



User Guide

AWS Elemental MediaTailor



AWS Elemental MediaTailor: User Guide

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

Die Handelsmarken und Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Was ist AWS Elemental MediaTailor?	1
Anforderungen an den Origin-Server	1
MediaTailor Konzepte	2
Typen von HLS-Wiedergabelisten	2
DASH-Manifesttypen	4
Konzepte zum Einfügen von Anzeigen	10
Konzepte zur Kanalmontage	11
Wie funktioniert das Einfügen von Anzeigen	12
Ablauf des Ereignisses beim Einfügen von Anzeigen	14
Zugehörige Services	17
Zugreifen MediaTailor	18
Preisgestaltung	18
Regionen	18
Einrichtung	19
Melden Sie sich an für eine AWS-Konto	19
Erstellen eines Benutzers mit Administratorzugriff	20
Erste Schritte mit MediaTailor	22
Erste Schritte mit der MediaTailor Anzeigenschaltung	22
Voraussetzungen	23
Schritt 1: Zugriff MediaTailor	23
Schritt 2: Bereite einen Stream vor	23
Schritt 3: Konfigurieren Sie die ADS-Anforderungs-URL und die Abfrageparameter	26
Schritt 4: Erstellen einer Konfiguration	27
Schritt 5: Testen Sie die Konfiguration	29
Schritt 6: Senden Sie die Wiedergabeanforderung an MediaTailor	30
(Optional) Schritt 7: Aktivität überwachen	31
Schritt 8: Bereinigen	32
Erste Schritte mit der MediaTailor Kanalmontage	33
Voraussetzungen	34
Schritt 1: Erstellen Sie einen Quellspeicherort	34
Schritt 2: Fügen Sie VOD-Quellen zu Ihrem Quellstandort hinzu	35
Schritt 3: Erstellen Sie einen Kanal	36
Schritt 4: Füge Programme zum Zeitplan deines Kanals hinzu	38

Schritt 5 (optional): Verwenden Sie diese Option MediaTailor , um personalisierte Anzeigen in Ihren Stream einzufügen	40
Schritt 6: Starte deinen Kanal	41
Schritt 7: Teste deinen Kanal	41
Schritt 8: Bereinigen	41
Anzeigen einfügen	42
Unterstützte Audio- und Videocodecs	43
Das Verhalten bei der Insertion von Anzeigen verstehen	43
Verhalten beim Zusammenfügen von Anzeigen bei VOD	44
Verhalten beim Zusammenfügen von Live-Anzeigen	47
Überblick über die serverseitige Anzeigeneinfügung	50
In der Wiedergabekonfiguration aktivieren	50
Erstellen Sie eine servergeführte Sitzung	50
Anzeigenverfolgung mit SGAI	52
Geführter Prefetch mit offensichtlichem Herzschlag	52
Kompatibilität der SGAI-Funktionen	53
SGAI-Live-Konfiguration	56
SGAI VOD-Konfiguration	59
Anforderungen an die Ad-Server-Integration	63
RIESIGE Anforderungen	64
VPAID-Anforderungen	67
Konfigurationsmanagement	68
Konfiguration erstellen	68
Konfigurationsdetails anzeigen	76
Konfigurationseinstellungen bearbeiten	77
Konfigurationen löschen	77
Integration einer Inhaltsquelle	78
Anforderungen an die Eingabequelle	78
Integration einer HLS-Quelle	79
Integration einer MPEG-DASH-Quelle	90
Sicherung der ursprünglichen Interaktionen mit SigV4	110
Integration mit Google Ad Manager	114
Serverseitige Integration	115
Kundenseitige Integration	116
Anpassen des Verhaltens von Werbeunterbrechungen mit der Unterdrückung von Werbeunterbrechungen	117

Konfiguration der Unterdrückung von Werbeunterbrechungen	117
Stoßstange und Einfügung	132
Stoßstangen konfigurieren	133
Einfügung einer Pre-Roll-Anzeige	134
Schiefer und Einfügung	135
Das Slate konfigurieren	136
So funktioniert der Schwellenwert für Personalisierung	137
Slate-Konfiguration und VPAID	141
Anzeigen vorab abrufen	141
So funktioniert der Vorabruf	142
Prefetch-Zeitpläne erstellen	147
TPS-basiertes Traffic-Shaping	154
Prefetch-Zeitpläne werden gelöscht	155
Vorkonditionierte Anzeigen	155
Anforderungen an vorkonditionierte Anzeigen	156
Arbeitsablauf für vorkonditionierte Anzeigen	159
Dynamische Anzeigenvariablen	161
Sitzungsvariablen	162
Spielervariablen	177
Domain-Variablen	180
Konfigurations-Aliase	180
Übergabe von ADS-Parametern	189
Parameter-Routing	192
MediaPackage-Integration	194
Verhalten der Sitzung	196
Parameterreferenz	198
Fehlerbehebung bei Parametern	199
Fehlerbehebung bei Alias	204
MediaTailor manifeste Abfrageparameter	208
Origin-Abfrageparameter	208
MediaTailorZeichenbeschränkungen	209
MediaTailorImplizite HLS-Sitzungen	211
MediaTailor Implizite DASH-Sitzungen	212
MediaTailor explizite Initialisierung der Sitzung	214
MediaTailorprotokollspezifisches Verhalten	216
MediaTailor CDN-Integration	219

Daten melden und verfolgen	220
Serverseitiges Tracking	221
Kundenseitiges Tracking	238
Overlay-Anzeigen	317
Voraussetzungen für die Verwendung von Overlay-Anzeigen	318
Erste Schritte	319
Protokollierung und Metriken	332
Abrechnung für Overlay-Anzeigen in MediaTailor	334
ID-Dekoration hinzufügen	335
Status der Sitzung	336
Manifeste und Einfügen von Anzeigenmetadaten	338
Interaktionen mit dem Ad Decision Server (ADS)	362
Client-seitige Tracking-API	363
Linear zusammengestellte Streams erstellen	365
Mit Quellstandorten arbeiten	365
Einen Quellspeicherort erstellen	366
Konfiguration der Authentifizierung für Ihren Quellstandort	368
Mit VOD-Quellen arbeiten	378
Arbeiten mit Live-Quellen	382
Verwenden von Paketkonfigurationen	387
Zwischenspeichern von Manifesten	387
Mit Kanälen arbeiten	388
Erstellen Sie einen Kanal	388
Verwende Quellgruppen mit den Ausgaben deines Kanals	391
Löscht einen Kanal	391
Ein Programm hinzufügen	392
Ein Programm erstellen	392
Definition von Zielgruppenkohorten und alternativen Inhalten	401
Generierung zielgruppenspezifischer Manifeste	404
Fügen Sie Anzeigen und Werbeunterbrechungen ein	405
Anzeigeneinfügung einrichten	405
SCTE-35-Nachrichten für Werbeunterbrechungen	406
Zeitversetzung bei der Wiedergabe eines Kanals	413
Verwenden eines CDN	415
CDN-Auswahl	416
Planen Sie die CDN-Integration	418

Schätzen Sie die Verkehrsanforderungen	419
Konfigurieren Sie Optimierungsstrategien	421
Passen Sie die Planung an	422
Richten Sie Überwachung und Skalierung ein	425
Optimieren Sie die Kosten	427
Testen Sie Ihre Implementierung	428
Beheben Sie häufig auftretende Probleme	430
Richten Sie die CDN-Integration ein	432
Komponenten und Anforderungen für die CDN-Integration	433
Voraussetzungen für die CDN-Integration	23
Schritte zur Einrichtung der CDN-Integration	434
Erforderliche Header für MediaTailor die CDN-Integration	435
Richten Sie das CDN-Routing-Verhalten ein	437
Richten Sie die CDN-Zuordnung ein	441
Bewährte Methoden für die Gewährleistung der Sicherheit	444
Einfügung von Anzeigen mit CDN	446
Was du benötigst	446
Bevor Sie beginnen	447
Vorteile der CDN-Integration	447
Verstehen Sie die CDN-Architektur	449
Richten Sie die grundlegende Anzeigeneinfügung ein	455
SSAI mit Kanalmontage	459
Optimieren Sie die CDN-Leistung	467
Überwachen Sie den CDN-Betrieb	469
Beheben Sie Probleme beim Einfügen von Anzeigen mit CDNs	475
Kanalmontage mit CDN	480
Was Sie benötigen	480
Bevor Sie beginnen	481
Vorteile der CDN-Integration	482
Verstehen Sie die CDN-Architektur	483
Grundlegende Einrichtung	485
Basis konfigurieren URLs	486
Implementieren Sie das Einfügen von Anzeigen	487
Konfigurieren Sie die zeitversetzte Anzeige	490
Überwachen Sie den CDN-Betrieb	492
Vollständiger Leitfaden zur Optimierung	492

MediaPackage CDN-Integration	493
Grundlegendes zum MediaPackage und CDN-Workflow	493
Voraussetzungen	494
Konfiguration der grundlegenden Einstellungen	494
Überprüfung der Integration	497
Nächste Schritte	498
Manifestfilterung	499
CDN-Caching	505
CDN-Leistung	510
Fehlerbehebung bei der CDN-Integration	516
CloudFront-Integration	520
Grundlegende CloudFront Einrichtung	521
Leistungsoptimierung	528
Resilienz in mehreren Regionen	532
Überwachung und Problembehebung	536
CDN-Setup eines Drittanbieters	541
Voraussetzungen	542
CDN-Caching-Regeln konfigurieren	543
Überprüfen Sie Ihre CDN-Konfiguration	546
Optimieren Sie die CDN-Leistung	547
Beheben Sie CDN-Probleme von Drittanbietern	549
CDN-Leistungsoptimierung	550
Caching-Optimierung	551
Routing-Optimierung anfordern	556
Leistungsbenchmarks	556
Fortgeschrittene Optimierung	557
CDN-Überwachung	560
Wichtige CDN-Leistungskennzahlen	561
Richten Sie Überwachungstools ein	563
Checkliste für die Implementierung	565
Verwandte Themen	566
Testen der CDN-Integration	566
Voraussetzungen für das Testen	567
Systematische Testmethodik	567
Checkliste vor der Bereitstellung	572
Referenz zu Testtools	573

Beheben Sie Probleme mit der CDN-Integration	574
Schnelle Diagnose-Checkliste	575
Integrationstests	577
Manifeste 404-Fehler	581
Probleme mit der Manifestzustellung	582
Probleme bei der Segmentzustellung	584
Probleme mit der Sitzungsverwaltung	586
Probleme beim Timing von Werbeunterbrechungen	589
Leistungs- und Latenzprobleme	591
Inkonsistentes Geräteverhalten	598
Vorbereitung der Fehlerbehebung	599
Workflow-spezifische Anleitungen zur Fehlerbehebung	601
Referenz zur Analyse des CDN-Integrationsprotokolls	603
Interpretation von CDN-Protokollen	603
Referenz zu Fehlercodes	606
Tools zur Protokollanalyse	608
MediaTailor Protokollierung von Konfigurationsressourcen	608
CloudFormation -Automatisierung	610
Warum verwenden AWS CloudFormation	611
Bereiten Sie sich auf die Bereitstellung vor	612
Stellen Sie die Vorlage bereit	615
Verwenden Sie die bereitgestellten Ressourcen	616
Testen und validieren Sie Ihre Bereitstellung	619
Fehlersuche bei Problemen mit der -Bereitstellung	620
Passen Sie die Vorlage an	622
Vorlage – Referenz	625
Produktionskonfiguration CloudFront	631
Architektur mit drei Ursprüngen	631
Rangfolge und Konfiguration des Cache-Verhaltens	633
Holen Sie sich Unterstützung bei der CDN-Integration	640
Wann Sie an den Support eskalieren sollten	641
Sammeln Sie Informationen für den Support	641
Ressourcen für den Support	642
Bewährte Methoden für Supportfälle	643
Sicherheit	644
Datenschutz	645

Datenverschlüsselung	646
Identitäts- und Zugriffsverwaltung	646
Zielgruppe	647
Authentifizierung mit Identitäten	647
Verwalten des Zugriffs mit Richtlinien	649
Wie AWS Elemental MediaTailor funktioniert mit IAM	650
Beispiele für identitätsbasierte Richtlinien	656
Beispiele für eine ressourcenbasierte Richtlinie	659
AWS verwaltete Richtlinien	659
Verwenden von servicegebundenen Rollen	661
Fehlerbehebung für -Identität und -Zugriff	665
Compliance-Validierung	668
Ausfallsicherheit	668
Sicherheit der Infrastruktur	669
Serviceübergreifende Confused-Deputy-Prävention	669
Protokollierung und Überwachung	670
CloudWatch Alarme	670
CloudTrail Logs	671
AWS Trusted Advisor	671
Überwachung und Tagging	672
Anzeigen von Protokollen	673
ADS-Protokolle	673
Manifest-Protokolle	725
Protokolle transkodieren	730
Verwendung von verkauften Protokollen	734
Protokolle in CloudWatch Logs schreiben	738
Steuerung des Volumens der Sitzungsprotokolle zur Anzeigeneinfügung	748
Filterung von Protokollen und Ereignissen	751
Debug-Protokolle generieren	754
Überwachung mit CloudWatch Metriken	760
AWS Elemental MediaTailor CloudWatch Metriken	760
AWS Elemental MediaTailor CloudWatch Abmessungen	772
Verwendung von Metriken zur Diagnose veralteter Manifeste	773
API-Aufrufe aufzeichnen	775
AWS Elemental MediaTailor Informationen in CloudTrail	776
Grundlegendes zu AWS Elemental MediaTailor Einträgen in Protokolldateien	777

Empfangen von Channel Assembly-Benachrichtigungen	779
Behebung von Warnmeldungen im Zusammenhang mit Manifesten	785
Benachrichtigungen anzeigen	785
Umgang mit Warnmeldungen	787
Taggen von Ressourcen	787
Unterstützte Ressourcen	787
Tag-Einschränkungen	788
Verwaltung von Tags	788
Workflow-Monitor	788
Komponenten des Workflow-Monitors	791
Unterstützte Services	791
Workflow-Monitor konfigurieren	792
Verwenden Sie den Workflow-Monitor	812
Kontingente	815
Kontingente für das Einfügen von Anzeigen	815
Kontingente für die Kanalmontage	820
Fehlerbehebung	826
Fehlerbehebung beim Ereignisablauf	826
Unvollständige Abläufe	826
Analyse des Zeitpunkts	829
Allgemeine Probleme	830
Bewährte Verfahren zur Überwachung	834
Weitere Ressourcen	835
Fehlerbehebung bei der Wiedergabe	836
Client-Fehler	837
Serverfehler	839
Beispiele	842
Beacon-Deduplizierung	843
Fehlerbehebung bei Parametern	843
Fehlerbehebung beim Überspringen von Anzeigen	844
Symptome	844
Gründe überspringen	844
NEW_CREATIVE beim Überspringen	845
ADS-Timeout beim Überspringen	847
VAST-Parsen und Überspringen	848
Überspringen der Dauer bei Nichtübereinstimmung	849

Überspringen von Sitzungsvariablen	850
Überwachung und Überspringen	852
Bewährte Verfahren zur Prävention	853
Referenzmaterialien	854
MediaTailor Ressourcen	857
Dokumentverlauf	859
.....	dcccclxxxii

Was ist AWS Elemental MediaTailor?

AWS Elemental MediaTailor ist ein skalierbarer Service zum Einfügen von Anzeigen und zur Kanalmontage, der in der AWS Cloud ausgeführt wird. Mit MediaTailor können Sie Zuschauern gezielte Werbeinhalte bereitstellen und lineare Streams erstellen, während die Übertragungsqualität in over-the-top (OTT-) Videoanwendungen erhalten bleibt. MediaTailor Die Anzeigeneinfügung unterstützt Apple HTTP Live Streaming (HLS) und MPEG Dynamic Adaptive Streaming over HTTP (DASH) für Video-on-Demand (VOD) und Live-Workflows.

AWS Elemental MediaTailor Das Einfügen von Werbeanzeigen bietet wichtige Fortschritte gegenüber herkömmlichen Ad-Tracking-Systemen: Anzeigen lassen sich besser monetarisieren, haben eine konsistentere Videoqualität und Auflösung und sind in plattformübergreifenden Umgebungen einfacher zu verwalten. MediaTailor vereinfacht Ihren Werbe-Workflow, indem es allen Geräten mit IP-Verbindung ermöglicht wird, Anzeigen auf die gleiche Weise zu rendern wie andere Inhalte. Der Service bietet auch erweiterte Nachverfolgung von Werbe-Aufrufen, womit sich die Monetisierung von Inhalten weiter erhöht.

AWS Elemental MediaTailor Channel Assembly ist ein reiner Dienst, mit dem Sie lineare Streaming-Kanäle mithilfe Ihrer vorhandenen Video-on-Demand-Inhalte (VOD) erstellen können. MediaTailor berührt niemals Ihre Inhaltssegmente, die direkt von Ihrem Ursprungsserver bereitgestellt werden. MediaTailor Ruft stattdessen die Manifeste von Ihrem Ursprung ab und verwendet sie, um ein Live-Slide-Manifestfenster zusammenzustellen, das auf die zugrunde liegenden Inhaltssegmente verweist.

