
- 広告ブレイクは広告ではなくコンテンツを表示します
- 異なる視聴セッション間での広告再生の不整合
- さまざまなスキップ理由を持つAdSkippedイベントを示す CloudWatch ログ

広告をスキップすると、収益に直接影響し、視聴者のエクスペリエンスが低下する可能性があります。これらの問題に迅速に対処して、最適なパフォーマンスを維持します。

スキップ理由の特定

CloudWatch Logs Insights を使用して、スキップされた広告のMediaTailor/AdDecisionServerInteractionsロググループをクエリします。

```
fields @timestamp, avail.availId, skippedAds.0.skippedReason,
skippedAds.0.creativeUniqueId
| filter eventType = "FILLED_AVAIL" and ispresent(skippedAds.0.skippedReason)
| sort @timestamp desc
```

このクエリは、パターンの特定に役立つ特定の理由を含む最新の広告スキップイベントを返します。

トピック

- [MediaTailor NEW_CREATIVE 広告スキップのトラブルシューティング](#)
- [MediaTailor ADS タイムアウト広告スキップのトラブルシューティング](#)
- [MediaTailor VAST 解析広告スキップのトラブルシューティング](#)
- [MediaTailor 期間不一致広告スキップのトラブルシューティング](#)
- [MediaTailor セッション変数の広告スキップのトラブルシューティング](#)
- [MediaTailor 広告スキップのモニタリングとアラート](#)
- [MediaTailor 広告スキップ防止のベストプラクティス](#)
- [MediaTailor 広告スキップリファレンスガイド](#)

MediaTailor NEW_CREATIVE 広告スキップのトラブルシューティング

広告がNEW_CREATIVE理由とともにスキップされると、は挿入前にトランスコードを必要とする広告 AWS Elemental MediaTailor を検出しました。このトラブルシューティングガイドでは、原因について説明し、step-by-step解決手順を示します。

NEW_CREATIVE スキップの原因

MediaTailor は、クリエイティブ ID、AWS アカウント ID、トランスコードバリエーションセット (基盤となるコンテンツストリームの再生レンディション) の 3 つの主要な要因に基づいて広告をトランスコードします。クリエイティブ ID またはトランスコードバリエーションセットの一部が異なる場合、MediaTailor は広告をトランスコードを必要とする新しいバリエーションとして認識します。

ビットレートの変更

ビットレートが変更されると、最も近い 8,000 ビットに四捨五入した後にビットレートが一致しない場合、NEW_CREATIVE がスキップされる可能性があります。これは通常、以下の状況で発生します。

- 基盤となるコンテンツストリームがプライマリマニフェストを変更する
- 新しいセッションは、既存のセッションとは異なるバリエーションで作成されます。
- コンテンツストリームのビットレートが再生セッション間で一貫性がない

ジョブ帯域幅分散の収集

収集ジョブの場合、MediaTailor は広告をコンテンツストリームに一致させるときに 15% の帯域幅分散しきい値を使用します。ライブストリームの帯域幅と収集された HLS ファイルの帯域幅に 15%

以上の差がある場合、広告は再度トランスコードされます。これにより、トランスコードプロセス中に NEW_CREATIVE がスキップされる可能性があります。

この動作は、一般的に次の場合に発生します。

- ライブストリーミングと VOD コンテンツの両方に同じトランスコード済み広告が使用されます
- 収集されたコンテンツには、元のライブストリームとは異なるエンコードパラメータがあります
- 帯域幅の差がライブコンテンツと収集コンテンツ間の 15% のしきい値を超えている

クリエイティブ ID の競合

MediaTailor が、すでにトランスコードされたメディアファイルの別のクリエイティブ ID を検出すると、次のシーケンスが発生します。

1. 広告は理由 NEW_CREATIVE でスキップされます
2. これにより、不要なトランスコーディングが試行されます。
3. クリエイティブが DUPLICATE_TRANSCODE または COPY_DEDUP としてマークされている

Note

MediaTailor は、トランスコードされた広告を有効期限切れにしたり削除したりしません。これらは MediaTailor 所有の S3 バケットに無期限に保存されます。

解決の手順

NEW_CREATIVE 広告スキップの問題を解決するには:

1. 広告決定サーバーが同じ広告コンテンツに対して一貫したクリエイティブ IDs を返すことを確認します。
2. コンテンツストリームで一貫したビットレートとバリエーションセットが維持されていることを確認します。
3. 再生前に広告がトランスコードされるように、広告プリフェッチの実装を検討してください。詳細については、「[the section called “広告のプリフェッチ”](#)」を参照してください。
4. 永続的な問題については、[AWS サポート](#)に連絡してトラブルシューティングに関するその他のサポートを依頼してください。

NEW_CREATIVE パターンのモニタリング

この CloudWatch Logs Insights クエリを使用して、NEW_CREATIVE 広告スキップのパターンを分析します。

```
fields @timestamp, sessionId, creativeId, skipReason, MediaFileSourceUrl
| filter skipReason = "NEW_CREATIVE"
| stats count() by creativeId, MediaFileSourceUrl
| sort count desc
| limit 50
```

MediaTailor ADS タイムアウト広告スキップのトラブルシューティング

広告が ADS_TIMEOUT または関連する理由でスキップされると、広告決定サーバーとの接続またはパフォーマンスの問題が発生します。広告を正常に挿入するには、ADS との信頼性の高い通信 AWS Elemental MediaTailor が必要です。このトラブルシューティングガイドでは、これらの接続の問題を特定して解決する方法について説明します。

ADS 接続に関する一般的な問題

ADS 接続に関する一般的な問題は次のとおりです。

- MediaTailor からアクセスできない広告決定サーバー
- 設定されたタイムアウト期間内に ADS が応答しない
- ADS がピーク期間中にリクエストボリュームを処理できない
- MediaTailor と ADS 間のネットワーク接続の問題