MediaTailor Die Kanalzusammenstellung macht es einfach, deinen Kanal zu monetarisieren, indem du Werbeunterbrechungen in deinen Stream einfügst, ohne dass du ihn mit SCTE-35-Markern versehen musst. Du kannst Channel Assembly mit MediaTailor Anzeigeneinfügung oder einen anderen serverseitigen Dienst zur Anzeigeneinfügung verwenden.

Anforderungen an den Origin-Server

AWS Elemental MediaTailor hat spezielle Anforderungen an die Kommunikation mit dem Ursprungsserver:

- Unterstützte Ports — akzeptiert MediaTailor nur Ursprünge, die Standard-HTTP- und HTTPS-Ports verwenden:
 - Port 80 für HTTP-Verbindungen
 - Port 443 für HTTPS-Verbindungen

MediaTailor unterstützt keine benutzerdefinierten Ports für die Kommunikation mit dem Ursprungsserver.

- Protokollanforderungen — Für eine sichere Kommunikation MediaTailor ist HTTPS für bestimmte Herkunftstypen und Authentifizierungsszenarien erforderlich. Weitere Informationen finden Sie unter [the section called “Integration einer Inhaltsquelle”](#).

MediaTailor Konzepte

Im Folgenden finden Sie eine Übersicht über die Konzepte, die im gesamten AWS Elemental MediaTailor Benutzerhandbuch verwendet werden.

Typen von HLS-Wiedergabelisten

HTTP Live Streaming (HLS) verwendet zwei Haupttypen von Playlisten: multivariante Playlisten und Medien-Playlisten. Für die Konfiguration und Problembehebung von Workflows ist es wichtig, den Unterschied zwischen diesen Playlist-Typen zu verstehen. MediaTailor

Multivariante Wiedergabeliste

Eine multivariante Playlist ist die Indexdatei der obersten Ebene in einem HLS-Streaming-Workflow, in der alle verfügbaren Wiedergabeversionen des Inhalts aufgeführt sind. Sie enthält Verweise auf Medien-Playlisten, enthält jedoch selbst keine Mediensegmente. Die multivariante Playlist ermöglicht es Spielern, je nach Netzwerkbedingungen, Gerätefunktionen oder Benutzereinstellungen die am besten geeignete Wiedergabeversion auszuwählen.

Dieser Playlist-Typ ist in verschiedenen Kontexten auch unter mehreren anderen Namen bekannt:

- Master-Playlist (veralteter Begriff)
- Master-Manifest (gebräuchlicher Begriff in der Branche)
- Primäre Playlist
- Haupt-Playlist
- Indexdatei
- Meister M3U8

In MediaTailor Workflows ist die multivariante Playlist der Einstiegspunkt für Wiedergabeanfragen und der Ausgangspunkt für die Personalisierung von Anzeigen.

Example Beispiel für eine multivariante Playlist

```
#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
```

Medien-Playlist

Eine Medien-Playlist enthält die aktuellen Mediensegmentinformationen für eine bestimmte Wiedergabe (Qualitätsstufe) des Inhalts. Sie enthält Informationen zum Timing, zu Segmenten und anderen Metadaten URLs, die für die Wiedergabe einer einzelnen Wiedergabe erforderlich sind. Jede Medien-Playlist steht für eine andere Kombination aus Auflösung, Bitrate und anderen Merkmalen.

Dieser Playlist-Typ ist auch unter mehreren anderen Namen bekannt:

- Varianten-Playlist
- Manifest für Kinder (gebräuchlicher Begriff in der Branche)
- Chunklist
- Medien M3U8
- Wiedergabe-Wiedergabeliste

In MediaTailor Workflows werden Medien-Playlisten so personalisiert, dass sie sowohl Inhaltssegmente als auch Anzeigensegmente in der richtigen Reihenfolge enthalten.

Example Beispiel für eine Medien-Playlist

```
#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

Sowohl multivariante Playlisten als auch Medien-Playlisten sind textbasierte Dateien, die die Erweiterung verwenden. .m3u8 Die Terminologie für diese Dateien variiert je nach Dokumentation und Kontext, aber die grundlegenden Rollen in der HLS-Streaming-Architektur bleiben dieselben.

Bei der Konfiguration MediaTailor geben Sie in der Regel die URL zur multivarianten Playlist in Ihrem Inhaltsursprung an. MediaTailor kümmert sich dann um die Personalisierung sowohl der multivarianten Playlist als auch der Medien-Playlisten, um Anzeigen entsprechend Ihrer Konfiguration einzufügen.

Weitere Informationen zu den Spezifikationen der HLS-Wiedergabeliste finden Sie im [HTTP Live Streaming-RFC](#).

DASH-Manifesttypen

Dynamic Adaptive Streaming over HTTP (DASH) verwendet ein MPD-Manifest (Media Presentation Description), um Streaming-Inhalte bereitzustellen. Das Verständnis der Struktur und der Komponenten von DASH-Manifesten ist für die Konfiguration und Fehlerbehebung von MediaTailor Workflows unerlässlich.

MPD (Beschreibung der Medienpräsentation)

Die MPD ist die primäre Manifestdatei beim DASH-Streaming, die die Struktur und Verfügbarkeit von Medieninhalten beschreibt. Sie enthält Informationen über Zeiträume, Adaptionssätze, Repräsentationen und Segmente, aus denen sich die Streaming-Inhalte zusammensetzen.

Dieser Manifesttyp ist in verschiedenen Kontexten auch unter mehreren anderen Namen bekannt:

- DASH-Manifest
- DASH-MODUS
- Master-Manifest (beim Vergleich mit HLS)
- Manifest der Präsentation

In MediaTailor Workflows ist das MPD der Einstiegspunkt für Wiedergabeanfragen und der Ausgangspunkt für die Personalisierung von Anzeigen.

Example Beispiel für ein MPD-Manifest

```
<?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>
  </Period>
</MPD>
```

```

<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>

```

Zeitraum

Eine Periode ist ein zeitlicher Abschnitt einer DASH-Präsentation. Jede Periode enthält ein oder mehrere Adaptionssets und steht für einen Zeitraum der Medienzeit. In Workflows zum Einfügen von Anzeigen werden in der Regel separate Perioden verwendet, um zwischen Inhalten und Anzeigen zu unterscheiden.

Diese Komponente ist auch unter mehreren anderen Namen bekannt:

- Inhaltssegment
- Abschnitt Zeitleiste
- Programmsegment

In MediaTailor Workflows werden Perioden verwendet, um den Hauptinhalt vom Anzeigeninhalt zu trennen, wobei jede Anzeige in der Regel durch einen eigenen Zeitraum repräsentiert wird.

Example Beispiel für einen Zeitraum

```

<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

Ein AdaptationSet gruppiert einen Satz austauschbarer codierter Versionen einer oder mehrerer Medieninhaltskomponenten. Eine AdaptationSet kann beispielsweise mehrere Videoqualitätsstufen enthalten, während eine andere mehrere Audiosprachenoptionen enthalten kann.

Diese Komponente ist auch bekannt als:

- Komponentengruppe „Medien“
- Stream-Set
- Gruppe verfolgen

In MediaTailor Workflows AdaptationSets werden sie beim Einfügen von Anzeigen beibehalten, um konsistente Medientypen zwischen Inhalten und Anzeigen zu gewährleisten.

Darstellung

Eine Darstellung ist eine bestimmte kodierte Version des Medieninhalts innerhalb eines AdaptationSet. Jede Darstellung unterscheidet sich in der Regel in der Bitrate, Auflösung oder anderen Kodierungsparametern, sodass Kunden je nach Netzwerkbedingungen und Gerätefunktionen die am besten geeignete Version auswählen können.

Diese Komponente ist auch bekannt als:

- Wiedergabe (ähnlich wie HLS)
- Qualitätsniveau
- Bitrate-Variante
- Stream-Variante

In MediaTailor Workflows werden Repräsentationen in Anzeigenperioden so genau wie möglich mit den Darstellungen in Inhaltsperioden abgeglichen, um ein reibungsloses Seherlebnis zu gewährleisten.

Segment

Ein Segment ist eine Einheit von Mediendaten, auf die individuell über eine URL verwiesen werden kann. Segmente enthalten den eigentlichen Medieninhalt (Video, Audio usw.) und werden in der MPD referenziert. In DASH gibt es zwei Haupttypen von Segmenten:

- Initialisierungssegment: Enthält Initialisierungsinformationen für eine Repräsentation, wie Codec-Parameter und Timing-Informationen.

- **Mediensegment:** Enthält die tatsächlichen Mediendaten für einen bestimmten Zeitraum innerhalb einer Repräsentation.

In MediaTailor Workflows URLs werden Segmente häufig so geändert, dass sie auf die entsprechenden Inhalte oder Werbemediendateien verweisen.

Note

DASH-Manifeste sind XML-basierte Dateien, die in der Regel die `.mpd` Erweiterung verwenden. Die Terminologie für diese Dateien und ihre Komponenten kann je nach Dokumentation und Kontext variieren, aber die grundlegende Struktur der DASH-Streaming-Architektur bleibt dieselbe.

Bei der Konfiguration MediaTailor geben Sie die URL zum MPD-Manifest in Ihrem Inhaltsursprung an. MediaTailor kümmert sich dann um die Personalisierung des Manifests, in der Regel durch das Einfügen zusätzlicher Perioden für Anzeigen entsprechend Ihrer Konfiguration.

Weitere Informationen zu den Spezifikationen des DASH-Manifests finden Sie in den [Implementierungsrichtlinien des DASH Industry Forum](#).

Fortgeschrittene DASH-Manifest-Konzepte

Wenn Sie mit DASH-Manifesten in arbeiten MediaTailor, kann Ihnen das Verständnis der folgenden fortgeschrittenen Konzepte dabei helfen, Ihre Streaming-Workflows effektiver zu konfigurieren und Fehler zu beheben:

Berechnung der Startzeit des Zeitraums

In DASH-Manifesten werden die Startzeiten der Periode auf der Grundlage des `availabilityStartTime` MPD-Attributs und des `start` Periodenattributs berechnet. Bei Live-Streams ist die Startzeit relativ zur Startzeit der Verfügbarkeit, während sie bei VOD-Inhalten in der Regel relativ zum Beginn der Präsentation ist.

Beim MediaTailor Einfügen von Werbezeiträumen werden die Startzeiten sorgfältig berechnet, um einen reibungslosen Übergang zwischen Inhalten und Anzeigen zu gewährleisten. Bei dieser Berechnung wird Folgendes berücksichtigt:

- Die Startzeit der ursprünglichen Periode

- Die Dauer der vorangegangenen Anzeigenperioden
- Beliebiger Zeitversatz, der in der Antwort des Ad Decision Servers angegeben wurde

Timing-Berechnungen vor dem Roll-Timing

Preroll-Anzeigen in DASH-Manifesten erfordern eine besondere Behandlung, da sie erscheinen, bevor der Hauptinhalt beginnt. MediaTailor fügt Preroll-Anzeigen als separate Perioden zu Beginn des Manifests mit entsprechenden Startzeiten und Dauern ein.

Für Preroll-Anzeigen gilt: MediaTailor

- Erstellt einen neuen Zeitraum für jede Preroll-Anzeige
- Setzt die Startzeit der ersten Preroll-Periode auf 0
- Passt die Startzeit des Hauptinhaltszeitraums an, um die Gesamtdauer aller Preroll-Anzeigen zu berücksichtigen

Live-to-VOD Übergänge

DASH-Manifeste können vom Live-Format zum VOD-Format (Video on Demand) wechseln, wenn ein Live-Stream endet. Dieser Übergang beinhaltet das Ändern des type MPD-Attributs von `dynamic` bis `static` und das Anpassen anderer Attribute wie `timeShiftBufferDepth` und `minimumUpdatePeriod`

Wenn bei live-to-VOD Übergängen MediaTailor Prozesse auftreten, wird sichergestellt, dass:

- Werbemarkierungen bleiben im VOD-Manifest erhalten
- Die Startzeiten der Perioden werden angepasst, um das richtige Timing zu gewährleisten
- Das Manifest ist weiterhin mit VOD-Wiedergabeclients kompatibel

DRM-Handhabung

Informationen zur Verwaltung digitaler Rechte (DRM) in DASH-Manifesten sind in der Regel in den `ContentProtection` Elementen innerhalb von `Representations` `AdaptationSets` enthalten. MediaTailor behält diese Elemente beim Einfügen von Anzeigen bei, um sicherzustellen, dass der Inhaltsschutz gewahrt bleibt.

Bei der Verarbeitung von DRM-geschützten Inhalten gilt Folgendes: MediaTailor

- Behält alle `ContentProtection` Elemente aus dem ursprünglichen Manifest bei
- Stellt sicher, dass Anzeigeninhalte gegebenenfalls kompatible DRM-Schemata verwenden
- Behält alle DRM-bezogenen Attribute und Elemente im gesamten Manifest bei

SCC-Flags

SCC-Flags (Supplemental Content Control) in DASH-Manifesten bieten zusätzliche Informationen zu Inhaltsmerkmalen und Wiedergabeanforderungen. Diese Flags sind in der Regel als Attribute oder Elemente in der MPD-Struktur enthalten.

Zu den gängigen SCC-Flags, die zu MediaTailor Prozessen gehören, gehören:

- Informationen zur Inhaltsbewertung
- Funktionen zur Barrierefreiheit (Untertitel, Audiobeschreibungen)
- Hinweise zu Inhalten
- Einschränkungen bei der Wiedergabe

MediaTailor behält diese Flags bei der Manifestverarbeitung bei, um sicherzustellen, dass alle Inhaltsmetadaten im personalisierten Manifest beibehalten werden.

Wenn Sie sich mit diesen erweiterten Konzepten vertraut machen, können Sie bei der Konfiguration MediaTailor eine optimale Leistung sicherstellen und alle Probleme beheben, die in Ihren DASH-Streaming-Workflows auftreten können.

Konzepte zum Einfügen von Anzeigen

Hier finden Sie einen Überblick über die Konzepte im Zusammenhang mit der Anzeigenschaltung.

Und verfügbar

Eine bestimmte Werbezeiteinheit innerhalb einer Werbepause, die an Werbetreibende verkauft werden kann. Eine Werbepause kann mehrere verfügbare Werbeanzeigen enthalten. Wenn MediaTailor eine VAST-Antwort von einem Anzeigenentscheidungsserver erhält, füllt er diese Verfügbarkeiten mit personalisierten Anzeigen.

Werbepause

Der Zeitraum während der Programmierung, in dem Werbespots gezeigt werden. Werbeunterbrechungen können vor dem Inhalt (Pre-Roll), während des Inhalts (Mid-Roll) oder nach dem Inhalt (Post-Roll) auftreten. MediaTailor identifiziert Werbeunterbrechungen in Manifesten anhand von Markierungen wie SCTE-35.

Ad Decision Server (ADS)

Ein Server, der Werbespot-Spezifikationen basierend auf Kriterien wie z. B. aktuelle Werbekampagnen und Betrachtervorzügen bereitstellt.

Konfiguration

Ein Objekt MediaTailor , mit dem Sie interagieren. Die Konfiguration enthält Standortinformationen über den Ursprungsserver und den Ad Decision Server (ADS). Die Konfiguration enthält auch Endpunkte, die Eingangs- und Ausgangspunkte bereitstellen. MediaTailor

Dynamische Transcodierung

Ein Prozess, der Qualität und Format der Werbung an die primären Videoinhalte anpasst, wenn Inhalte angefordert werden. Dynamische Transcodierung reduziert die Speicheranforderungen und sorgt für einen nahtlosen Übergang der Wiedergabe zwischen Werbung und Videoinhalten.

Manifeste Manipulation

Bei diesem Prozess werden Manifeste vom Ursprungs-Server so umgeschrieben, dass sie auf die entsprechenden Werbe- und Inhalts-Fragmente verweisen. Anzeigen werden anhand der VAST-Antwort des Ad Decision Servers (ADS) bestimmt. Fügt im Verlauf der Wiedergabe MediaTailor die Anzeige ein oder ersetzt sie in den Inhaltsstream.

VAST und VMAP

Video Ad Serving Template (VAST) und Video Multiple Ad Playlist (VMAP) sind XML-Antworten, die der Anzeigenentscheidungsserver auf Anzeigenanfragen sendet. MediaTailor Die Antworten bestimmen, welche Anzeigen in das MediaTailor Manifest eingefügt werden. VMAP beinhaltet auch den Zeitpunkt für Werbeunterbrechungen und die darin enthaltene Verfügbarkeit von Werbeanzeigen. Weitere Informationen über die der MediaTailor-Werbeeinblendung zugrunde liegende Logik finden Sie unter [Das Verhalten bei der AWS Elemental MediaTailor Anzeigeneinblendung verstehen](#). Weitere Information zur Funktionsweise von MediaTailor mit VAST finden Sie unter [the section called “Anforderungen an die Ad-Server-Integration”](#).

Konzepte zur Kanalmontage

Hier finden Sie einen Überblick über die Konzepte, die sich auf die Kanalmontage beziehen.

Kanäle

Ein Kanal fügt Ihre Quellmanifeste zu einem linearen Stream zusammen. Jeder Kanal hat einen oder mehrere Ausgänge, die die Wiedergabe ermöglichen, auf die Spieler URLs zugreifen. Die Kanalausgänge entsprechen den Paketkonfigurationseinstellungen, die Sie für Ihre VOD-Quellen erstellen. Ein Kanal enthält einen Zeitplan, der festlegt, wann VOD-Quellen im Stream des Kanals wiedergegeben werden.

Konfiguration Package

Eine Packager-Konfiguration ist eine Darstellung Ihrer VOD-Quelle, die bestimmte Merkmale des Paketformats enthält. Sie verknüpfen Ihre Paketkonfigurationen mit Kanalausgängen, um Wiedergabestreams für die Paketformate Ihrer VOD-Quelle zu erstellen, z. B. HTTP Live Streaming (HLS).

Zeitplan

Jeder Kanal besteht aus Programmen, die im Zeitplan des Kanals angeordnet sind. Der Zeitplan bestimmt, zu welcher Uhrzeit die Programme im linearen Stream des Kanals abgespielt werden.

Standorte der Quellen

Ein Quellstandort stellt den Ursprungsserver dar, auf dem Ihre Ressourcen gespeichert sind. Es kann sich um Amazon S3, einen HTTP-Server, ein Content Delivery Network (CDN) oder eine Verpackungsinfrastruktur wie MediaPackage handeln.

VOD-Quellen

Eine VOD-Quelle steht für einen einzelnen Inhalt, z. B. einen Film oder eine Folge einer Fernsehsendung. Sie verknüpfen VOD-Quellen mit Programmen, um sie dem linearen Stream Ihres Kanals hinzuzufügen.

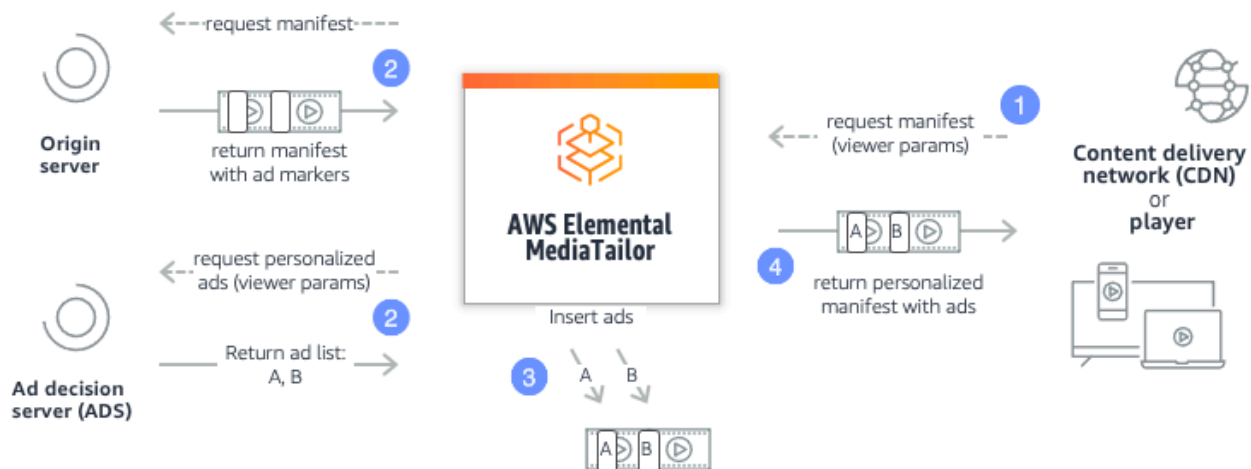
Publikum

Eine Zielgruppe definiert eine Zuschauerkohorte, die optional alternative Inhalte haben kann. Sie können Zielgruppen auf linearen Standardkanälen definieren.

Wie funktioniert das Einfügen von MediaTailor Anzeigen

AWS Elemental MediaTailor interagiert zwischen Ihrem Content Delivery Network (CDN), Ihrem Ursprungsserver und Ihrem Ad Decision Server (ADS), um personalisierte Anzeigen in Werbeunterbrechungen innerhalb von Live- und Video-on-Demand-Inhalten einzubinden.

Im Folgenden finden Sie eine Übersicht darüber, wie das Einfügen von MediaTailor Anzeigen funktioniert:



1. Ein Player oder CDN wie Amazon CloudFront sendet eine Anfrage MediaTailor für HLS- oder DASH-Inhalte an. Die Anfrage enthält Parameter des Players mit Informationen über den Zuschauer, die für die Personalisierung von Anzeigen verwendet werden.
2. MediaTailor sendet eine Anfrage an das ADS, die die Zuschauerinformationen enthält. Der ADS wählt Werbespots basierend auf den Informationen zum Betrachter und aktuellen Werbekampagnen aus. Sie gibt die URLs an die Werbekunden in einer VAST- oder VMAP-Antwort an zurück. MediaTailor

Wenn Sie die Anzeigen vorab konditioniert haben, URLs verweisen sie auf die vortranskodierte Anzeigen. Informationen zum Anzeigen-Stitching mit vortranskodierte Anzeigen finden Sie unter [Vorkonditionierte Anzeigen](#)

3. MediaTailor manipuliert das Manifest so, dass es die Anzeige enthält, die vom ADS URLs zurückgegeben wurde, und zwar transcodiert, sodass sie den Kodierungsmerkmalen des ursprünglichen Inhalts entspricht. Wenn Sie vorkonditionierte Anzeigen verwenden, liegt es in Ihrer Verantwortung, sicherzustellen, dass die Anzeige dem Vorlagenmanifest entspricht.

Wenn eine Anzeige noch nicht so transkodiert wurde, dass sie dem Inhalt entspricht, MediaTailor wird das Einfügen übersprungen und die Anzeige wird MediaConvert zur Vorbereitung der Anzeige verwendet, sodass sie für die nächste Anfrage bereit ist.

4. MediaTailor sendet das vollständig personalisierte Manifest an das anfragende CDN oder den Player zurück.

Das ADS verfolgt die aufgerufenen Anzeigen anhand von Meilensteinen wie dem Beginn der Anzeige, der Mitte der Anzeige und dem Ende der Anzeige. Im Verlauf der Wiedergabe MediaTailor sendet der Player oder der Player Werbe-Tracking-Beacons an die ADS-Anzeigen-Tracking-URL, um aufzuzeichnen, wie viel von einer Anzeige angesehen wurde. Bei der Sitzungsinitialisierung mit gibt der Player an MediaTailor, ob er diese Beacons für die Sitzung senden MediaTailor soll oder soll.

Hinweise zu den ersten Schritten mit der Anzeigeneinfügung finden Sie unter. [Erste Schritte mit MediaTailor](#)

AWS Elemental MediaTailor Ablauf des Ablaufs der Anzeigeneinfügung

AWS Elemental MediaTailor verarbeitet die Manifestpersonalisierung für die serverseitige Anzeigeneinblendung anhand einer vorhersehbaren Abfolge von Ereignissen. Wenn Sie diesen Ereignisablauf verstehen, können Sie besser verstehen, wie MediaTailor Werbemöglichkeiten in personalisierte Zuschauererlebnisse umgewandelt werden, und Sie erhalten eine Grundlage für die Behebung von Problemen.

Jede Gelegenheit zur Anzeigenschaltung folgt einer chronologischen Abfolge von Ereignissen, die zur besseren Sichtbarkeit und Überwachung MediaTailor protokolliert werden. Diese Ereignisse stellen wichtige Meilensteine im Prozess der Personalisierung von Manifesten dar, von der Erkennung einer Werbechance bis hin zur Bereitstellung von Tracking-Informationen.

Standardabfolge von Ereignissen

Bei MediaTailor erfolgreicher Bearbeitung einer Verkaufschance für eine Anzeige tritt in der Regel die folgende Abfolge von Ereignissen ein:

1. Erkennung von Werbechancen — MediaTailor erkennt eine Werbemarkierung (z. B. SCTE-35) im Inhaltsmanifest, was darauf hinweist, dass eine Möglichkeit zur Anzeigenpersonalisierung verfügbar ist.
2. Ad Decision Server-Anfrage — MediaTailor sendet eine Anfrage an den konfigurierten Ad Decision Server (ADS), um Werbung für die erkannte Opportunity abzurufen, einschließlich der Zuschauerparameter und der Dauer der Werbeunterbrechung.
3. Verarbeitung von Anzeigenantworten — MediaTailor erhält eine Antwort vom ADS, die Informationen zur Anzeigengestaltung URLs, Tracking und Metadaten zu den Werbeanzeigen enthält, die eingebunden werden sollen. Weitere Informationen zu den Antwortformaten VAST, VMAP und VPAID finden Sie unter. [Anforderungen an die Ad-Server-Integration](#)
4. Personalisierung mit Manifest — verarbeitet MediaTailor erfolgreich die Anzeigenantwort, transkodiert die Werbeanzeigen, falls erforderlich, und generiert ein personalisiertes Manifest, das

sowohl relative Inhalte URLs für die Anzeigensegmente als auch den ursprünglichen Inhalt enthält. MediaTailor sendet dann das personalisierte Manifest an das Wiedergabegerät.

5. Aktivierung eines Tracking-Beacons — Ein Tracking-Beacon wird ausgelöst, um Anzeigenereignisse (wie Impressionen, Quartile und Fertigstellung) an den Anzeigenserver oder andere Messsysteme zurückzumelden. Im serverseitigen Berichtsmodus (Standard) wird das Beacon auf der Grundlage von Spielersegmentanfragen MediaTailor ausgelöst. Im clientseitigen Berichtsmodus löst das Wiedergabegerät den Beacon mithilfe der im personalisierten Manifest URLs angegebenen Daten aus.

Typisches Timing: Diese gesamte Sequenz ist in der Regel innerhalb von 2 bis 5 Sekunden abgeschlossen, abhängig von der Antwortzeit des Ad Decision Servers und den Anforderungen an die Verarbeitung der Werbegestaltung.

Häufige Varianten von Ereignissen

Nicht alle Versuche, Anzeigen einzufügen, folgen der standardmäßigen erfolgreichen Reihenfolge. MediaTailor behandelt verschiedene Szenarien, die den Ereignisablauf verändern können:

Leeres Anzeigenantwort-Szenario

Wenn der Anzeigenentscheidungsserver keine Werbung zurückgibt:

1. Die Erkennung von Werbechancen erfolgt normal
2. Die Anfrage des Ad Decision Servers wurde erfolgreich gesendet
3. ADS gibt eine leere Antwort ohne Werbung zurück
4. Das personalisierte Manifest für diese Opportunity enthält keine Werbung

Dies ist ein normales Szenario, das aufgrund der Verfügbarkeit des Anzeigeninventars, der Targeting-Kriterien oder der auf dem Anzeigenentscheidungsserver konfigurierten Geschäftsregeln auftreten kann. In diesem Fall werden die zugrunde liegenden Inhalte angezeigt, sofern es sich nicht um einen live-to-VOD Live-Stream oder Stream handelt und Slate konfiguriert wurde. Weitere Informationen finden Sie unter [MediaTailor Schiefer und Einfügung](#).

Fehlerszenarien

Wenn technische Probleme eine erfolgreiche Anzeigeneinblendung verhindern, kann der Ereignisablauf an verschiedenen Stellen unterbrochen werden:

- Timeout für den Anzeigenentscheidungsserver — Die Anfrage an den ADS überschreitet den konfigurierten Timeout-Schwellenwert, wodurch der Anzeigenabruf verhindert wird.
- Kommunikationsfehler — Netzwerk- oder Verbindungsprobleme MediaTailor verhindern, dass der Anzeigenentscheidungsserver erreicht wird.
- Fehler bei der Antwortanalyse — Das ADS gibt eine Antwort zurück, die aufgrund eines ungültigen VAST-Formats oder einer ungültigen VAST-Struktur MediaTailor nicht analysiert werden kann.

In Fehlerszenarien wird in MediaTailor der Regel die Wiedergabe von Inhalten ohne Werbung fortgesetzt, um ein unterbrechungsfreies Zuschauererlebnis zu gewährleisten. Bei Live- oder live-to-VOD Streams mit konfigurierter Slate werden möglicherweise stattdessen Slate-Inhalte angezeigt. Weitere Informationen finden Sie unter [MediaTailor Schiefer und Einfügung](#).

VAST-Umleitungsszenarien

Wenn die erste VAST-Antwort eine Weiterleitung zu einem anderen Werbeserver enthält:

1. Die Erkennung von Werbemöglichkeiten und die erste ADS-Anfrage erfolgen normal
2. Die erste Antwort enthält eine Weiterleitungsanweisung
3. MediaTailor folgt der Weiterleitung, um die endgültige Anzeigenantwort abzurufen
4. Die Manifest-Personalisierung wird mit dem endgültigen Anzeigeninhalt fortgesetzt
5. Die Aktivierung des Tracking-Beacons erfolgt normal

VAST-Weiterleitungen sind in programmatischer Werbung weit verbreitet und ermöglichen die Verkettung von Werbeservern und die Entscheidungsfindung in Echtzeit. MediaTailor ermöglicht bis zu 7 Wrapper-Weiterleitungen und eine unbegrenzte Anzahl von HTTP-Weiterleitungen während des Anzeigenabrufs.

Überblick über den Zeitplan der Ereignisse

Wenn Sie die zeitlichen Merkmale von MediaTailor Ereignisabläufen verstehen, können Sie angemessene Erwartungen an die Leistung der Anzeigeneinblendung setzen:

- Erkennung von Werbechancen — Tritt auf MediaTailor, wenn eine `GetManifest` Anfrage vom Player eingeht und auf die Werbemarkierungen in der Inhaltsliste gestoßen wird.
- Interaktion mit dem Ad Decision Server — In der Regel dauert es je nach ADS-Antwortzeit und Netzwerkbedingungen 100 bis 500 Millisekunden.

- **Manifeste Personalisierung** — Wird in der Regel innerhalb von 50 bis 200 Millisekunden nach Erhalt der ADS-Antwort abgeschlossen.
- **Timing des Tracking-Beacons** — Variiert je nach Berichtsmodus. Bei serverseitigen Beacons basiert das Timing auf Segmentanfragen vom Client-Player; clientseitige Beacons werden auf der Grundlage der Player-Implementierung ausgelöst.

Überlegungen zur Leistung: Die Gesamtzeit von der Erkennung von Werbechancen bis zur Auslieferung des Manifests sollte in der Regel unter 5 Sekunden liegen, um ein optimales Zuschauererlebnis zu gewährleisten. Einzelne Komponenten werden wie oben beschrieben innerhalb von Millisekunden abgeschlossen. Der 5-Sekunden-Schwellenwert berücksichtigt jedoch mögliche Timeout-Werte, Wiederholungsversuche und Netzwerkvariabilität, die während des gesamten Workflows zur Anzeigenschaltung auftreten können. Längere Verzögerungen können auf ADS-Leistungsprobleme oder Netzwerkverbindungsprobleme hindeuten.

Nutzung von Wissen über den Ablauf von Ereignissen

Das Verständnis des MediaTailor Ereignisablaufs bietet die Grundlage für:

- **Implementierungsplanung** — Wenn Sie die Reihenfolge der Ereignisse kennen, können Sie die Player-Integration und die Anzeigenserverkonfiguration besser planen.
- **Leistungsoptimierung** — Das Verständnis der zeitlichen Erwartungen ermöglicht die Identifizierung von Engpässen und Optimierungsmöglichkeiten.
- **Vorbereitung auf die Fehlerbehebung** — Wenn Sie mit normalen Ereignisabläufen vertraut sind, können Sie leichter erkennen, wenn etwas schief geht.

Eine ausführliche Anleitung zur Problembehebung mithilfe der Ereignisflussanalyse finden Sie unter [Fehlerbehebung](#). Technische Informationen zur Ereignisprotokollierung und -überwachung finden Sie unter [Anzeigen von Protokollen](#).

Zugehörige Services

- Amazon CloudFront ist ein globaler Content Delivery Network (CDN) -Service, der Daten und Videos sicher an Ihre Zuschauer liefert. Verwenden Sie CloudFront, um Inhalte mit der bestmöglichen Leistung bereitzustellen. Weitere Informationen zu CloudFront finden Sie auf der [CloudFrontAmazon-Website](#).