解決の手順

ADS タイムアウトの問題を解決するには:

1. MediaTailor から広告決定サーバーにアクセスできることを確認します。
2. ADS が設定されたタイムアウト期間内に応答することを確認します。
3. ピーク期間中に ADS がリクエストボリュームを処理できることを確認します。
4. プライマリ ADS が使用できない場合のフォールバック広告戦略の実装を検討してください。

ADS パフォーマンスのモニタリング

AdDecisionServer.Timeouts メトリクスの CloudWatch アラームを設定して、ADS 接続の問題を積極的に検出します。

MediaTailor VAST 解析広告スキップのトラブルシューティング

VAST_PARSING_ERROR または で広告をスキップするとMEDIA_FILE_UNAVAILABLE、VAST レスポンスまたは広告メディアファイルに問題が発生します。広告挿入を成功させるには、適切にフォーマットされた VAST レスポンスとアクセス可能なメディアファイル AWS Elemental MediaTailor が必要です。このトラブルシューティングガイドでは、これらの問題を特定して解決する方法について説明します。

VAST の一般的な問題

VAST の一般的な問題は次のとおりです。

- VAST レスポンス形式が VAST 仕様に準拠していない
- VAST レスポンスのメディアファイル URLsパブリックにアクセスできない
- VAST XML で正しくエンコードされていない特殊文字
- VAST レスポンスに必要なメディアファイルまたは形式がありません

解決の手順

VAST 解析の問題を解決するには:

1. VAST レスポンス形式を VAST 仕様と照合して検証します。
2. VAST レスポンス内のすべてのメディアファイル URLsがパブリックにアクセス可能であることを確認します。
3. VAST XML で特殊文字が適切にエンコードされていることを確認します。
4. VAST レスポンスに MediaTailor と互換性のある形式のメディアファイルが含まれていることを確認します。

VAST ラッパーのトラブルシューティング

INVALID_VAST_WRAPPER_AD または の問題の場合REJECTED_REPLICA_VAST:

- VAST ラッパーレスポンスを VAST 仕様と照合して検証する
- すべてのラッパー要素が適切にフォーマットされ、有効な VASTAdTagURI が含まれていることを確認します。
- 重複コンテンツ検出ポリシーの広告サーバー設定を確認する
- VAST レスポンスに同じセッション内の一意のクリエイティブコンテンツが含まれていることを確認する

MediaTailor 期間不一致広告スキップのトラブルシューティング

AVAIL_DURATION_EXCEEDED、LEFTOVER_AVAIL_EXCEEDED_THRESHOLDまたは形式関連の理由で広告をスキップする場合、期間または形式の互換性の問題があります。には、広告コンテンツと利用可能な広告時間枠間の適切な期間フォーマットとマッチング AWS Elemental MediaTailor が必要です。このトラブルシューティングガイドでは、これらの期間関連の問題を特定して解決する方法について説明します。

一般的な期間の問題

一般的な期間の問題には、次のようなものがあります。

- 利用可能な広告時間枠よりも長い広告
- コンテンツシグナリングの広告時間枠マーカが間違っている
- 広告挿入のパーソナライゼーションしきい値が満たされていない
- EXT-X-CUE-OUT タグの期間形式が正しくない

期間形式の要件

EXT-X-CUE-OUT タグ期間パラメータは、ISO 8601 期間形式ではなく整数値としてフォーマットする必要があります。

期間形式の要件

形式	例	ステータス
整数 (正しい)	32	サポート - 32 秒を表します
10 進数 (正しい)	30.000	サポート - 30 秒を表します

形式	例	ステータス
ISO 8601 (誤)	PT32S	サポートされていない - 挿入が失敗する

解決の手順

期間不一致の問題を解決するには:

1. ADS が利用可能な広告時間枠期間内に収まる広告を返すことを確認します。
2. コンテンツ内の広告ブレイクマーカが意図した期間を正しく通知していることを確認します。
3. EXT-X-CUE-OUT 期間パラメータが整数形式を使用していることを確認します。
4. ユースケースに適した場合は、パーソナライゼーションのしきい値を調整することを検討してください。

MediaTailor セッション変数の広告スキップのトラブルシューティング

セッション変数は、広告のターゲティングと選択において重要な役割を果たします AWS Elemental MediaTailor。セッション変数の設定が正しくないことは、広告スキップの問題の一般的な原因です。この包括的なトラブルシューティングガイドでは、広告挿入の成功を妨げるセッション変数の問題を特定して解決する方法について説明します。

セッション変数の一般的な問題

一般的なセッション変数の問題には、次のようなものがあります。

- 必須変数がない: 広告決定サーバーには、指定されていない特定の変数が必要になる場合があります。
- 変数構文が正しくない: 変数は正しい構文を使用する必要があります (たとえば、`[session.id]`ではなく)`${session.id}`。
- URL エンコードの問題: 変数値の特殊文字には、適切な URL エンコードが必要になる場合があります。
- プレイヤーパラメータの不整合: プレイヤーパラメータはセッション間で一貫して渡す必要があります。
- 動的変数解決の失敗: 解決できない変数は空の文字列に置き換えられます。
- SCTE-35 UPID 解析の問題: セグメンテーション UPID 処理の問題により、セッション変数の解決に失敗する可能性があります。

セッション変数の解決の検証

セッション変数が適切に解決されていることを確認するには:

1. MediaTailor 設定のデバッグログ記録を有効にする
2. 実際の ADS リクエスト URL のMediaTailor/AdDecisionServerInteractionsロググループを確認する URLs
3. テンプレート URL 内のすべての変数が適切な値に置き換えられていることを確認します。
4. 解決の失敗を示している可能性のある空の文字列に置き換えられた変数を探します。