- AWS Elemental MediaPackage ist ein just-in-time Verpackungs- und Originalservice, der Live-Video-Assets für die Verteilung in einem Format anpasst, das mit dem Gerät, das die Anfrage stellt, kompatibel ist. Verwenden Sie AWS Elemental MediaPackage als Ursprungsserver, um Inhalte vorzubereiten und Werbemarker hinzuzufügen, bevor Sie Streams an MediaTailor senden. Weitere Informationen zur MediaTailor Funktionsweise mit Originalservern finden Sie unter [Wie funktioniert das Einfügen von MediaTailor Anzeigen](#)
- AWS Identity and Access Management (IAM) ist ein Webdienst, mit dem Sie den Zugriff Ihrer Benutzer auf AWS Ressourcen sicher kontrollieren können. Verwenden Sie IAM, um zu kontrollieren, wer Ihre AWS Ressourcen nutzen kann (Authentifizierung) und welche Ressourcen sie auf welche Weise verwenden können (Autorisierung). Weitere Informationen finden Sie unter [Einrichten AWS Elemental MediaTailor](#).

Zugreifen MediaTailor

Sie können MediaTailor über die Konsole des Dienstes darauf zugreifen.

Greifen Sie auf Ihre zu, AWS-Konto indem Sie Anmeldeinformationen angeben, mit denen bestätigt wird, dass Sie zur Nutzung der Dienste berechtigt sind.

Verwenden Sie den folgenden Link, um sich bei der MediaTailor Konsole anzumelden: **<https://console.aws.amazon.com/mediatailor/home>**.

Preisgestaltung für MediaTailor

Wie bei anderen AWS Produkten gibt es keine Verträge oder Mindestverpflichtungen für die Nutzung MediaTailor. Ihnen werden Gebühren basierend auf Ihrer Nutzung des Service berechnet. Weitere Informationen finden Sie unter [MediaTailor Preise](#).

Regionen für MediaTailor

Um die Datenlatenz in Ihren Anwendungen zu reduzieren, MediaTailor bietet es regionale Endpunkte, an denen Sie Ihre Anfragen stellen können. Eine Liste der Regionen, in denen MediaTailor es verfügbar ist, finden Sie unter [Regionale Endpunkte](#).

Einrichten AWS Elemental MediaTailor

Dieser Abschnitt führt Sie durch die Schritte zum Konfigurieren von Benutzern für den Zugriff auf AWS Elemental MediaTailor. Hintergrundinformationen und zusätzliche Informationen zur Identitäts- und Zugriffsverwaltung für finden Sie MediaTailor unter [Identity and Access Management für AWS Elemental MediaTailor](#).

Führen Sie die folgenden Schritte aus AWS Elemental MediaTailor, um mit der Verwendung zu beginnen.

Themen

- [Melden Sie sich an für eine AWS-Konto](#)
- [Erstellen eines Benutzers mit Administratorzugriff](#)

Melden Sie sich an für eine AWS-Konto

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

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie <https://portal.aws.amazon.com/billing/die-Anmeldung>.
2. Folgen Sie den Online-Anweisungen.

Während der Anmeldung erhalten Sie einen Telefonanruf oder eine Textnachricht und müssen einen Verifizierungscode über die Telefontasten eingeben.

Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontos wird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Benutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um [Aufgaben auszuführen, die Root-Benutzerzugriff erfordern](#).

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Du kannst jederzeit deine aktuellen Kontoaktivitäten einsehen und dein Konto verwalten, indem du zu <https://aws.amazon.com/> gehst und Mein Konto auswählst.

Erstellen eines Benutzers mit Administratorzugriff

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

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich [AWS-Managementkonsole](#) als Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter [Anmelden als Root-Benutzer](#) im AWS-Anmeldung -Benutzerhandbuch zu.

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

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

Erstellen eines Benutzers mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

Anweisungen finden Sie unter [Aktivieren AWS IAM Identity Center](#) im AWS IAM Identity Center - Benutzerhandbuch.

2. Gewähren Sie einem Administratorbenutzer im IAM Identity Center Benutzerzugriff.

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

Anmelden als Administratorbenutzer

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

Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie [im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Access-Portal](#).

Weiteren Benutzern Zugriff zuweisen

1. Erstellen Sie im IAM-Identity-Center einen Berechtigungssatz, der den bewährten Vorgehensweisen für die Anwendung von geringsten Berechtigungen folgt.

Anweisungen hierzu finden Sie unter [Berechtigungssatz erstellen](#) im AWS IAM Identity Center - Benutzerhandbuch.

2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

Eine genaue Anleitung finden Sie unter [Gruppen hinzufügen](#) im AWS IAM Identity Center - Benutzerhandbuch.

Erste Schritte mit MediaTailor

Für den Einstieg MediaTailor können Sie zwischen zwei Tutorials wählen: eines zur Einrichtung der Anzeigeneinfügung und eines zur Kanalmontage. Sie können auch einen praktischen Workshop für Live-Events ausprobieren. Das Tutorial zum Einfügen von Anzeigen führt Sie durch das Senden einer Wiedergabeanfrage, MediaTailor um personalisierte Anzeigen in Ihren Content-Stream zu integrieren. In der Anleitung zur Kanalzusammenstellung erfahren Sie, wie Sie den Stream Ihres Kanals, einschließlich der personalisierten Anzeigen, mithilfe einer Wiedergabe-URL direkt in einem Browser ansehen können.

Sie können auch den [Live Event Framework Workshop](#) erkunden. Dieser praxisnahe Workshop führt Sie durch den gesamten Lebenszyklus eines AWS Elemental MediaTailor Live-Event-Setups, von der ersten Entdeckung und Bereitstellung bis hin zur Optimierung und Skalierung. Sie können den Workshop für eine einfache Konfiguration verfolgen, die ein MediaTailor Setup für alle Ereignisse und Kanäle verwendet, oder für eine erweiterte Architektur, die spezielle Konfigurationen pro Event oder Kanal verwendet.

Themen

- [Erste Schritte mit der MediaTailor Anzeigenschaltung](#)
- [Erste Schritte mit der MediaTailor Kanalmontage](#)

Erste Schritte mit der MediaTailor Anzeigenschaltung

Für die Verwendung benötigen Sie die Berechtigungen AWS-Konto und AWS Elemental MediaTailor, um auf MediaTailor Konfigurationen zuzugreifen, sie anzusehen und zu bearbeiten. Weitere Informationen hierzu finden Sie unter [Einrichten AWS Elemental MediaTailor](#).

Im Tutorial "Erste Schritte" erfahren Sie, wie Sie die folgenden Aufgaben ausführen:

- Bereiten Sie Ihre HLS- oder DASH-Inhaltsstreams vor
- Konfigurieren Sie eine Vorlagen-URL für einen Ad Decision Server (ADS)
- Erstellen Sie eine MediaTailor Konfiguration, die einen Wiedergabeendpunkt enthält
- Verwenden Sie Ihren Player oder Ihr Content Delivery Network (CDN), um eine Wiedergabeanforderung zu stellen MediaTailor

Informationen zur erweiterten ADS-URL-Konfiguration mithilfe dynamischer Variablen und Player-Parameter finden Sie unter [MediaTailor dynamische Anzeigenvariablen für ADS-Anfragen](#). Hinweise zur Weitergabe von Parametern finden CDNs Sie unter [MediaTailor Manifest-Abfrageparameter](#).

Wenn Sie fertig sind, können Sie eine Wiedergabeanfrage MediaTailor für personalisierte Werbeinhalte in Ihrem Stream senden.

Themen

- [Voraussetzungen](#)
- [Schritt 1: Zugriff AWS Elemental MediaTailor](#)
- [Schritt 2: Bereiten Sie einen Stream vor](#)
- [Schritt 3: Konfigurieren Sie die ADS-Anforderungs-URL und die Abfrageparameter](#)
- [Schritt 4: Erstellen einer Konfiguration](#)
- [Schritt 5: Testen Sie die Konfiguration](#)
- [Schritt 6: Senden Sie die Wiedergabeanforderung an AWS Elemental MediaTailor](#)
- [Schritt 7 \(optional\): AWS Elemental MediaTailor Aktivität überwachen](#)
- [Schritt 8: Bereinigen](#)

Voraussetzungen

Bevor Sie beginnen, sollten Sie sicherstellen, dass Sie die in [Einrichten AWS Elemental MediaTailor](#) beschriebenen Schritte ausgeführt haben.

Schritt 1: Zugriff AWS Elemental MediaTailor

Melden Sie sich mit Ihren IAM-Anmeldeinformationen unter an der MediaTailor Konsole an **`https://console.aws.amazon.com/mediatailor/home`**.

Schritt 2: Bereiten Sie einen Stream vor

Konfigurieren Sie Ihren Ursprungs-Server so, dass mit ihm Manifeste für HLS oder DASH erstellt werden, die mit AWS Elemental MediaTailor kompatibel sind. Ihr Ursprungsserver muss Standard-HTTP-Ports (Port 80) oder HTTPS-Ports (Port 443) verwenden. MediaTailor unterstützt keine benutzerdefinierten Ports für die Kommunikation mit dem Ursprungsserver.

Bereiten Sie einen HLS-Stream vor

HLS-Manifeste müssen die folgenden Anforderungen erfüllen:

- Manifeste müssen über das öffentliche Internet zugänglich sein.
- Die Manifeste müssen live oder Video-on-Demand (VOD) sein.
- Manifeste müssen einen EXT-X-VERSION von 3 oder höher haben.
- Für Live-Inhalte müssen Manifeste Marker zur Abgrenzung von Ad-Avails enthalten. Dies ist für On-Demand-Inhalte optional, bei denen stattdessen VMAP-Timeoffsets verwendet werden können.

In der Manifestdatei müssen sich Werbeplätze mit den folgenden Markern befinden:

- #EXT-X-CUE-OUT /#EXT -X-CUE-IN (häufiger) mit einer Dauer wie im folgenden Beispiel gezeigt.

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

- #EXT-X-DATERANGE (weniger gebräuchlicher) mit Angaben zur Dauer wie im folgenden Beispiel dargestellt.

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

Alle für #EXT-X-DATERANGE angezeigten Felder sind erforderlich.

Von der Art und Weise, wie Sie die Werbe-Marker im Manifest konfigurieren, ist abhängig, ob Werbung in einen Stream eingefügt wird oder ob andere Fragmente im Stream durch sie ersetzt werden. Weitere Informationen finden Sie unter [the section called “Das Verhalten bei der Insertion von Anzeigen verstehen”](#).

- [Multivariante HLS-Playlisten müssen der HLS-Spezifikation entsprechen, die unter HTTP-Livestreaming: Master-Playlist-Tags dokumentiert ist](#). Insbesondere muss #EXT-X-STREAM-INF die Felder RESOLUTION, BANDWIDTH und CODEC enthalten.

Nachdem Sie den Stream konfiguriert haben, notieren Sie sich das URL-Präfix für den Ursprung des Inhalts für die multivariante Playlist. Sie benötigen es zum Erstellen der Konfiguration in AWS Elemental MediaTailor an späterer Stelle in diesem Tutorial.

Bereiten Sie einen DASH-Stream vor

DASH-Manifeste müssen die folgenden Anforderungen erfüllen:

- Manifeste müssen über das öffentliche Internet zugänglich sein.
- Die Manifeste müssen live oder Video-on-Demand (VOD) sein.
- Manifeste müssen Ereignisse als Ad-Avails markieren und dazu entweder Splice-Insert-Marker oder Zeitsignal-Marker verwenden. Sie können die Werbemarker im transparenten XML- oder im base64-kodierten binären Format angeben. Für das Einfügen von Spleißverbindungen muss der out-of-network Indikator aktiviert sein. Bei Zeitsignal-Markierungen muss es sich bei der Segmentierungstyp-ID, die sich innerhalb der Segmentierungs-UPID befindet, um einen Cue-Out-Wert handeln, der von erkannt wird. AWS Elemental MediaTailor Der Ad-Avail beginnt am Anfang des Ereignisses und dauert über die Ereignisdauer an, sofern angegeben, oder bis das nächste Ereignis beginnt.

Das folgende Beispiel zeigt ein Ereignis, das mittels Splice-Insert-Marker als Ad-Avail gekennzeichnet ist. Die Dauer dieses Ad-Avail entspricht der Ereignisdauer.

```
<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>
```

- Ad-Avails müssen die gleichen AdaptationSet- und Representation-Einstellungen wie die Streams der Inhalte besitzen. AWS Elemental MediaTailor verwendet diese Einstellungen zur Transcodierung der Werbung entsprechend dem Stream der Inhalte, um einen reibungslosen Wechsel zwischen ihnen zu ermöglichen.

Nachdem Sie den Stream konfiguriert haben, beachten Sie das Präfix der Ursprungs-URL von Inhalten für das DASH-Manifest. Sie benötigen sie, um die Konfiguration später in AWS Elemental MediaTailor diesem Tutorial zu erstellen.

Schritt 3: Konfigurieren Sie die ADS-Anforderungs-URL und die Abfrageparameter

Um zu bestimmen, welche Abfrageparameter der ADS benötigt, erstellen Sie eine Werbe-Tag-URL vom ADS. Diese URL fungiert als Vorlage für Anforderungen an den ADS und enthält Folgendes:

- Statische Werte
- Werte, die von AWS Elemental MediaTailor (bezeichnet mit `session` oder `avail` Abfrageparametern) generiert wurden
- Von Playern generierte Werte, die von der Client-Anwendung abgerufen wurden (gekennzeichnet durch `player_params.-`Abfrageparameter)

Example Werbe-Tag-URL von einem ADS

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

Wobei Folgendes gilt:

- `output` und `content_id` sind statische Werte
- `playerSession= [session.id]` ist ein dynamischer Wert, der von bereitgestellt wird. AWS Elemental MediaTailor Der Wert von `[session.id]` ändert sich mit jeder Player-Sitzung und ergibt eine jeweils andere URL für die VAST-Anforderung für jede Sitzung.
- `cust_params` sind vom Player bereitgestellte dynamische Werte

Die multivariante Playlist-Anfrage des Players muss Schlüssel-Wert-Paare bereitstellen, die den Abfrageparametern in der ADS-Anforderungs-URL entsprechen. `player_params.` Weitere

Hinweise zur Konfiguration von Schlüssel-Wert-Paaren in der Anfrage an finden Sie unter. AWS Elemental MediaTailor [MediaTailor dynamische Anzeigenvariablen für ADS-Anfragen](#)

Geben Sie die konfigurierte „Vorlage“-URL ein, wenn Sie das server/ADS Ursprungs-Mapping in MediaTailor, in erstellen. [Schritt 4: Erstellen einer Konfiguration](#)

Testen

Sie können eine statische VAST-Antwort von Ihrem ADS für Testzwecke verwenden. Im Idealfall gibt die VAST-Antwort eine MP4 Wiedergabe in Mezzanine-Qualität zurück, die transkodiert werden kann. AWS Elemental MediaTailor Wenn die ADS-Antwort mehrere Wiedergabeversionen enthält, wird die Wiedergabeversion mit der höchsten Qualität und Auflösung MediaTailor MP4 ausgewählt und an den Transcoder gesendet.

Schritt 4: Erstellen einer Konfiguration

Die AWS Elemental MediaTailor Konfiguration enthält Zuordnungsinformationen für den Ursprungsserver und ADS.

So erstellen Sie eine Konfiguration (Konsole)

1. Öffnen Sie die MediaTailor Konsole unter <https://console.aws.amazon.com/mediatailor/>.
2. Wählen Sie auf der Seite Configurations (Konfigurationen) die Option Create configuration (Konfiguration erstellen).
3. Geben Sie im Bereich Configuration (Konfiguration) unten auf der Seite für Configuration name (Konfigurationsname) einen eindeutigen Namen zur Beschreibung der Konfiguration ein. Der Name ist der primäre Kennung für die Konfiguration. Die maximale zulässige Länge beträgt 512 Zeichen.
4. Geben Sie unter Videoinhaltsquelle das URL-Präfix für die multivariante HLS-Wiedergabeliste oder das DASH-Manifest für diesen Stream ein, abzüglich der Asset-ID. Wenn die URL der multivarianten Playlist beispielsweise lautet `http://origin-server.com/a/master.m3u8`, würden Sie Folgendes eingeben. `http://origin-server.com/a/` Alternativ können Sie ein kürzeres Präfix, wie z. B. `http://origin-server.com`, eingeben, Sie müssen dann aber `/a/` in die Komponenten-ID der Player-Anforderung für Inhalte einschließen. Die maximale Länge beträgt 512 Zeichen.

 Note

Wenn der Ursprungs-Server Ihrer Inhalte HTTPS verwendet, muss sein Zertifikat von einer bekannten Zertifizierungsstelle stammen. (Es darf kein selbstsigniertes Zertifikat sein.) Andernfalls AWS Elemental MediaTailor kann keine Verbindung zum Inhaltsursprung hergestellt werden und es können keine Manifeste als Antwort auf Spieleranfragen bereitgestellt werden.

5. Geben Sie für Ad Decision Server die URL für Ihre ADS ein. Dies ist entweder die unter [Schritt 3: Konfigurieren Sie die ADS-Anforderungs-URL und die Abfrageparameter](#) beschriebene URL mit Variablen oder die statische VAST-URL, die Sie für Testzwecke verwenden. Die maximale Länge beträgt 25.000 Zeichen.