SCTE-35 UPID 解析のトラブルシューティング

SCTE-35 セグメンテーション UPID 処理に関する問題は、セッション変数の問題を引き起こす可能性があります。

- 形式要件: UPID は 12 segmentation_upid_typeで、適切な処理format_identifierのために を含める必要があります
- 解析ルール: デコードされた UPID には、複数の値のコロン区切り文字を含めることができます。テンプレート変数とデコードされた UPID トークンの数は等しくなければなりません
- 無効な形式: 解析に失敗するため、値のない二重コロン (例: : または :46175218::4053) は避けてください。

MediaTailor 広告スキップのモニタリングとアラート

プロアクティブモニタリングは、収益に大きな影響を与える前に広告スキップの問題を検出して解決するのに役立ちます。は、広告挿入パフォーマンスの効果的なモニタリングを可能にする包括的なメトリクスとログ記録機能 AWS Elemental MediaTailor を提供します。このガイドでは、広告スキップの問題に対する効果的なモニタリングを設定する方法について説明します。

モニタリングする主要な CloudWatch メトリクス

これらの主要な MediaTailor メトリクスの CloudWatch アラームを設定します。

- AdDecisionServer.Ads.Skipped - スキップされた広告の数
- AdDecisionServer.Timeouts - ADS タイムアウトの数
- Avail.FilledDuration - 埋められた広告時間枠の期間

- Avail.SlateOnly - スレートのみで満たされた広告時間枠の数

高度な CloudWatch Logs Insights クエリ

詳細なトラブルシューティングには、次の特殊なクエリを使用します。

包括的なセッション分析

特定のセッションの広告挿入動作の詳細な分析:

```
fields @timestamp, sessionId, eventType, creativeId, skipReason, adBreakIndex
| filter sessionId = "your-session-id-here"
| filter eventType in ["FILLED_AVAIL", "SKIPPED_AVAIL", "MAKING_ADS_REQUEST"]
| sort @timestamp asc
| limit 100
```

クリエイティブ IDs の検索

FILLED_AVAIL イベントからクリエイティブ IDs を識別するには:

```
fields @timestamp, sessionId, eventType
| filter sessionId like /sessionId/ and eventType!='BEACON_FIRED'
| sort @timestamp desc
```

Note

を、調査している実際のセッション ID sessionIdに置き換えます。

MediaTailor 広告スキップ防止のベストプラクティス

これらのベストプラクティスを実装することで、広告スキップの問題が発生する前に防止し、広告挿入のパフォーマンスと収益保護を強化できます AWS Elemental MediaTailor。これらのプロアクティブメジャーは、広告スキップの最も一般的な原因に対処し、一貫した広告配信を維持するのに役立ちます。

プロアクティブメジャー

- 広告プリフェッチの実装: MediaTailor のプリフェッチ機能を使用して、再生前に広告がトランスコードされるようにします。実装の詳細については[the section called “広告のプリフェッチ”](#)、「」を参照してください。
- 一貫したクリエイティブ IDs を維持する: 広告決定サーバーがセッション間で同じ広告コンテンツに対して一貫したクリエイティブ IDs を使用していることを確認する
- 適切な期間フォーマットを確認する: ISO 8601 形式の代わりに EXT-X-CUE-OUT 期間パラメータに整数値を使用する
- VOD 最適化を設定する: 複数の広告ブレイクを持つ VOD ストリームに `maxConcurrentAdsRequests` を設定し、サーバーの負荷を軽減します
- ADS パフォーマンスの最適化: 迅速に応答し、ピークトラフィック量処理するように広告決定サーバーを設定する

実装ガイドライン

- 適切なエラー処理を実装する: 広告を挿入できない場合に広告時間枠を埋めるようにスレートコンテンツを設定する
- 徹底的にテストする: さまざまなデバイスやネットワーク条件にわたって広告挿入ワークフローを検証する
- フォールバック戦略の実装: プライマリ広告ソースが失敗した場合のバックアップ広告ソースまたはデフォルト広告を設定する
- トランスコードパターンのモニタリング: CloudWatch ログのトランスコード効率をモニタリングし、トランスコードの問題を示すパターンに気付いた場合は [AWS サポート](#) にお問い合わせください。

MediaTailor 広告スキップリファレンスガイド

このセクションでは、広告スキップの理由に関する包括的なリファレンス情報と、 の関連ドキュメントへのリンクを提供します AWS Elemental MediaTailor。このリファレンスガイドを使用して、さまざまなスキップ理由の具体的な意味を理解し、追加のトラブルシューティングリソースを見つけます。

広告スキップ理由のリファレンスを完了する

MediaTailor は、ロググループからの FILLED_AVAIL イベントログメッセージで広告がスキップされる特定の理由を MediaTailor/AdDecisionServerInteractions ログに記録します。

広告スキップの理由を完了する

理由のスキップ	説明
NEW_CREATIVE	広告はまだトランスコードされていません。これは、MediaTailor が挿入前にトランスコードを必要とする新しい広告クリエイティブに遭遇した場合に発生します。
PROFILE_NOT_FOUND	セッションの設定に関連付けられた MediaConvert トランスコードプロファイルが存在しないため、広告の準備ができません。
TRANSCODE_ERROR	広告トランスコードプロセスでエラーが発生し、完了できませんでした。
TRANSCODE_IN_PROGRESS	広告トランスコードはまだ進行中であり、まだ挿入する準備ができていません。
INTERNAL_ERROR	広告の処理中に内部 MediaTailor エラーが発生し、挿入が妨げられました。
AVAIL_DURATION_EXCEEDED	広告が広告時間枠の残りの期間に収まらない。
LEFTOVER_AVAIL_EXCEEDED_THRESHOLD	挿入された可能性のあるすべての広告の累積期間は、セッション用に設定されたパーソナライゼーションしきい値を満たしていません。
VAST_PARSING_ERROR	広告決定サーバーからの VAST レスポンスにエラーが含まれているか、形式が正しくありません。
ADS_TIMEOUT	広告決定サーバーが設定されたタイムアウト期間内に応答しませんでした。