 Note

Wenn der ADS HTTPS verwendet, muss sein Zertifikat von einer bekannten Zertifizierungsstelle stammen. (Es darf kein selbstsigniertes Zertifikat sein.) Das Gleiche gilt für Mezzanine-Anzeigen, die vom ADS URLs zurückgegeben werden. Andernfalls kann MediaTailor keine Werbespots vom Ursprungs-Server der Inhalte abrufen und in den Manifesten zusammenfügen.

6. (Optional, je nach Bedarf für DASH) Wählen Sie für Standort die Option DEAKTIVIERT, wenn Sie CDN-Routing-Regeln für den Zugriff auf MediaTailor Manifeste eingerichtet haben und entweder clientseitige Berichte verwenden oder Ihre Player Sticky-HTTP-Weiterleitungen unterstützen.

Weitere Informationen über die Funktion Location (Speicherort) finden Sie unter [the section called “Standort-Funktion”](#).

7. (Optional) Wenn Ihr Ursprungs-Server DASH-Manifeste mit einzelnen Zeiträumen erstellt, wählen Sie DASH mpd manifest origin typ (DASH mpd-Manifest-Ursprungstyp) und danach SINGLE_PERIOD aus. MediaTailor behandelt DASH-Manifeste standardmäßig als Manifeste mit mehreren Perioden. Weitere Informationen finden Sie unter [the section called “Integration einer MPEG-DASH-Quelle”](#).
8. Wählen Sie Create configuration (Konfiguration erstellen).

AWS Elemental MediaTailor zeigt die neue Konfiguration auf der Konfigurationsseite an.

Schritt 5: Testen Sie die Konfiguration

Nachdem Sie die Konfiguration gespeichert haben, testen Sie den Stream mithilfe einer URL im entsprechenden Format für Ihr Streaming-Protokoll:

- Beispiel: HLS

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

- Beispiel: DASH

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

Wobei Folgendes gilt:

- `playback-endpoint` ist der eindeutige Wiedergabe-Endpunkt, den AWS Elemental MediaTailor beim Erstellen der Konfiguration generiert hat.

Beispiel

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

- `hashed-account-id` ist deine AWS-Konto ID.

Beispiel

```
777788889999
```

- `origin-id` ist der Name, den Sie beim Erstellen der Konfiguration gewählt haben.

Beispiel

```
myOrigin
```

- `master.m3u8` oder `manifest.mpd` ist der Name des Manifests aus dem Test-Stream samt seiner Dateierweiterung. Definieren Sie diese so, dass Sie ein vollständig identifiziertes Manifest erhalten, wenn Sie diese an die Quelle der Videoinhalte anhängen, die Sie unter [the section called “Schritt 4: Erstellen einer Konfiguration”](#) konfiguriert haben.

Unter Verwendung der Werte aus den vorangegangenen Beispielen ergeben sich die vollständigen URLs Werte wie folgt.

- Beispiel: HLS

```
https://7777888899999.mediatailor.us-east-1.amazonaws.com/v1/master/  
AKIAIOSFODNN7EXAMPLE/myOrigin/master.m3u8
```

- Beispiel: DASH

```
https://7777888899999.mediatailor.us-east-1.amazonaws.com/v1/dash/  
AKIAIOSFODNN7EXAMPLE/myOrigin/manifest.mpd
```

Sie können den Stream mit einer der folgenden Methoden testen.

- Geben Sie wie im vorherigen Beispiel dargestellt die URL in einem eigenständigen Player ein.
- Testen Sie den Stream in Ihrer eigenen Player-Umgebung.

Schritt 6: Senden Sie die Wiedergabeanforderung an AWS Elemental MediaTailor

Konfigurieren Sie den Downstream-Player oder das CDN zum Senden von Wiedergabeanforderungen an den von AWS Elemental MediaTailor bereitgestellten Wiedergabe-Endpunkt der Konfiguration. Alle vom Player definierten dynamischen Variablen, die Sie in der ADS-Anforderungs-URL in [Schritt 3: Konfigurieren Sie die ADS-Anforderungs-URL und die Abfrageparameter](#) verwendet haben, müssen in der Manifest-Anforderung vom Player definiert sein.

Example

Angenommen, Ihre Vorlagen-ADS-URL lautet wie folgt.

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

Definieren Sie dann `[player_params.cust_params]` in der Player-Anfrage, indem Sie dem Schlüssel-Wert-Paar das Präfix voranstellen. `ads.` AWS Elemental MediaTailor übergibt Parameter, denen kein Präfix vorangestellt ist `ads.`, an den Ursprungsserver statt an das ADS.

Die Player-Anforderungs-URL ist eine Variante der folgenden HLS- und DASH-Beispiele.

```
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
```

Wenn AWS Elemental MediaTailor die Spieleraanfrage erhält, definiert er die Spielervariablen auf der Grundlage der Informationen in der Anfrage. Die resultierende ADS-Anforderungs-URL ist eine Variante hiervon.

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

Weitere Informationen über das Konfigurieren von Schlüssel-Wert-Paaren zur Übergabe an den ADS finden Sie unter [MediaTailor dynamische Anzeigenvariablen für ADS-Anfragen](#).

Schritt 7 (optional): AWS Elemental MediaTailor Aktivität überwachen

Verwenden Sie Amazon CloudWatch und Amazon CloudWatch Logs, um AWS Elemental MediaTailor Aktivitäten nachzuverfolgen, z. B. die Anzahl der Anfragen, Fehler und ausgefüllten Werbeanzeigen.

Wenn Sie zum ersten Mal verwenden CloudWatch AWS Elemental MediaTailor, erstellen Sie eine AWS Identity and Access Management (IAM) -Rolle, um die Kommunikation zwischen den Services zu ermöglichen.

Um den AWS Elemental MediaTailor Zugriff auf CloudWatch (Konsole) zu ermöglichen

1. Öffnen Sie unter <https://console.aws.amazon.com/iam/> die IAM-Konsole.
2. Klicken Sie im Navigationsbereich der IAM-Konsole auf Rollen, und wählen Sie dann Rolle erstellen.
3. Wählen Sie den AWS-Konto Rollentyp „Anderer“.
4. Geben Sie als Konto-ID Ihre AWS-Konto ID ein.
5. Wählen Sie Require external ID (Externe ID fordern) und geben Sie **midas** ein. Mit dieser Option wird automatisch eine Bedingung zur Vertrauensrichtlinie hinzugefügt, die es dem Service nur

dann ermöglicht, die Rolle anzunehmen, wenn in der Anforderung die richtige `sts:ExternalID` enthalten ist.

6. Wählen Sie Weiter: Berechtigungen aus.
7. Fügen Sie eine Berechtigungsrichtlinie hinzu, in der die Aktionen angegeben werden, die von dieser Rolle ausgeführt werden dürfen. Wählen Sie eine der folgenden Optionen aus und wählen Sie dann Next: Review (Nächster Schritt: Prüfen):
 - CloudWatchLogsFullAccessum vollen Zugriff auf Amazon CloudWatch Logs zu gewähren
 - CloudWatchFullAccessum vollen Zugriff auf Amazon zu gewähren CloudWatch
8. Geben Sie für Role name (Rollenname) den Namen **MediaTailorLogger** ein und klicken Sie auf Create role (Rolle erstellen).
9. Wählen Sie auf der Seite Roles (Rollen) die von Ihnen soeben erstellte Rolle aus.
10. Bearbeiten Sie die Vertrauensstellung, um den Prinzipal zu aktualisieren:
 1. Wählen Sie auf der Seite Summary (Übersicht) der Rolle die Registerkarte Trust relationship (Vertrauensstellung) aus.
 2. Wählen Sie Vertrauensstellung bearbeiten aus.
 3. Ändern Sie im Richtliniendokument den Prinzipal auf den AWS Elemental MediaTailor - Service. Sie sollte wie folgt aussehen.

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

Die gesamte Richtlinie sollte folgendermaßen lauten.

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

Schritt 8: Bereinigen

Um unerwünschte Gebühren zu vermeiden, löschen Sie alle unnötigen Konfigurationen.

So löschen Sie eine Konfiguration (Konsole)

1. Führen Sie auf der Seite AWS Elemental MediaTailor Konfigurationen einen der folgenden Schritte aus:

- Wählen Sie für die Konfiguration, die Sie löschen möchten, einen Namen für Configuration name (Konfigurationsname) aus.
 - Aktivieren Sie in der Spalte Configuration name (Konfigurationsname) das Optionsfeld neben dem Namen und wählen Sie dann Delete (Löschen).
2. Geben Sie im Bestätigungsdialogfeld Delete configuration (Konfiguration löschen) den Text **Delete** ein und wählen Sie erneut Delete (Löschen) aus.

AWS Elemental MediaTailor entfernt die Konfiguration.

Erste Schritte mit der MediaTailor Kanalmontage

Im Tutorial "Erste Schritte" erfahren Sie, wie Sie die folgenden Aufgaben ausführen:

- Erstellen Sie einen Quellspeicherort und fügen Sie diesem Quellinhalt hinzu
- Erstelle einen Kanal
- Erstelle eine Programmliste, um die Inhalte deines Kanals nach einem bestimmten Zeitplan abzuspielen
- Fügen Sie mithilfe AWS Elemental MediaTailor der Anzeigeneinfügung personalisierte Anzeigen zum Kanal-Stream hinzu

Wenn du fertig bist, kannst du einen Browser öffnen, die Wiedergabe-URL für deinen Kanal eingeben und dir den Stream deines Kanals ansehen, der personalisierte Werbung enthält.

Dieses Tutorial führt dich durch die grundlegenden Schritte, um mit der MediaTailor Kanalmontage zu beginnen. Weiterführende Informationen finden Sie unter [Wird verwendet AWS Elemental MediaTailor , um linear zusammengestellte Streams zu erstellen](#).

Geschätzte Kosten

- Die Gebühr für einen aktiven Kanal beträgt 0,10\$ pro Stunde. Für Kanäle, die inaktiv sind, werden Ihnen keine Gebühren berechnet.

Themen

- [Voraussetzungen](#)
- [Schritt 1: Erstellen Sie einen Quellspeicherort](#)

- [Schritt 2: Fügen Sie VOD-Quellen zu Ihrem Quellstandort hinzu](#)
- [Schritt 3: Erstellen Sie einen Kanal](#)
- [Schritt 4: Füge Programme zum Zeitplan deines Kanals hinzu](#)
- [Schritt 5 \(optional\): Verwenden Sie diese Option MediaTailor , um personalisierte Anzeigen in Ihren Stream einzufügen](#)
- [Schritt 6: Starte deinen Kanal](#)
- [Schritt 7: Teste deinen Kanal](#)
- [Schritt 8: Bereinigen](#)

Voraussetzungen

Bevor Sie mit diesem Tutorial beginnen, müssen Sie die folgenden Anforderungen erfüllen:

- Vergewissern Sie sich, dass Sie die Schritte unter abgeschlossen haben [Einrichten AWS Elemental MediaTailor](#).
- Sie müssen über Ressourcen sowohl für VOD-Quellinhalte als auch für Ad Slate verfügen. Sie müssen den Pfad zu den Manifesten für die Assets kennen.

Note

Wenn Sie ABR (Automated Adaptive Bitrate) oder die Kodierung pro Titel verwenden, müssen Sie Ihre Inhalte so codieren, dass alle Varianten dieselbe Länge haben und dieselbe Anzahl an untergeordneten Titeln haben. Wir empfehlen, eine Kodierungsvorlage mit einer Mindestsegmentlänge von einer Sekunde zu verwenden.

Schritt 1: Erstellen Sie einen Quellspeicherort

Ein Quellspeicherort stellt den Ursprungsserver dar, auf dem Ihre Inhalte gespeichert sind. Dabei kann es sich um Amazon S3, einen Standard-Webserver, ein Content Delivery Network (CDN) oder einen Verpackungshersteller handeln, z. AWS Elemental MediaPackage

MediaTailor ruft die Inhaltsmanifeste von Ihrem Quellverzeichnis ab und verwendet sie, um ein Live-Slide-Manifestfenster zusammenzustellen, das auf die zugrunde liegenden Inhaltssegmente verweist.

Gehen Sie wie folgt vor, um einen Quellspeicherort zu erstellen.

Um einen Quellspeicherort zu erstellen

1. Öffnen Sie die Media Tailor Konsole unter <https://console.aws.amazon.com/mediatailor/>.
2. Wählen Sie im Navigationsbereich Channel-Assembly > Source Locations aus.
3. Wählen Sie in der Navigationsleiste die Option Quellspeicherort erstellen aus.
4. Geben Sie unter Konfiguration des Quellverzeichnisses eine Kennung und den Speicherort Ihres Quellinhalts ein:
 - Name: Eine Kennung für Ihren Quellstandort, z. B. my-origin.
 - Basis-URL: Die Basis-URL des Ursprungsservers, auf dem Ihre Inhalte gehostet werden, z. B. <https://111111111111.cloudfront.net>. Die URL muss in einem Standard-HTTP-URL-Format mit dem Präfix <http://> oder <https://> vorliegen.
5. Wählen Sie Quellspeicherort erstellen aus.

Schritt 2: Fügen Sie VOD-Quellen zu Ihrem Quellstandort hinzu

Nachdem du einen oder mehrere Quellorte für deinen Kanal definiert hast, kannst du eine oder mehrere VOD-Quellen hinzufügen. Jede VOD-Quelle steht für einen einzelnen Inhalt, z. B. einen einzelnen Film, eine Folge einer TV-Sendung oder einen Highlight-Clip.

Sie müssen mindestens eine Paketkonfiguration für Ihre VOD-Quelle erstellen. Jede Paketkonfiguration enthält das Paketformat und die Manifesteinstellungen für Ihre VOD-Quellen. Anschließend fügen Sie Ihrem Kanal Ihre Paketkonfigurationen hinzu, um Ausgaben zu erstellen.

Sie können mehrere Paketkonfigurationen verwenden, um unterschiedliche Kanalausgänge zu erstellen. Wenn Ihre VOD-Quelle beispielsweise sowohl als HLS als auch als DASH verpackt ist, können Sie zwei Paketkonfigurationen für jedes Format erstellen. Sie können dann die Quellgruppen der Paketkonfiguration verwenden, um zwei Kanalausgänge zu erstellen: einen für HLS, einen für DASH.

Um VOD-Quellen hinzuzufügen und Paketkonfigurationen zu erstellen

1. Öffnen Sie die MediaTailor Konsole unter. <https://console.aws.amazon.com/mediatailor/>
2. Wählen Sie im Navigationsbereich Channel-Assembly > Source Locations aus.
3. Wählen Sie im Bereich Quellverzeichnisse den Quellspeicherort aus, den Sie in dem [Um einen Quellspeicherort zu erstellen](#) Verfahren erstellt haben.
4. Wählen Sie VOD-Quelle hinzufügen.

5. Geben Sie unter VOD-Quelldetails einen Namen für Ihre VOD-Quelle ein, z. B. my-example-video
6. **source-group-name** Geben Sie unter Paketkonfigurationen > Informationen zur Paketkonfiguration ein:

 Note

Die Paketkonfigurationen Ihrer Quelle müssen alle dieselbe Dauer haben, wie im Manifest der Quelle festgelegt. Und alle Quellen innerhalb einer Paketkonfiguration müssen dieselbe Anzahl von untergeordneten Streams haben. Um diese Anforderungen zu erfüllen, empfehlen wir Ihnen, eine Kodierungsvorlage für Ihre Assets zu verwenden. Wir empfehlen, eine Kodierungsvorlage mit einer Mindestsegmentlänge von einer Sekunde zu verwenden. MediaTailor unterstützt kein Streaming pro Titel oder automatisiertes Streaming mit adaptiver Bitrate (ABR), da diese Kodierungsmethoden gegen diese Anforderungen verstoßen.

- Quellgruppe: Geben Sie einen Quellgruppennamen ein, der diese Paketkonfiguration beschreibt, z. B. HLS-4k. Notieren Sie sich diesen Namen. Sie werden ihn angeben, wenn Sie die Ausgabe Ihres Kanals erstellen. Weitere Informationen finden Sie unter [Verwende Quellgruppen mit den Ausgaben deines Kanals](#).
 - Typ: Wählen Sie das Paketformat für diese Konfiguration aus. MediaTailor unterstützt HLS und DASH.
 - Relativer Pfad: Der relative Pfad von der Basis-HTTP-URL des Quellspeicherorts zum Manifest. Zum Beispiel /my/path/index.m3u8.
7. Wählen Sie Add source (Quelle hinzufügen) aus.
 8. Wiederholen Sie die Schritte 4-7 in diesem Verfahren, um die VOD-Quelle für Ihre Anzeigenliste hinzuzufügen.

Schritt 3: Erstellen Sie einen Kanal

Ein Kanal fasst Ihre Quellen zu einem linearen Live-Stream zusammen. Jeder Kanal enthält einen oder mehrere Ausgänge, die den Paketkonfigurationen Ihrer VOD-Quelle entsprechen.

Zuerst erstellen Sie einen Kanal und fügen dann Ihre VOD-Quellen dem Zeitplan des Kanals hinzu, indem Sie Programme erstellen.