理由のスキップ	説明
MEDIA_FILE_UNAVAILABLE	VAST レスポンスで指定された広告メディアファイルにアクセスできません。
SESSION_INITIALIZATION_FAILED	MediaTailor セッションは、多くの場合、セッション変数が正しくないため、正しく初期化できませんでした。
EARLY_CUE_IN	広告ブレイクは、早期のキューインシグナルが原因で予想よりも早く終了し、広告が完全に挿入されなくなりました。
NO_VARIANT_MATCH	広告クリエイティブには、コンテンツストリームのエンコードパラメータ (ビットレート、解像度、コーデック) に一致するバリエーションはありません。
NO_MODEL_CREATIVE_MATCH	広告クリエイティブが、現在の再生設定で予想されるモデルまたは形式の要件と一致しません。
REJECTED_REPLICA_VAST	レプリカまたは重複コンテンツ検出ポリシーにより、VAST レスポンスが拒否されました。
INVALID_VAST_WRAPPER_AD	VAST ラッパー広告には、広告挿入の成功を妨げる無効なラッパー要素または不正な形式のラッパー要素が含まれています。
IMPORT_ERROR	広告のインポートプロセス中にエラーが発生しました。広告が挿入のために処理されません。
IMPORT_IN_PROGRESS	広告インポートプロセスは現在進行中であり、まだ完了していません。

関連リソース

広告スキップ問題のトラブルシューティングの詳細については、以下の関連トピックを参照してください。

- [the section called “動的広告変数”](#) - MediaTailor の動的広告変数に関する包括的なガイド
- [the section called “広告のプリフェッチ”](#) - トランスコード関連のスキップを防ぐために広告プリフェッチを実装する方法

- [the section called “CloudWatch メトリクスによるモニタリング”](#) - CloudWatch メトリクスによる MediaTailor のモニタリング
- [the section called “ ログの表示”](#) - MediaTailor ログを表示および分析する方法
- [the section called “イベントフローのトラブルシューティング”](#) - 広告挿入イベントフローを理解する

AWS Elemental MediaTailor リソース

以下の表には、AWS Elemental MediaTailorを利用するのに役立つ関連リソースを示しています。

[リソース]	説明
SCTE 規格: SCTE 35	SCTE35 に関する SCTE 規格のドキュメントです。
クラスとワークショップ	AWS スキルを磨き、実践的な経験を積むのに役立つ、ロールベースのコースや専門コース、セルフペースラボへのリンク。
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開発者ツール、SDKs、IDE ツールキット、コマンドラインツールへのリンク。
AWS ホワイトペーパー	AWS ソリューションアーキテクトや他の技術専門家によって作成された、アーキテクチャ、セキュリティ、経済学などのトピックを網羅した技術 AWS ホワイトペーパーの包括的なリストへのリンク。
AWS サポートセンター	AWS サポートケースを作成および管理するためのハブ。フォーラム、技術的なFAQs、サービスヘルスステータス、AWS Trusted Advisor など、その他の役立つリソースへのリンクも含まれています。
AWS サポート	AWS サポートに関する情報のプライマリウェブページ。クラウドでのアプリケーションの構築と実行に役立つ one-on-one の高速応答サポートチャネルです。
お問い合わせ	AWS 請求、アカウント、イベント、不正使用、その他の問題に関する問い合わせのための一元的な連絡先。

[リソース]	説明
AWS サイト用語	当社の著作権、商標、お客様のアカウント、ライセンス、サイトへのアクセス、およびその他のトピックに関する詳細情報です。

のドキュメント履歴 AWS Elemental MediaTailor

以下の表はこのドキュメントの重要な変更点をまとめたものです。

変更	説明	日付
スキップされた広告理由のアカウントレベルのディメンション	SkippedReason メトリクスにアカウントレベル (ディメンションなし) の排出量を追加しました。	2026 年 7 月 21 日
新しいメトリクスとアカウントレベルのディメンション	Avail.Complete メトリクスを追加しました。、Requests、、AdsBilled、、Avail.Duration、および AdDecisionServer.Ads メトリクスのアカウントレベル (ディメンションなし) Avail.Impression Avail.Complete Avail.FilledDuration の排出量を追加しました。	2026 年 6 月 30 日
Google の広告プラットフォームによるサーバー側の自動認証	MediaTailor が Google 広告マネージャー (GAM)、Google キャンペーンマネージャー (GCM)、Google Display & Video 360 (DV360) との server-to-server 接続を自動的に認証するようになりました。カスタマーアクションやサポートチケットは必要ありません。MediaTailor が適用する URI ベースのルールを説明するサーバー側のセクション	2026 年 4 月 15 日

	を更新し、手動サポートチケットの手順を削除しました。	
TPS トラフィックシェーピングを追加	TPS に基づく ADS へのリクエストの制限に関する情報を含むようにセクションを更新しました。	2025 年 10 月 2 日
「prefetchScheduleName」を追加	ADS ログ prefetchScheduleName に に関する情報を追加しました。	2025 年 9 月 24 日
広告スキップのトラブルシューティングガイドを再構築	実行する 1 つの特定のジョブに各セクションを集中させるためのトラブルシューティングガイドを再構築しました。	2025 年 8 月 26 日
クエリパラメータ処理ドキュメントを追加	MediaTailor が MediaTailor MediaPackage でのタイムシフト表示など、オリジン固有の機能のクエリパラメータを処理する方法を説明するセクションを追加しました。	2025 年 8 月 26 日
SGAI VOD 設定ガイド	video-on-demandコンテンツを使用したサーバーガイド広告挿入の設定ガイダンスを追加しました。	2025 年 8 月 7 日
統合されたパラメータのトラブルシューティングガイド	文字制限エラー、長さ制限の問題、設定エイリアスの問題、パラメータ処理フローのデバッグ、セキュリティのベストプラクティスをカバーする包括的なトラブルシューティングセクションを追加しました。	2025 年 8 月 4 日

[拡張パラメータフォーマットドキュメント](#)

マニフェストクエリパラメータと ADS パラメータの包括的な文字制限、URL エンコードのサポート、長さ制限を追加しました。

2025 年 8 月 1 日

[包括的なリファレンスを含む拡張パラメータドキュメント](#)

AWS Guide 分析に基づく包括的なパラメータ文字制限、長さ制限、URL エンコードサポート、設定エイリアスドキュメント、セッション初期化方法を追加しました。

2025 年 8 月 1 日

[マニフェストクエリパラメータドキュメントの強化](#)

AWS ガイド分析に基づく包括的なパラメータ配置動作、文字制限、URL エンコードのサポート、プロトコル固有の違いを追加しました。

2025 年 8 月 1 日

[拡張ドメイン変数のドキュメント](#)

AWS ガイドの分析に基づく包括的な設定エイリアスドキュメント、サポートされているフィールド、ドメインレベルのパラメータルールを追加しました。

2025 年 8 月 1 日

[イベントフローのトラブルシューティングガイダンスを追加](#)

MediaTailor イベントフロー分析を使用して広告挿入の問題を診断および解決するための実践的なガイダンスを提供する新しいセクション。

2025 年 7 月 31 日

広告挿入イベントフロードキュメントを追加	ユーザーがワークフローを理解するのに役立つ、マニフェストのパーソナライゼーション中の MediaTailor イベントの時系列フローを説明する新しいセクション。	2025 年 7 月 23 日
強化されたクライアント側の追跡ワークフロードキュメント	詳細なシーケンス図と step-by-step のプロセスの説明を含む、包括的なクライアント側のレポートワークフロードキュメントを追加しました。	2025 年 7 月 16 日
CDN 統合ドキュメント	MediaTailor での CDN の設定に関するより包括的な情報を追加しました。	2025 年 6 月 30 日
チャネルアセンブリの CDN 統合を追加	CDN を Channel Assembly と統合してコンテンツ配信を最適化するためのドキュメントを追加しました。	2025 年 5 月 16 日
HLS プレイリストタイプ	HLS マルチバリエーションとメディアプレイリストの定義を追加しました。	2025 年 5 月 9 日
DASH マニフェストタイプ	DASH マニフェストコンポーネントの定義を追加しました。	2025 年 5 月 9 日
HLS プレイリストタイプ情報を追加	例のコンテキストを提供するために、多変量プレイリストとバリエーションプレイリストの説明を追加しました。	2025 年 5 月 9 日

DASH マニフェストタイプ情報を追加	例のコンテキストを提供するために、MPD 構造とコンポーネントの説明を追加しました。	2025 年 5 月 9 日
定期的なプリフェッチを追加	定期的なプリフェッチスケジュールに関する情報を含むようにセクションを更新しました。	2025 年 4 月 28 日
設定ごとのログフィルタリング情報を追加	MediaTailor が設定に対して出力するログのフィルタリングに関する情報を追加しました。	2025 年 3 月 31 日
販売ログ情報を追加	MediaTailor が発行するログの発行に提供されたログの使用に関する新しいセクションを追加しました。	2025 年 2 月 4 日
前提条件付き広告の概要を追加	前提条件付き広告の使用時に MediaTailor が広告挿入を管理する方法を説明する新しいセクションを追加しました。	2025 年 1 月 30 日
広告条件の追加	ストリーミングメディアファイルの条件設定を追加しました。	2025 年 1 月 30 日
SGAI ライブ設定ガイド	ライブコンテンツを使用したサーバーガイド広告挿入の設定ガイダンスを追加しました。	2025 年 1 月 15 日
追加されたログタイプ	MediaTailor が出力するログタイプを一覧表示する新しいセクションを追加しました。	2025 年 1 月 15 日

ログフィルタリング情報を追加	MediaTailor が出力するログのフィルタリングに関する新しいセクションを追加しました。	2025 年 1 月 15 日
サポートされているクエリパラメータのフォーマットを追加	マニフェストクエリパラメータと ADS クエリパラメータの書式設定のセクションを追加しました。	2025 年 1 月 2 日
Google 広告マネージャーの統合情報	MediaTailor と Google 広告マネージャーの統合に関するセクションをサーバー側とクライアント側から追加しました。	2024 年 11 月 25 日
更新された変数	breakabilityStartTime をに変更しましたavailabilityStartTime。	2024 年 5 月 6 日
パーソナライゼーションの詳細を追加	Insertion Mode パーソナライゼーションの詳細を追加しました。	2024 年 5 月 6 日
プログラムルール	プログラムルールに新しいコンテンツを追加しました。	2024 年 4 月 25 日
チャンネルの作成に関するコンテンツを更新しました	チャンネル作成時のプログラムルールに関する情報を追加しました。	2024 年 4 月 20 日
プログラムの追加に関する更新内容	プログラムを追加するときのプログラムルールに関する情報を追加しました。	2024 年 4 月 20 日