So erstellen Sie einen Kanal

1. Öffnen Sie die MediaTailor Konsole unter <https://console.aws.amazon.com/mediatailor/>
2. Wählen Sie im Navigationsbereich Channel-Assembly > Channels aus.
3. Wählen Sie in der Navigationsleiste die Option Kanal erstellen aus.
4. Gib unter Kanaldetails Details zu deinem Kanal ein:
 - Name: Gib einen Namen für deinen Kanal ein.
 - Wiedergabemodus: Legt fest, welche Art von Programmübergängen zulässig sind und was mit einem Programm passiert, wenn es beendet ist. Verwenden Sie den Standard-Loop-Modus.
5. Wählen Sie Weiter aus.
6. Definieren Sie unter Ausgabedetails die Einstellungen für diese Ausgabe:
 - Manifestname: Geben Sie einen Manifestnamen ein, z. **index**B. MediaTailor hängt die Formaterweiterung an, z. B. .m3u8 für HLS.

Note

Sie müssen pro Kanalausgabe einen eindeutigen Manifestnamen eingeben.

- Formattyp: Wählen Sie das Streaming-Format für den Kanal aus. DASH und HLS werden unterstützt. Wählen Sie das Format, das der Paketkonfiguration entspricht, in [Schritt 1: Erstellen Sie einen Quellspeicherort](#) der Sie es erstellt haben.
 - Quellgruppe: Geben Sie den Namen der Quellgruppe ein, in der Sie erstellt haben [Schritt 1: Erstellen Sie einen Quellspeicherort](#).
7. Geben Sie unter Manifesteinstellungen zusätzliche Informationen zu Ihren Manifesteinstellungen ein:
 - Manifestfenster (Sekunden): Das Zeitfenster (in Sekunden), das in jedem Manifest enthalten ist. Der Mindestwert ist 30 Sekunden und der Höchstwert ist 3600 Sekunden.
 8. Wählen Sie Weiter aus.
 9. Wählen Sie unter Kanalrichtlinie die Option Kanalrichtlinie nicht anhängen aus. Diese Option beschränkt die Wiedergabe nur auf Personen, die Zugriff auf Ihre AWS-Konto Anmeldeinformationen haben.
 10. Wählen Sie Weiter aus.
 11. Überprüfen Sie Ihre Einstellungen im Bereich Überprüfen und erstellen.

12. Wählen Sie Create channel (Channel erstellen).

Note

Kanäle werden im Status „Gestoppt“ erstellt. Dein Kanal wird erst aktiv sein, wenn du ihn startest.

Schritt 4: Füge Programme zum Zeitplan deines Kanals hinzu

Jetzt, wo du einen Kanal hast, fügst du Programme zum Zeitplan des Kanals hinzu. Jedes Programm enthält eine VOD-Quelle von einem Quellort in Ihrem Konto. Der Senderplan bestimmt die Reihenfolge, in der Ihre Programme im Stream des Kanals abgespielt werden.

Jedes Programm kann eine oder mehrere Werbeunterbrechungen haben. Sie fügen eine Werbeunterbrechung ein, indem Sie eine VOD-Quelle angeben, die als Werbetafel verwendet werden soll. Die Dauer der Werbeunterbrechung wird durch die Dauer der Werbepause bestimmt. Sie können optional einen serverseitigen Anzeigeneinfügungsserver, z. B. die MediaTailor Anzeigeneinfügung, verwenden, um Ihre Werbeunterbrechungen zu personalisieren.

Um Programme zum Zeitplan deines Kanals hinzuzufügen

1. Öffne die MediaTailor Konsole unter <https://console.aws.amazon.com/mediatailor/>.
2. Wählen Sie im Navigationsbereich Channel-Assembly > Channels aus.
3. Wählen Sie im Bereich Kanäle den Kanal aus, den Sie in dem [Schritt 3: Erstellen Sie einen Kanal](#) Verfahren erstellt haben.
4. Geben Sie unter Programmdetails Details zu Ihrem Programm ein:
 - Name: Dies ist der Name des Programms, das zum Zeitplan deines Kanals hinzugefügt werden soll.
 - Name des Quellstandorts: Wähle „Bestehenden Quellstandort auswählen“ und wähle den Quellstandort [Schritt 1: Erstellen Sie einen Quellspeicherort](#) aus, den du im Drop-down-Menü „Quellstandort auswählen“ erstellt hast.
 - Name der VOD-Quelle: Wählen Sie „Eine vorhandene VOD-Quelle auswählen“ und wählen Sie die VOD-Quelle aus, die Sie zuvor in diesem Tutorial erstellt haben.
5. Definieren Sie unter Wiedergabekonfiguration, wie und wann ein Programm in den Zeitplan eines Kanals eingefügt wird:

- **Übergangstyp:** Dieser Wert ist auf Relativ festgelegt. Der relative Übergangstyp gibt an, dass dieses Programm relativ zu anderen Programmen in der Programmliste auftritt.
- **Relative Position:** Wenn dies das erste Programm im Zeitplan Ihres Kanals ist, können Sie diese Einstellung überspringen. Wenn es nicht das erste Programm im Programm Ihres Kanals ist, wähle aus, an welcher Stelle in der Programmliste das Programm angehängt werden soll. Du kannst „Vor der Sendung“ oder „Nach der Sendung“ auswählen.
- **Relatives Programm:** Wenn dies das erste Programm in Ihrem Zeitplan ist, können Sie diese Einstellung überspringen. Wenn es nicht das erste Programm im Programm Ihres Kanals ist, wähle Bestehendes Programm verwenden und wähle den Namen des Programms aus, in dem du es erstellt hast [Um Programme zum Zeitplan deines Kanals hinzuzufügen](#).

6.

Wähle „Werbeunterbrechung hinzufügen“ aus. Konfigurieren Sie unter Werbeunterbrechungen die Einstellungen für die Werbeunterbrechung:

- **Name des Slate-Quellspeicherorts:** Wählen Sie „Bestehenden Quellspeicherort auswählen“ und wählen Sie den Quellspeicherort aus, an dem Ihr Slate gespeichert ist, den Sie zuvor in diesem Tutorial erstellt haben.
- **Name der VOD-Quelle:** Wählen Sie „Eine vorhandene VOD-Quelle auswählen“ und wählen Sie die VOD-Quelle aus, die Sie für Slate verwenden und die Sie zuvor in diesem Tutorial hinzugefügt haben. Die Dauer des Slate bestimmt die Dauer der Werbepause.
- **Für Offset in Millisekunden:** Dieser Wert bestimmt die Startzeit der Werbeunterbrechung in Millisekunden als Offset im Verhältnis zum Beginn des Programms. Geben Sie einen Wert ein, der kürzer als die Dauer der VOD-Quelle ist und der an einer Segmentgrenze für alle Titel innerhalb der VOD-Quelle des Programms (alle Audio-, Video- und Untertitelspuren) ausgerichtet ist. Andernfalls wird die Werbeunterbrechung übersprungen. Wenn Sie beispielsweise 0 eingeben, entsteht eine Pre-Roll-Werbeunterbrechung, die vor Programmbeginn abgespielt wird. Hinweis:.

7. Wählen Sie Programm hinzufügen.

Weitere Informationen zu Programmen finden Sie unter [Configuring ad breaks for your program](#).

Weiterführende Informationen zur Verwendung von Anzeigen mit Ihrem linearen Stream finden Sie unter [Optionale Konfigurationseinstellungen](#).

Schritt 5 (optional): Verwenden Sie diese Option MediaTailor , um personalisierte Anzeigen in Ihren Stream einzufügen

Sie haben jetzt einen Kanal mit Programmen. Wenn du möchtest, kannst du MediaTailor damit personalisierte Werbung in die Werbeunterbrechungen deiner Programme im Stream des Kanals einfügen.

Voraussetzungen

Bevor Sie fortfahren können, müssen Sie die folgenden Anforderungen erfüllen:

- Sie müssen über einen Ad Decision Server (ADS) verfügen.
- Sie müssen im Rahmen des [Ein Programm zum Zeitplan eines Kanals hinzufügen](#) Verfahrens die Einstellungen für Werbeunterbrechungen konfiguriert haben.

Um personalisierte Werbung zum Stream deines Kanals hinzuzufügen, verwende MediaTailor

1. Öffne die MediaTailor Konsole unter <https://console.aws.amazon.com/mediatailor/>.
2. Wählen Sie im Navigationsbereich Konfigurationen aus.
3. Geben Sie unter Erforderliche Einstellungen die grundlegenden erforderlichen Informationen zu Ihrer Konfiguration ein:
 - Name: Der Name Ihrer Konfiguration.
 - Inhaltsquelle: Gib die Wiedergabe-URL aus der Ausgabe deines Kanals ein, abzüglich des Dateinamens und der Erweiterung. Weiterführende Informationen zur MediaTailor Konfiguration finden Sie unter [Erforderliche Einstellungen](#).
 - Entscheidungsserver hinzufügen: Geben Sie die URL für Ihr ADS ein.
4. Sie können optional die Konfigurationsalias, die Personalisierungsdetails und die erweiterten Einstellungen konfigurieren. Informationen zu diesen Einstellungen finden Sie unter [Optionale Konfigurationseinstellungen](#)
5. Wählen Sie in der Navigationsleiste Konfiguration erstellen aus.

Weiterführende Informationen zur Verwendung der MediaTailor Anzeigeneinfügung finden Sie unter [AWS Elemental MediaTailor Zum Einfügen von Anzeigen verwenden](#).

Schritt 6: Starte deinen Kanal

Du hast jetzt einen Kanal. Bevor Sie jedoch auf den Stream des Kanals zugreifen können, müssen Sie Ihren Kanal starten. Wenn Sie versuchen, auf einen Kanal zuzugreifen, bevor er aktiv ist, wird ein 4xx HTTP-Fehlercode MediaTailor zurückgegeben.

Starte deinen Kanal

1. Öffne die MediaTailor Konsole unter <https://console.aws.amazon.com/mediatailor/>.
2. Wählen Sie im Navigationsbereich Channel-Assembly > Channels aus.
3. Wählen Sie in der Navigationsleiste „Start“ aus.

Schritt 7: Teste deinen Kanal

Um zu überprüfen, ob dein Kanal ordnungsgemäß funktioniert, öffne einen Webbrowser und gib die URL aus der Ausgabe deines Kanals ein. Du solltest den Stream deines Kanals sehen.

In einigen Fällen müssen Sie möglicherweise den Cache löschen, um das erwartete Verhalten zu sehen.

Schritt 8: Bereinigen

Wenn Sie mit dem Kanal fertig sind, den Sie für dieses Tutorial erstellt haben, sollten Sie ihn löschen, indem Sie ihn löschen.

Sobald sich der Kanalstatus auf Gestoppt ändert, fallen keine Gebühren mehr für diesen Kanal an. Wenn du deinen Kanal für später behalten möchtest, ohne dass Gebühren anfallen, kannst du den Kanal jetzt beenden und später wieder starten.

Um deinen Kanal zu löschen

1. Öffne die MediaTailor Konsole unter <https://console.aws.amazon.com/mediatailor/>.
2. Wählen Sie im Navigationsbereich Channel-Assembly > Channels aus.
3. Wählen Sie den Kanal aus, den Sie löschen möchten.
4. Wenn dein Kanal läuft, wähle im Drop-down-Menü Aktionen die Option Stopp aus. Du musst deinen Kanal beenden, bevor du ihn löschen kannst.
5. Wenn dein Kanal gestoppt ist, wähle im Drop-down-Menü Aktionen die Option Löschen aus.

AWS Elemental MediaTailor Zum Einfügen von Anzeigen verwenden

Eine Konfiguration ist ein Objekt, mit dem Sie in AWS Elemental MediaTailor interagieren. Die Konfiguration enthält die Zuordnungsinformationen für den Ursprungsserver und den Ad Decision Server (ADS). Sie können auch eine Standardwiedergabe definieren MediaTailor , die verwendet wird, wenn eine Anzeige nicht verfügbar ist oder nicht das gesamte Anzeigenangebot ausfüllt.

Wenn Sie ein Content Delivery Network (CDN) mit verwenden MediaTailor, müssen Sie die Verhaltensregeln im CDN einrichten, bevor Sie CDN-Informationen zur Konfiguration hinzufügen. Weitere Informationen zum Einrichten Ihres CDN finden Sie unter [Verwendung eines CDN zur Optimierung der MediaTailor Anzeigenpersonalisierung und Inhaltsbereitstellung](#).

Themen

- [Unterstützte Audio- und Videocodecs](#)
- [Das Verhalten bei der AWS Elemental MediaTailor Anzeigeneinblendung verstehen](#)
- [MediaTailor Überblick und Implementierung der servergesteuerten Anzeigeneinfügung](#)
- [MediaTailor Anforderungen an die Ad-Server-Integration](#)
- [MediaTailor Verwaltung der Wiedergabekonfiguration](#)
- [Integration einer Inhaltsquelle für die MediaTailor Anzeigeneinfügung](#)
- [Integration AWS Elemental MediaTailor mit Google Ad Manager](#)
- [Anpassen des Verhaltens von Werbeunterbrechungen mit der Unterdrückung von Werbeunterbrechungen](#)
- [MediaTailor Stoßstange und Einlage](#)
- [MediaTailor Einfügen von Pre-Roll-Anzeigen](#)
- [MediaTailor Schiefer und Einfügung](#)
- [Anzeigen vorab abrufen](#)
- [Verwenden von vorkonditionierten Anzeigen mit AWS Elemental MediaTailor](#)
- [MediaTailor dynamische Anzeigenvariablen für ADS-Anfragen](#)
- [MediaTailor Manifest-Abfrageparameter](#)
- [Daten melden und verfolgen](#)
- [Overlay-Anzeigen](#)

- [Werbe-ID-Dekoration](#)

Unterstützte Audio- und Videocodecs

MediaTailor unterstützt die folgenden Codecs.

- Audio-Codecs: , und mp4a ac-3 ec-3
- Videocodecs: h.264 (AVC), (), h.265 (HEVC) av01 AV1

Das Verhalten bei der AWS Elemental MediaTailor Anzeigeneinblendung verstehen

AWS Elemental MediaTailor fügt Anzeigen in Live- oder Video-on-Demand-Inhalte (VOD) ein, indem Anzeigen entweder ersetzt oder in das Original-Manifest eingefügt werden. Ob Anzeigen eingefügt oder ersetzt werden, hängt davon ab, wie die Werbeunterbrechungen im Ursprungsmanifest konfiguriert sind und ob es sich um VOD- oder Live-Inhalte handelt. Bei einer Werbeunterbrechung handelt es sich um den Zeitraum während der Programmierung, in dem Werbespots gezeigt werden, wohingegen Ad-Avails die spezifischen Einheiten der Werbezeit innerhalb einer Werbepause sind, die mit Werbung gefüllt werden können.

- Bei der Anzeigenersetzung werden Inhaltssegmente durch Anzeigen MediaTailor ersetzt.
- MediaTailor Fügt beim Einfügen von Anzeigen Anzeigeninhalte dort ein, wo es keine Segmente gibt.

Informationen dazu, wie MediaTailor Anzeigen in Live- und VOD-Inhalte eingebunden werden, erhalten Sie unter dem entsprechenden Thema.

Topics

- [Verhalten beim Zusammenfügen von Anzeigen bei VOD](#)
- [Verhalten beim Zusammenfügen von Live-Anzeigen](#)

Verhalten beim Zusammenfügen von Anzeigen bei VOD

MediaTailor fügt Anzeigen in VOD-Inhalte ein oder ersetzt sie, je nachdem, wie die Anzeigenmarkierungen im Ursprungsmanifest konfiguriert sind und ob der Ad Decision Server (ADS) VMAP-Antworten sendet.

Die nachstehenden Abschnitte gehen auf das Werbeverhalten je nach Marker-Konfiguration ein.

Wenn Werbemarkierungen vorhanden sind

AWS Elemental MediaTailor fügt Anzeigen ein, bei denen SCTE-35-Werbemarkierungen im Ursprungsmanifest vorhanden sind. Werbemarkierungen mit dem EXT-X-CUE-OUT Wert 0 Dauer weisen auf das Einfügen einer Anzeige hin.

Richtlinien für HLS-Werbemarkierungen

Beachten Sie diese Richtlinien für die SCTE-Signalisierung nach dem Rollvorgang und beim Ad-Pod:

Pre-Roll-Anzeigen

Für HLS-Post-Rolls müssen dem letzten Inhaltssegment CUE-OUT/IN-Markierungen vorangestellt werden. Der Grund hierfür ist, dass laut HLS-Spezifikation Tag-Decoratoren explizit vor einem Segment deklariert werden müssen.

Betrachten wir beispielsweise die folgende Deklaration.

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

AWS Elemental MediaTailor fügt eine Postroll-Anzeige wie die folgende ein.

```
#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: Pods hinzufügen

CUE-OUT/INTags müssen explizit an ein Segment angehängt werden. Es ist nicht möglich, mehrere CUE-OUT/IN-Tags nacheinander zu verwenden, um Werbe-Pod-Verhalten nachzuahmen.

Die folgende Deklaration ist beispielsweise eine gültige Verwendung von CUE-OUT/IN, um einen Werbe-Pod darzustellen.