ワークフローモニター	AWS メディアサービスを分析し、それらのサービス間にシグナルマップ、メディアワークフローの視覚化を作成します。シグナルマップを使用して、CloudWatch、Event Bridge、および「CloudFormation」を使用して、監視アラームと通知を生成します。	2024 年 4 月 11 日
AlternateMedia と As Run ログ	AlternateMedia が As Run Log にどのように影響するかに関するメモを追加しました。	2024 年 2 月 1 日
マニフェスト設定の更新	選択した広告マークアップタイプに応じて、パススルータグに関する情報を追加しました。	2023 年 11 月 28 日
広告時間枠の SCTE-35 メッセージ	Daterange と Scte35 Enhanced 広告マークアップタイプの挿入された SCTE-35 タグに関する情報を追加しました。	2023 年 11 月 28 日
Enhanced Scte35 広告マークアップタイプのキーと値のペア	MediaTailor が Enhanced Scte35 広告マークアップタイプの送信されたキーと値のペアを処理する方法に関する情報を追加しました。	2023 年 11 月 28 日
VOD ソース広告の機会	MediaTailor は、VOD ソースの広告機会を自動的に検出できるようになりました。	2023 年 10 月 6 日

新しい Autodetect SigV4 認証タイプ	MediaTailor が AUTODETECT_SIGV4 アクセスタイプをサポートするようになりました。	2023 年 8 月 18 日
クライアント側の追跡コンテンツを更新	クライアント側の追跡コンテンツを更新し、追加情報を追加しました。	2023 年 8 月 12 日
ライブソースを使用するように MediaTailor と MediaPackage ワークフローを設定する	MediaTailor と AWS Elemental MediaPackage ワークフローがライブソースを使用する場合のセットアップ、一般的な要件、動作に関する情報を追加しました。	2023 年 5 月 24 日
マニフェストクエリパラメータのドキュメント	マニフェストクエリパラメータを記述するセクションを追加しました。	2023 年 4 月 26 日
オーバーレイ広告ドキュメント	オーバーレイ広告について説明するセクションを追加しました。	2023 年 4 月 24 日
マニフェストドキュメントの広告 ID デコレーション	マニフェストの広告 ID デコレーションについて説明するセクションを追加しました。	2023 年 4 月 24 日
AFTER_LIVE_EDGE 抑制モードを追加	AFTER_LIVE_EDGE 広告抑制モードが BEFORE_LIVE_EDGE モードに加えて利用可能になりました。	2023 年 2 月 21 日
新しい As Run ログ	As Run ログの新しいトピック。	2023 年 1 月 19 日

[IAM ベストプラクティスの更新](#)

IAM ベストプラクティスに沿ってガイドを更新しました。詳細については、「[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。

2022 年 12 月 27 日

[IAM ベストプラクティスの更新](#)

IAM ベストプラクティスに沿ってガイドを更新しました。詳細については、「[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。

2022 年 12 月 27 日

[新しいセグメンテーション UPID プライベートデータ変数](#)

MediaTailor は、コロン区切りのプライベートデータをセグメンテーション UPID タイプ 12 (MPU) から分割 `scte.segmentation_upid.private_data.{index}` する追加のセッション変数をサポートするようになりました。

2022 年 12 月 22 日

[クォータの内容を更新](#)

クォータ情報を更新して再編成しました。

2022 年 9 月 13 日

[チャネルアセンブリクォータの追加と修正](#)

MediaTailor チャネルアセンブリサービスに、ライブソース、セグメント配信設定、マニフェストリクエスト、チャネルトランザクションのクォータを追加しました。

2022 年 9 月 13 日

チャンネルアセンブリクォータの追加と修正	MediaTailor チャンネルアセンブリサービスで、ログ記録、チャンネル、ライブソース、プログラム、ソースの場所、チャンネルポリシーのクォータを追加しました。	2022 年 9 月 11 日
新しいチャンネルアセンブリアラートテーブル	チャンネルアセンブリアラートを説明するテーブルが表示されるようになりました。	2022 年 9 月 1 日
新しい Amazon CloudWatch メトリクス	新しい CloudWatch メトリクスを追加しました。	2022 年 6 月 26 日
広告通話のトピック	VAST レスポンスのサポートをプリフェッチするためのリンクを追加しました。	2022 年 5 月 25 日

[新しい ADS リクエスト変数](#)

MediaTailor は、ADS リクエストでこれらの追加の SCTE-35 `scte.segments_expected` `scte.sub_segments_expected` `scte.device_restrictions` 変数をサポートするようになりました。 `scte.segmentation_type_id` 、 `scte.availables_expected` `scte.delivery_not_restricted_flag` 、 `scte.segment_num` 、 `scte.subsegment_num` 、 `scte.regional_blackout_flag` 、 `scte.archive_allowed_flag` および `scte.segmentation_event_id` 。

2022 年 4 月 27 日

[新しい IAM 管理ポリシーピック](#)

MediaTailor 向けの新しいマネージドポリシーを 2 つ追加しました。

2021 年 11 月 24 日

[新しい AWSElementalMediaTailorReadOnly マネージドポリシー](#)

MediaTailor リソースへの読み取り専用アクセスを許可するアクセス許可を付与する新しい AWS マネージドポリシーを追加しました。

2021 年 11 月 10 日

[新しい AWSElementalMediaTailorFullAccess マネージドポリシー](#)

MediaTailor リソースへのフルアクセスを許可する新しい AWS 管理ポリシーを追加しました。

2021 年 11 月 10 日

新しい混乱した使節トピック	混乱した使節の問題の防止方法を説明するトピックを追加しました。	2021 年 11 月 4 日
広告のプリフェッチトピック	MediaTailor では、広告ブレークの配信前に広告をプリフェッチできるようになりました。	2021 年 10 月 12 日
再生設定のためのロギング設定を追加しました	ロギング設定を使用して、再生設定ログに関連する設定を制御します。	2021 年 9 月 28 日
広告時間枠の SCTE-35 メッセージ	time_signal メッセージとsegmentation_descriptor メッセージの使用に関する情報を追加しました。	2021 年 9 月 1 日
新しいリニア再生モード	新しいリニア再生モードを追加しました。	2021 年 9 月 1 日
新しい絶対的な遷移タイプ	絶対遷移タイプのサポートが追加されました。これを使用して、リニアチャネルでプログラムのウォールクロック開始時間を設定できます。	2021 年 9 月 1 日
新しいチャンネルアセンブリアラートトピック	MediaTailor アラートを使用してチャンネルアセンブリリソースをモニタリングできるようになりました。チャンネルアセンブリリソースで問題または潜在的な問題が発生すると、MediaTailor がアラートを生成します。	2021 年 7 月 14 日

[チャンネル出力リクエストの チャンネルアセンブリクォータ を修正しました](#)

MediaTailor チャンネルアセンブリサービスのチャンネル出力リクエストのクォータを修正しました。

2021 年 6 月 29 日

[新しいソースロケーション認 証タイプ](#)

MediaTailor は、Secrets Manager アクセストークン認証をサポートするようになりました。

2021年6月16日

[新しい階層情報](#)

各階層がサポートするモードとソースタイプに関する情報を追加しました。

2021 年 6 月 13 日

[新しいソースタイプ情報](#)

標準チャンネルの場合、プログラムが再生するソースのタイプに関する情報を追加しました。

2021 年 6 月 13 日

[新しい MediaTailor ライブ ソースドキュメント](#)

ライブソースは、ソースの場所に追加する単一のライブストリームを表します。チャンネルを作成したら、ソースの場所にライブソースを追加し、各ライブソースをプログラムに関連付けることができます。

2021 年 6 月 13 日

[追加の UPID タイプに対する サポート](#)

MediaTailor が、ADS 情報 (0xE)、およびユーザー定義 (0x1) のセグメンテーション UPID タイプをサポートするようになりました。

2021 年 4 月 15 日

[新しいセグメンテーション UPID 動的変数](#)

scte.segmentation_
upid.asse
tId、scte.segm
entation_upid.cueD
ata.key、および
scte.segmentation_
upid.cueData.value
の 3 つの新しい動的変数
があります。これらの変数
は、podbustor ワークフロー
のために MPU セグメンテー
ション UPID タイプ (0xC) と
併せて使用されます。

2021 年 4 月 15 日

[新しいチャンネルアセンブリ サービスの説明](#)

新しいチャンネルアセンブリ
サービスに関する情報を追加
しました。

2021 年 3 月 11 日

[新しい MediaTailor チャンネル アセンブリサービスのドキュ メント](#)

チャンネルアセンブリは、既存
のビデオオンデマンド (VOD)
コンテンツを使用してリニア
ストリーミングチャンネルを作
成することを可能にする新し
いマニフェスト専用のサービ
スです。

2021 年 3 月 11 日

[チャンネルアセンブリクォータ を追加](#)

新しい MediaTailor チャンネル
アセンブリサービスのクォー
タを追加しました。

2021 年 3 月 11 日

[新しいチャンネルアセンブリ 用語](#)

新しいチャンネルアセンブリ
サービスに対応する用語を追
加しました。

2021 年 3 月 10 日

チャンネルアセンブリのタグ付けのサポート	でのチャンネルアセンブリリソースのタグ付けのサポートが追加されました AWS Elemental MediaTailor。Channels、Source Locations、および VodSources がタグ付けをサポートしています。	2021 年 3 月 9 日
新しい動的変数トピック	MediaTailor が動的ドメイン変数をサポートするようになりました。	2021 年 2 月 25 日
オプションの設定エイリアスの設定を追加しました	設定エイリアスをドメイン変数と共に使用して、セッション開始中にドメインを動的に設定します。	2021 年 2 月 25 日
新しいscte.segmentation_upid 動的広告変数	新しい scte.segmentation_upid セッションデータ動的広告変数のサポートを追加しました。	2020 年 12 月 5 日
新しい広告マーカースパスルートピック	HLS マニフェストに広告マーカースパスルーを利用できるようになりました。	2020 年 10 月 29 日
設定のアドバンスト設定を更新しました	広告マーカースパスルーは、再生設定の新しいアドバンスト設定です。	2020 年 10 月 14 日
新しいデバッグログモード	デバッグログモードに関する新しいトピック。	2020 年 8 月 14 日
バンパーに関する EXT-X-CUE-OUT 継続時間属性の明確化	HLS については各 duration タグに EXT-X-CUE-OUT 属性が必要となるように、バンパー要件が更新されました。	2020 年 8 月 5 日

セッション初期化レポート モードパラメータ	セッション初期化リクエストで reportingMode パラメータのサポートが追加されました。セッションの初期化時に MediaTailor がクライアント側またはサーバー側の広告追跡を実行するかどうかを指定できるようになりました。	2020 年 7 月 30 日
新しいバンパートピック	新しいバンパートピックを追加しました	2020 年 7 月 27 日
広告抑制が DASH で利用可能に	広告抑制が DASH で使用できるようになりました。広告抑制トピックから「HLS のみ」の制限を削除しました。	2020 年 6 月 3 日
コンソール固有の名前を更新する	コンソール固有の名前を更新し、新しいバージョンのコンソール UI を反映しました。	2020 年 5 月 1 日
新しいavail.index 動的広告変数	新しい avail.index セッションデータの動的広告変数のサポートを追加しました。	2020 年 3 月 13 日
新しい AdVerifications および Extensions 要素	クライアント側のレポートに対して、AdVerifications および Extensions 要素がサポートされます。	2020 年 3 月 10 日
パーソナライゼーションのしきい値の設定	パーソナライゼーションしきい値のオプション設定のサポートを追加しました。	2020 年 2 月 14 日

DASH VOD マニフェスト	マルチピリオドマニフェスト出力による、オリジンサーバーからのビデオオンデマンド (VOD) DASH マニフェストのサポートを追加しました。	2019 年 12 月 23 日
トランスコードプロファイル名のコンソールサポート	設定にトランスコードプロファイル名の説明を追加しました。	2019 年 12 月 23 日
制限テーブルを更新しました	ADS リダイレクトと ADS タイムアウトの制限が更新されました。	2019 年 12 月 18 日
ライブプレロール動作の文書化	ロール前広告挿入セクションを追加し、ライブプレロール広告が とどのように連携するかについて説明しました AWS Elemental MediaTailor。	2019 年 11 月 26 日
ライブプレロール広告のサポート	ライブストリームの先頭にプレロール広告を挿入するためのサポートを追加しました。	2019 年 9 月 11 日
Amazon CloudWatch Logs Insights での ADS ログの分析	ADS AWS Elemental MediaTailor ログと CloudWatch Logs Insights を使用して MediaTailor セッションを分析するための情報を追加しました。	2019 年 8 月 13 日
セキュリティに関する新しい章	範囲を拡大して標準化するセキュリティ章を追加しました。	2019 年 5 月 23 日

DASH 単一期間マニフェスト	マルチピリオドマニフェスト出力による、オリジンサーバーからのシングルピリオド DASH マニフェストのサポートを追加しました。	2019 年 4 月 4 日
ADS URL での SCTE-35 UPIDs のサポート	広告決定サーバー (ADS) の URL に一意のプログラム ID (UPID) を含めるためのサポートを追加しました。これにより、ADS ではライブラリニアストリーム内でプログラムレベルの広告ターゲティングを提供できます。	2019 年 3 月 28 日
クライアント側のレポートがコンパニオン広告をサポート	クライアント側のレポートでは、AWS Elemental MediaTailor 追跡 URL レスポンスにコンパニオン広告メタデータが含まれるようになりました。	2019 年 3 月 28 日
HLS 広告マーカのドキュメント	サポートされている HLS 広告マーカについて説明するセクションを追加しました。	2019 年 3 月 1 日
タグ付けのサポート	AWS Elemental MediaTailor の設定リソースのタグ付けのサポートを追加しました。タグ付けにより、AWS リソースを特定して整理し、リソースへのアクセスを制御し、AWS コストを追跡できます。	2019 年 2 月 14 日

AWS CloudTrail ログ記録情報を追加	CloudTrail を使用した API でのアクションの AWS Elemental MediaTailor ログ記録に関するトピックを追加しました。	2019 年 2 月 11 日
再生エラーに関するセクションを追加しました	プレイヤーまたはコンテンツ配信ネットワーク (CDN) からのリクエストに応答して、再生中に MediaTailor が返す可能性があるエラーに関する情報を追加しました。	2019 年 2 月 4 日
DASH base64 でエンコードされたバイナリ	<scte35:Signal> <scte35:Binary> マーカー内のマニフェストでスプライス情報を base64 でエンコードしたバイナリで提供するサポートを追加しました。	2019 年 1 月 4 日
DASH タイムシグナル	<scte35:TimeSignal> マーカー内のマニフェストでスプライス情報を提供するサポートを追加しました。	2018 年 12 月 5 日
DASH ロケーションのサポート	MPEG-DASH <Location> タグのサポートを追加しました。	2018 年 12 月 4 日
DASH サポート	MPEG-DASH マニフェストのサポートを追加しました。	2018 年 11 月 14 日
制限テーブルを更新しました	設定とマニフェストサイズの制限が更新されました。	2018 年 10 月 13 日

新規および更新されたメトリクス	広告決定サーバー (ADS) とオリジンのタイムアウトに関するメトリクスを追加し、タイムアウトレスポンスを含むように ADS とオリジンエラーの定義を更新しました。	2018 年 10 月 13 日
サーバー側とクライアント側の広告挿入のユースケースに関するドキュメントカバレッジの向上	サーバー側とクライアント側の広告挿入のための動的広告変数の使用を取り上げた説明と例を充実させました。	2018 年 10 月 1 日
新しいリージョン	PDX および FRA リージョンのサポートを追加しました。	2018 年 7 月 18 日
VAST/VPAID	VAST と VPAID に関する情報を追加しました。	2018 年 3 月 16 日
CloudWatch	使用可能な CloudWatch メトリクス、名前空間、ディメンションに関する情報を追加しました。	2018 年 3 月 16 日
新しいリージョン	アジアパシフィック (シンガポール)、アジアパシフィック (シドニー)、アジアパシフィック (東京) の各リージョンのサポートを追加しました。	2018 年 2 月 8 日
IAM ポリシー情報	固有の IAM ポリシー情報を追加しました AWS Elemental MediaTailor。限定的な許可を持つ管理者以外のロールの作成手順を追加しました。	2018 年 1 月 3 日
最初のリリース	このドキュメントの初回リリースです。	2017 年 11 月 27 日

 Note

- AWS メディアサービスは、ライフセーフティオペレーション、ナビゲーションまたは通信システム、航空交通コントロール、ライフサポートマシンなど、サービスの可用性、中断、または障害が死亡、人身傷害、物的損害、または環境損害につながる可能性のある、フェイルセーフなパフォーマンスを必要とするアプリケーションでの使用を目的として設計または意図されていません。

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。