```
#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
```

Die obige Deklaration führt zu einer Ausgabe wie der folgenden.

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

Die folgende Deklaration ist ungültig.

```
#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
```

Wenn keine Werbemarkierungen vorhanden sind

Werbemarkierungen sind die empfohlene Methode, um Werbeunterbrechungen in einem Manifest zu signalisieren. Werbemarkierungen sind jedoch nicht erforderlich. Wenn das Manifest keine Werbemarkierungen für DASH oder HLS enthält, MediaTailor ruft das ADS einmal auf und erstellt auf der Grundlage der Antwort Werbeunterbrechungen:

- Wenn das ADS eine VAST-Antwort sendet, werden zu Beginn des Manifests alle Anzeigen aus der Antwort in einen Werbeblock MediaTailor eingefügt. Dies wird als Pre-Roll (Vorlauf) bezeichnet.
- Wenn das ADS eine VMAP-Antwort sendet, MediaTailor verwendet es die Zeitversätze für Werbeunterbrechungen, um Pausen zu erstellen und sie zu den angegebenen Zeiten (Pre-Roll, Mid-Roll oder Post-Roll) in das gesamte Manifest einzufügen. MediaTailor verwendet alle Anzeigen aus den einzelnen Werbeunterbrechungen in der VMAP-Antwort für jede Werbeunterbrechung im Manifest.

Note

Wenn ein Segment einen Einfügepunkt mit VMAP für VOD-Inhalten überlappt, rundet MediaTailor auf den nächsten Einfügepunkt ab.

Tip

Wenn Sie Mid-Roll-Werbepausen erstellen möchten, der ADS aber nicht VMAP unterstützt, muss das Manifest Werbemarken enthalten. MediaTailor fügt (wie in den nachstehenden Abschnitten beschrieben) an den Markern Werbung ein.

Note

Bei Methoden zur servergesteuerten MediaTailor Anzeigeneinfügung werden Pre-Roll-Anzeigen oben im Manifest eingefügt und der Player spielt sie vor anderen Anzeigentypen ab.

Verhalten beim Zusammenfügen von Live-Anzeigen

Führt in Livestreams AWS Elemental MediaTailor immer eine Anzeigenersetzung durch, wobei die Gesamtzeit zwischen den Anzeigenmarkierungen so genau wie möglich beibehalten wird. Wenn Anzeigenmarkierungen das DURATION Attribut enthalten, bestimmt MediaTailor anhand des Werts die Dauer der Werbeunterbrechung. In Live-Workflows muss jeder CUE-OUT Indikator eine Dauer oder einen entsprechenden CUE-IN Indikator haben.

MediaTailor führt einen Werbersatz für HLS- und DASH-Live-Inhalte durch. Informationen zur MediaTailor Berechnung der Platzierung und des Zeitpunkts von Werbeunterbrechungen finden Sie unter [the section called “Werbe-Marker”](#) und [the section called “Werbe-Marker”](#)

Auswahl und Ersetzung von Anzeigen

AWS Elemental MediaTailor beinhaltet Werbung aus der VAST-Antwort des Ad Decision Servers (ADS) wie folgt:

- Wenn eine Dauer angegeben ist, wird eine Gruppe von Anzeigen MediaTailor ausgewählt, die in die Dauer passen, und schließt sie ein.
- Wenn keine Dauer angegeben ist, werden so viele Anzeigen wie möglich MediaTailor abgespielt, bis eine Werbemarkierung erscheint, die auf eine Rückkehr zum Hauptinhalt hinweist.

AWS Elemental MediaTailor hält sich beim Ersetzen von Live-Anzeigen an die folgenden Richtlinien:

- MediaTailor versucht, komplette Werbeanzeigen abzuspielen, ohne sie auszuschneiden oder zu kürzen.
- Immer wenn Sie MediaTailor auf eine Werbemarkierung stoßen, die auf das Ende der Werbeunterbrechung hinweist, kehrt sie zum zugrunde liegenden Inhalt zurück. Dies kann bedeuten, dass eine laufende Werbung gekürzt wird.
- Kehrt am Ende der Laufzeit zum zugrunde liegenden Inhalt MediaTailor zurück.
- Wenn MediaTailor für die Dauer einer Werbepause keine Werbeanzeigen mehr verfügbar sind, wird entweder der Slate abgespielt, sofern ein solcher konfiguriert ist, oder die Wiedergabe des zugrunde liegenden Inhaltsstreams fortgesetzt. Dies geschieht normalerweise, wenn nicht genügend transkodierte Werbung vorhanden ist, um die Dauer der Werbepause auszufüllen.

 Tip

Mit der Konfigurationseinstellung für eine Personalisierungsschwelle können Sie den Grenzwert für die nicht ausgefüllte Werbezeit in einer Pause festlegen. Weitere Informationen finden Sie in der [PlaybackConfiguration Referenz](#).

Live-Preroll für die serverseitige Anzeigeneinblendung

Live-Preroll funktioniert bei Methoden zur serverseitigen Anzeigeneinfügung anders als bei serverseitiger Anzeigeneinfügung:

Serverseitige Anzeigeneinfügung (gestickter Modus)

Preroll-Anzeigen ersetzen einen Teil der Live-Inhalte zu Beginn der jeweiligen Zuschauersitzung. Jeder Zuschauer sieht Prerolls zu unterschiedlichen Zeiten, je nachdem, wann er dem Stream beitrifft.

Methoden zur servergestützten Anzeigeneinfügung

MediaTailor platziert ein Pre-Roll-Tag für den Datumsbereich an erster Stelle aller Medienmanifeste mit Attributen. CUE="PRE , ONCE" Dadurch müssen Spieler zu Beginn der Wiedergabe einmal Preroll-Anzeigen anfordern und abspielen, obwohl sie dasselbe nicht personalisierte Manifest teilen.

Anforderungen an die Konfiguration:

- Entscheidungsserver für Live-Preroll-Anzeigen: Konfigurieren Sie einen VAST-Endpunkt für Preroll-Anzeigen (kann sich von Mid-Roll-Anzeigen unterscheiden)
- Zulässige Höchstdauer für Live-Preroll-Anzeigen: Legen Sie die maximale Dauer für Preroll-Anzeigen fest (optional — falls nicht angegeben, werden alle zurückgegebenen Anzeigen verwendet)

Technische Umsetzung:

- verwendet das Preroll-Tag für den Datumsbereich START-DATE="1970-01-01T00:00:00.000Z" (Unix-Epoche)

- Für Asset-Listenanfragen für Preroll wird anstelle des regulären ADS der konfigurierte Entscheidungsserver für Preroll-Anzeigen verwendet
- Spieler identifizieren Preroll-Anfragen anhand der Daten `availId="aws-mediataylor-preroll-1"` in der Asset-Liste

Important

Bei Live-Streams decken Preroll-Anzeigen den Inhalt ab, anstatt ihn zu verzögern. Künftige Versionen unterstützen möglicherweise den Modus „Inhaltsverzögerung“ durch zusätzliche Konfigurationsoptionen.

Note

Bei der serverseitigen Anzeigeneinblendung variiert das Preroll-Verhalten zwischen Live- und VOD-Inhalten. Live-Inhalte erfordern eine explizite Preroll-Konfiguration, während VOD-Inhalte standardmäßig Prerolls beinhalten, wobei der reguläre Anzeigenentscheidungsserver verwendet wird.

Beispiele

- Wenn die Dauer der Werbepause auf 70 Sekunden festgelegt ist und die ADS-Antwort zwei 40-Sekunden-Anzeigen enthält, wird eine der 40-Sekunden-Anzeigen AWS Elemental MediaTailor abgespielt. In der übrigen Zeit wechselt es zum konfigurierten Pausenfüller oder zu den zugrunde liegenden Inhalten. Wenn MediaTailor zu irgendeinem Zeitpunkt während dieses Prozesses einen Cue-in-Indikator feststellt, wechselt es sofort zu den zugrunde liegenden Inhalten.
- Wenn die Dauer der Werbeunterbrechung auf 30 Sekunden festgelegt ist und die kürzeste Anzeige in der ADS-Antwort 40 Sekunden lang ist, werden keine Anzeigen MediaTailor abgespielt. Wenn ein Ad Slate konfiguriert ist, wird dieser 30 Sekunden lang MediaTailor abgespielt oder bis ein Cue-In-Indikator angezeigt wird. Andernfalls wird der zugrunde liegende MediaTailor Inhalt abgespielt.

MediaTailor Überblick und Implementierung der servergesteuerten Anzeigeneinfügung

AWS Elemental MediaTailor Die serverseitige Anzeigeneinfügung (SGAI) bietet eine Alternative zur serverseitigen Anzeigeneinfügung, indem auf Anzeigen als separate Playlisten verwiesen wird, anstatt sie direkt in Medien-Playlisten zusammenzufügen. Dieser Ansatz verbessert die Leistung durch zwischenspeicherbare Manifeste und ermöglicht eine bessere Skalierbarkeit.

Informationen zur Verwendung der servergesteuerten Anzeigeneinfügung mit erhalten Sie MediaTailor, indem Sie das entsprechende Thema aus der folgenden Liste auswählen.

In der Wiedergabekonfiguration aktivieren

Damit Spieler die servergesteuerte Anzeigeneinfügung verwenden können, müssen Sie `PLAYER_SELECT` in der MediaTailor Wiedergabekonfiguration die Einstellung `Insertion Mode` auf einstellen. Auf diese Weise können Spieler bei der Sitzungsinitialisierung zwischen gestickter oder gesteuerter Anzeigeneinfügung wählen.

Erstellen Sie eine servergeführte Sitzung

Wählen Sie beim Erstellen von Wiedergabesitzungen den geführten Modus. Wie das geht, hängt davon ab, ob Ihre Spieler implizite oder explizite Sessions verwenden.

Implizit erstellte servergeführte Sitzungen

An die `aws.insertionMode=GUIDED` Multivarianten-Playlist-Anfrage von HLS anhängen. Beispiel:

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

Wobei Folgendes gilt:

- `playback-endpoint` ist der eindeutige Wiedergabe-Endpunkt, den AWS Elemental MediaTailor beim Erstellen der Konfiguration generiert hat.

Beispiel

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

- `hashed-account-id` ist deine ID. AWS-Konto

Beispiel

```
777788889999
```

- `origin-id` ist der Name, den Sie beim Erstellen der Konfiguration gewählt haben.

Beispiel

```
myOrigin
```

- `index.m3u8` oder ist der Name des Manifests aus dem Teststream plus dessen Dateierweiterung. Definieren Sie diese so, dass Sie ein vollständig identifiziertes Manifest erhalten, wenn Sie diese an die Quelle der Videoinhalte anhängen, die Sie unter [the section called “Schritt 4: Erstellen einer Konfiguration”](#) konfiguriert haben.

Unter Verwendung der Werte aus den vorangegangenen Beispielen ergeben sich die vollständigen URLs Werte wie folgt.

- Beispiel:

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

Explizit erstellte servergeführte Sitzungen

Fügen Sie `insertionMode=GUIDED` die JSON-Metadaten hinzu, die der Player im HTTP POST an den Präfixendpunkt für die Sitzungsinitialisierung der MediaTailor Konfiguration sendet.

Das folgende Beispiel zeigt die Struktur der JSON-Metadaten:

```
{  
  # other keys, e.g. "adsParams"  
  "insertionMode": "GUIDED"           # this can be either GUIDED or STITCHED  
}
```

Mit diesen Initialisierungsmetadaten verwendet die Wiedergabesitzung die serverseitige Anzeigeneinfügung.

Anzeigenverfolgung mit SGAI

SGAI unterstützt sowohl serverseitiges als auch clientseitiges Ad-Tracking. Sie legen den Berichtsmodus bei der Sitzungsinitialisierung fest. Der Modus kann sich während der Sitzung nicht ändern.

Serverseitiges Tracking (Standard)

MediaTailor feuert VAST-Beacons automatisch ab, wenn der Player Anzeigensegmente anfordert. Die Anzeige URIs in der Asset-Liste enthält verschlüsselte Beacon-Metadaten (`awsBeaconData`, `awsBeaconDomain`, `awsConfigurationName`). Der Spieler muss die `#EXT-X-DEFINE:QUERYPARAM` HLS-Variablenersetzung unterstützen. Die Antwort auf die Assetliste enthält keinen TRACKING Abschnitt.

Einzelheiten zur Funktionsweise von serverseitigem SGAI-Beaconing finden Sie unter.

[Serverseitiges Tracking mit serverseitiger Anzeigeneinfügung \(SGAI\)](#)

Clientseitiges Tracking

Fügen Sie `aws.reportingMode=CLIENT` es zu Ihrer Anfrage zur Sitzungsinitialisierung hinzu. Die Antwort auf die Asset-Liste enthält einen TRACKING Abschnitt mit einem Beacon URLs , den der Player während der Anzeigenwiedergabe auslöst. Der `GetTracking` API-Endpunkt wird nicht für SGAI-Sitzungen verwendet. Stattdessen enthält jede Antwort auf die Asset-Liste direkte Tracking-Daten. Die Tracking-Daten verwenden dasselbe JSON-Schema wie die Tracking-Antwort auf serverseitige Anzeigeneinfügung (SSAI).

Details hierzu finden Sie unter [Servergesteuerte Anzeigeneinfügung](#).

Geführter Prefetch mit offensichtlichem Herzschlag

Für Live-SGAI-Streams können Sie den manifestbasierten Vorabruf von Anzeigen aktivieren, indem Sie zu Ihrer Anfrage zur Sitzungsinitialisierung Folgendes hinzufügen:

`aws.guidedPrefetchMode=MANIFEST`

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

Wenn diese Option aktiviert ist, wird an jede URL des interstitiellen Medienmanifests (`aws.sessionId=<id>`) in der multivarianten Playlist eine Sitzungs-ID () als Abfrageparameter

MediaTailor angehängt. `/v1/i-media` Jedes Mal, wenn der Player ein i-Media-Manifest aktualisiert, erreicht die Anfrage die Sitzungs-ID, die MediaTailor verwendet wird, um MediaTailor Prefetch-Anfragen für bevorstehende Werbeunterbrechungen in die Warteschlange zu stellen.

- Der `aws.guidedPrefetchMode` Parameter akzeptiert zwei Werte: MANIFEST (aktiviert) und OFF (deaktiviert, Standard).
- Der geführte Prefetch-Modus ist nur für SGAI-Sitzungen gültig. Wenn Sie ihn mit zusammengeführten Sitzungen verwenden, wird ein Fehler zurückgegeben.
- DASH unterstützt den Guided-Prefetch-Modus noch nicht.
- Der geführte Prefetch ist unabhängig vom Berichtsmodus — Beacons werden zur Wiedergabezeit ausgelöst, nicht zur Prefetch-Zeit.
- Zwischenspeichern Sie i-Media-Manifeste nicht in Ihrem CDN, wenn Sie Guided Prefetch verwenden. Der Prefetch-Mechanismus hängt davon ab, ob die Manifest-Aktualisierungsanfragen des Players direkt erreicht werden. MediaTailor Wenn Ihr CDN `/v1/i-media` Antworten zwischenspeichert, die Heartbeat-Anfragen MediaTailor nicht empfängt und kein Prefetching auslösen kann.

MediaTailor Kompatibilitätsmatrix für Funktionen zur servergesteuerten Anzeigeneinfügung

AWS Elemental MediaTailor bietet zwei Methoden zum Einfügen von Anzeigen mit unterschiedlicher Funktionskompatibilität. Die serverseitige Anzeigeneinfügung funktioniert anders als die serverseitige Anzeigeneinfügung, was sich auf die Kompatibilität mit einigen MediaTailor Funktionen auswirkt. Anhand dieser Tabelle erfahren Sie, welche Funktionen mit den einzelnen Methoden zum Einfügen von Anzeigen funktionieren.

Funktionskompatibilität nach Methode zum Einfügen von Anzeigen

Feature	Serverseitige Anzeigeneinfügung (SSAI)	Serverseitige Anzeigeneinfügung (SGAI)
Vorabruf von Anzeigen	✓ Wird unterstützt	Noch nicht unterstützt
Unterdrückung von Anzeigen	✓ Unterstützt	Nicht zutreffend

Feature	Serverseitige Anzeigeneinfügung (SSAI)	Serverseitige Anzeigeneinfügung (SGAI)
Verhalten von Pre-Roll-Anzeigen	Wird durch die Konfiguration MediaTailor gesteuert	Wird durch die MediaTailor Konfiguration gesteuert
Kundenseitiges Ad-Tracking	Verwendet API GetTracking	Benutzt den TRACKING Abschnitt in der Asset-Liste (GetTracking API wird nicht verwendet)
Serverseitiges Ad-Tracking	✓ Unterstützt — Beacons werden basierend auf /v1/segment Anfragen ausgelöst, die die Sitzungs-ID verwenden	✓ Unterstützt (nur HLS) — verwendet Sitzungsloses Beacons mit verschlüsselten Beacon-Daten, die in Ad Via eingebettet sind. URIs #EXT-X-DEFINE:QUERYPARAM Erfordert HLS v11 oder höher. DASH wird noch nicht unterstützt.
Ad-ID-Dekoration	✓ Wird unterstützt	✗ Nicht kompatibel

Einzelheiten zur Kompatibilität

Vorabruf anzeigen

Der Vorabruf von Anzeigen wird derzeit nicht unterstützt.

Unterdrückung von Werbung

Die Anzeigenunterdrückung wird bei servergesteuerten Methoden zur Anzeigeneinfügung nicht unterstützt, da Spieler nur Anzeigen abrufen, die sie auch abspielen werden.

Verhalten von Pre-Roll-Anzeigen

Das Timing von Pre-Roll-Anzeigen funktioniert je nach Einblendmethode unterschiedlich:

- Serverseitige Anzeigeneinfügung: MediaTailor Steuert anhand der Konfigurationseinstellungen, wann Pre-Roll-Anzeigen abgespielt werden
- Serverseitige Anzeigeneinfügung: MediaTailor Fügt Pre-Roll-Anzeigen oben im Manifest ein. Ihr Player zeigt diese Anzeigen zuerst und startet dann Ihre Inhalte

Nachverfolgung von Werbeanzeigen

Das clientseitige Tracking verwendet je nach Methode zur Anzeigenschaltung unterschiedliche Mechanismen:

- Serverseitige Anzeigeneinfügung (SSAI): Verwendet den API-Endpunkt `GetTracking`
- Servergesteuerte Anzeigeneinfügung (SGAI): MediaTailor stellt Informationen zur Nachverfolgung im TRACKING Abschnitt jeder Antwort auf die Assetliste bereit. Der `GetTracking` API-Endpunkt wird nicht verwendet. Die Antwort auf die Initialisierung der Sitzung enthält `keintrackingUrl`.

Das serverseitige Tracking unterscheidet sich auch zwischen den Methoden:

- Serverseitige Anzeigeneinfügung (SSAI): Löst MediaTailor Beacons aus, wenn der Player mithilfe der Sitzungs-ID zusammengefügte Anzeigensegmente abrufen. `/v1/segment/`
- Servergesteuerte Anzeigeneinfügung (SGAI): verwendet sitzungsloses Beaconing. MediaTailor bündelt verschlüsselte Beacon-Daten (`awsBeaconData`, `awsConfigurationName`) in das Anzeigenmanifest URLs ein `awsBeaconDomain`, das in der Asset-Liste zurückgegeben wird. Das Anzeigenmanifest verwendet `#EXT-X-DEFINE:QUERYPARAM` Tags, sodass der Player diese Werte in ein Segment einfügt. URLs Wenn der Player jedes Anzeigensegment anfordert, werden die Daten MediaTailor entschlüsselt, das entsprechende Beacon ausgelöst und zum Inhaltssegment weitergeleitet. Wenn serverseitige Berichterstattung aktiv ist, wird der TRACKING Abschnitt in der Antwort auf MediaTailor die Asset-Liste weggelassen. Details hierzu finden Sie unter [Serverseitiges Tracking mit serverseitiger Anzeigeneinfügung \(SGAI\)](#).

Dekoration der Ad-ID

Die Dekoration von Werbeanzeigen ist nicht mit der serverseitigen Anzeigeneinfügung kompatibel, da die Felder, die die X-AD-CREATIVE-SIGNALING Kopfzeilen füllen, nur bekannt sind, wenn die Asset-Liste abgerufen wird, nicht aber, wenn das Manifest geschrieben wird.

MediaTailor Konfiguration der servergeführten Anzeigeneinfügung für Live-Streams

AWS Elemental MediaTailor Die servergesteuerte Anzeigeneinfügung für Live-Inhalte bietet erhebliche Leistungsvorteile durch zwischenspeicherbare Manifeste. Die Konfiguration von SGAI für Live-Inhalte verwendet dieselben Kernparameter wie VOD, wobei besondere Überlegungen zu den Live-Stream-Eigenschaften und der Echtzeitverarbeitung berücksichtigt werden.

Anforderungen für Live-SGAI

Bevor Sie SGAI für Live-Inhalte aktivieren, stellen Sie sicher, dass Sie über Folgendes verfügen:

- Ihr Livestream enthält ordnungsgemäß formatierte DATERANGE-Markierungen
- Die Dauer von Werbeunterbrechungen ist konsistent und vorhersehbar
- Ihr CDN ist so konfiguriert, dass SGAI-Manifeste entsprechend zwischengespeichert werden
- Spieler unterstützen servergesteuerte Workflows zum Einfügen von Anzeigen
- Ihr Anzeigenentscheidungsserver kann Anfragen nach Live-Inhalten in Echtzeit bearbeiten

Anforderungen an die Spieler

Die Spieler müssen so konfiguriert sein, dass sie SGAI-Live-Manifeste ordnungsgemäß verarbeiten können:

- Support für servergesteuerte Workflows zur Anzeigeneinfügung
- Fähigkeit, Anleitungen zur Anzeigeneinfügung anhand von Manifesten zu verarbeiten
- Richtiger Umgang mit dem Timing und der Synchronisation von Live-Streams
- Für HLS-Inhalte: Support für HLS Version 8 und EXT-X-DATERANGE mit CLASS-Attribut. Version 11 für serverseitiges Beaconsing.
- Für HLS-Inhalte: EXT-X-DEFINE Unterstützung für die Variablenersetzung

Konfiguration der Live-Wiedergabe

Um SGAI für Live-Inhalte zu aktivieren, erstellen Sie eine Wiedergabekonfiguration mit den folgenden Einstellungen:

Example Konfiguration für die SGAI-Live-Wiedergabe

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

Im Folgenden sind die wichtigsten Überlegungen zur Live-SGAI-Konfiguration aufgeführt:

VideoContentSourceUrl

Muss auf einen Live-HLS-Stream mit ordnungsgemäß formatierten SCTE-35-DATERANGE-Markierungen verweisen. Der Stream sollte konsistente Segmentdauern und Bitratenvarianten beibehalten.

SGAI-Live-Manifest-Anfragen

SGAI-Live-Manifeste verwenden dasselbe URL-Muster wie die herkömmliche Anzeigeneinfügung:

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

Manifestbasierter Prefetch für Live-SGAI

Für Live-SGAI-Workflows können Sie manifestbasiertes Prefetch-Heartbeat aktivieren, um die Latenz beim Ausfüllen von Anzeigen zu reduzieren. Fügen Sie der Manifestanforderung Folgendes hinzu:

`aws.guidedPrefetchMode=MANIFEST`

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

Wenn diese Option aktiviert ist, wird eine Sitzungs-ID (`?aws.sessionId=<id>`) als Abfrageparameter an jede URL des interstitiellen Medienmanifests (`/v1/i-media`) in der multivarianten Playlist MediaTailor angehängt. Jedes Mal, wenn der Player ein i-Media-Manifest aktualisiert, wird die Anfrage MediaTailor mit der Sitzungs-ID erreicht. Diese dient dazu, die Sitzung zu identifizieren und Prefetch-Anfragen für MediaTailor bevorstehende Werbeunterbrechungen in die Warteschlange zu stellen.

Important

Zwischenspeichern Sie i-Media-Manifeste nicht in Ihrem CDN, wenn Sie Guided Prefetch verwenden. Der Prefetch-Heartbeating-Mechanismus hängt davon ab, ob die Manifest-Aktualisierungsanforderungen des Players direkt erreicht werden. MediaTailor Wenn Ihr CDN `/v1/i-media` Antworten zwischenspeichert und bereitstellt, die Heartbeat-Anfragen MediaTailor nicht empfängt und kein Prefetching auslösen kann. Konfigurieren Sie Ihr CDN so, dass `/v1/i-media/*` Anfragen nur dann weitergeleitet werden, wenn sie verwendet werden. MediaTailor `aws.guidedPrefetchMode=MANIFEST`

Guided Prefetch ist unabhängig vom Berichtsmodus. Unabhängig davon, ob Sie serverseitiges (Standard) oder clientseitiges (`aws.reportingMode=CLIENT`) Tracking verwenden, werden Beacons zur Wiedergabezeit ausgelöst, nicht, wenn Anzeigen vorab abgerufen werden. Allgemeine Informationen zur Funktionsweise des Vorabrufs von Anzeigen in finden Sie unter. MediaTailor [Anzeigen vorab abrufen](#)

Testen der SGAI-Live-Konfiguration

Überprüfen Sie Ihr SGAI Live-Setup mit diesen Validierungsschritten:

1. Generierung von Testmanifesten

Fordern Sie die URL des SGAI-Live-Manifests an und stellen Sie sicher, dass zwischenspeicherbare Inhalte mit den richtigen Hinweisen zum Einfügen von Anzeigen zurückgegeben werden.

2. Überprüfen Sie das CDN-Caching

Vergewissern Sie sich, dass Ihr CDN SGAI-Manifeste gemäß den konfigurierten TTL-Werten zwischenspeichert.

3. Testen Sie das Einfügen von Anzeigen

Vergewissere dich anhand der Hinweise in den SGAI-Manifesten, dass Spieler erfolgreich Anzeigen einfügen können.

4. Überwachen Sie die Leistung

Verwenden Sie CloudWatch Metriken, um zu überprüfen, ob die Ausgangslast reduziert und die Cache-Trefferquoten verbessert wurden.

MediaTailor Konfiguration der servergeführten Anzeigeneinfügung für VOD-Inhalte

AWS Elemental MediaTailor Die servergesteuerte Anzeigeneinfügung für VOD-Inhalte bietet erhebliche Leistungsvorteile, da Manifeste in hohem Maße zwischengespeichert werden können und die Serververarbeitung reduziert wird. Bei der Konfiguration von SGAI für VOD-Inhalte wird der statische Charakter von video-on-demand Assets genutzt, um die Effizienz beim Zwischenspeichern zu maximieren und Anfragen zu minimieren, sodass es sich ideal für große Inhaltsbibliotheken mit wiederholten Anzeigemustern eignet.

Anforderungen für VOD SGAI

Bevor Sie SGAI für VOD-Inhalte aktivieren, stellen Sie sicher, dass Sie über Folgendes verfügen:

- Ihr VOD-Inhalt enthält ordnungsgemäß formatierte Werbemarkierungen (SCTE-35 oder zeitgesteuerte Metadaten)
- Die Inhalte werden an einem zuverlässigen Ort mit konsistenter Verfügbarkeit gespeichert
- Ihr CDN ist so konfiguriert, dass SGAI-Manifeste mit entsprechenden TTL-Werten zwischengespeichert werden
- Player unterstützen servergestützte Workflows zur Anzeigeneinfügung
- Ihr Anzeigenentscheidungsserver kann VOD-spezifische Metadaten und Targeting verarbeiten

Anforderungen an den Spieler

Die Player müssen so konfiguriert sein, dass sie SGAI VOD-Manifeste und das Einfügen von Werbung verarbeiten können:

- Support für servergesteuerte Workflows zur Anzeigeneinfügung
- Fähigkeit, Anleitungen zur Anzeigeneinfügung anhand von VOD-Manifesten zu verarbeiten
- Support für die clientseitige Anzeigeneinfügung während der VOD-Wiedergabe
- Richtiger Umgang mit Suchvorgängen bei Werbeunterbrechungen
- Support für Inhaltsdauer und Positionsverfolgung

Konfiguration der VOD-Wiedergabe

Um SGAI für VOD-Inhalte zu aktivieren, erstellen Sie eine Wiedergabekonfiguration mit den folgenden Einstellungen:

Example SGAI VOD-Wiedergabekonfiguration

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

Im Folgenden sind die wichtigsten Überlegungen zur VOD-SGAI-Konfiguration aufgeführt:

VideoContentSourceUrl

Sollte auf Ihre VOD-Inhaltsbibliothek mit konsistenten URL-Mustern verweisen. Stellen Sie sicher, dass der Absender das erwartete Anforderungsvolumen bewältigen kann und eine zuverlässige Bereitstellung der Inhalte gewährleistet.

ConfigurationAliases

Fügen Sie VOD-spezifische Parameter wie Inhaltsdauer, Genre oder Serieninformationen hinzu, die für die Anzeigenausrichtung verwendet werden können, ohne die Cachefähigkeit der Manifeste zu beeinträchtigen.

ManifestProcessingRules

Aktivieren Sie das Passthrough von Werbemarkern, um die ursprünglichen Informationen zum Zeitpunkt der Inhalte beizubehalten. Dies ist besonders wichtig für VOD-Inhalte mit vordefinierten Positionen für Werbeunterbrechungen.

Anfragen an das SGAI-VOD-Manifest

SGAI VOD-Manifeste verwenden dasselbe URL-Muster wie beim Einfügen von VOD-Anzeigen.

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

VOD-spezifisches Anzeigen-Targeting

VOD-Inhalte ermöglichen einzigartige Möglichkeiten zur gezielten Anzeigenausrichtung:

Ausrichtung auf Inhaltsmetadaten

Nutzen Sie Metadaten von VOD-Inhalten für eine verbesserte Anzeigenausrichtung:

- Genre und Kategorie: Richten Sie Anzeigen auf der Grundlage des Inhaltstyps aus (Drama, Komödie, Dokumentarfilm)
- Inhaltsbewertung: Sorgen Sie für altersgerechte Anzeigeninhalte (G-, PG-, R-Bewertungen)
- Serie und Staffel: Richten Sie Anzeigen gezielt auf Serienkontinuität oder verwandte Inhalte aus
- Veröffentlichungsdatum: Die Ausrichtung basiert auf dem Alter der Inhalte (Neuerscheinungen im Vergleich zu Kataloginhalten)
- Inhaltsdauer: Passen Sie die Anzeigenauslastung an die Gesamtlänge des Inhalts an

Kontext-Targeting anzeigen

VOD-Betrachtungsmuster ermöglichen eine kontextbezogene Anzeigenausrichtung:

- Tageszeit: Richten Sie Anzeigen danach aus, wann Inhalte angesehen werden
- Binge-Watching: Passen Sie die Anzeigenfrequenz für Nutzer an, die sich mehrere Folgen ansehen

- **Abschlussrate:** Das Ziel basiert auf den Vervollständigungsmustern der Nutzer in der Vergangenheit
- **Gerätetyp:** Optimieren Sie die Anzeigenformate für das Wiedergabegerät (Fernseher, Handy, Tablet)

Testen der SGAI VOD-Konfiguration

Überprüfen Sie Ihr SGAI VOD-Setup mit diesen Validierungsschritten:

1. Generierung von Testmanifesten

Fordern Sie das SGAI-VOD-Manifest URLs für verschiedene Inhaltstypen an und stellen Sie sicher, dass zwischenspeicherbare Inhalte mit den richtigen Anweisungen zum Einfügen von Anzeigen zurückgegeben werden.

2. Überprüfen Sie das CDN-Caching

Stellen Sie sicher, dass Ihr CDN SGAI-Manifeste gemäß den konfigurierten TTL-Werten zwischenspeichert und hohe Cache-Trefferraten erzielt.

3. Testen Sie die Anzeigeneinfügung

Vergewissere dich, dass Spieler erfolgreich Anzeigen einfügen können, indem du die Anleitungen in den SGAI-Manifesten für verschiedene VOD-Inhalte verwendest.

4. Testen Sie Suchvorgänge

Stellen Sie sicher, dass die Suche innerhalb von VOD-Inhalten über Werbeunterbrechungen hinweg korrekt funktioniert und dass die richtige Wiedergabeposition beibehalten wird.

5. Überwachen Sie die Leistung

Verwenden Sie CloudWatch Metriken, um die geringere Ausgangslast, verbesserte Cache-Trefferquoten und erfolgreiche Anzeigeneinblendungsraten zu überprüfen.

Die wichtigsten Testszenarien

Testen Sie diese spezifischen VOD-Szenarien:

- **Beliebte Inhalte:** Überprüfen Sie die hohen Cache-Trefferraten für häufig aufgerufene VOD-Ressourcen

- **Langform-Inhalt:** Testen Sie die Anzeigenschaltung in Filmen oder langen Folgen mit mehreren Werbeunterbrechungen
- **Serieninhalt:** Stellen Sie sicher, dass die Anzeigenausrichtung für alle Folgen einer Serie einheitlich ist
- **Verschiedene Genres:** Testen Sie die Anzeigenausrichtung anhand von Inhaltsmetadaten und Genres

Bewährte Methoden zur VOD-SGAI-Optimierung

Optimieren Sie Ihre SGAI VOD-Implementierung für maximale Leistung:

Cache-Optimierung

- **TTL-Werte maximieren:** Verwenden Sie längere Cashedauern für VOD-Manifeste, da sich der Inhalt nicht ändert
- **Cache-Schlüssel minimieren:** Reduzieren Sie die Variationen der Cache-Schlüssel, um die Trefferquoten zu erhöhen
- **Vorwärmen beliebter Inhalte:** Cache-Manifeste für beliebte oder beliebte VOD-Inhalte
- **Überwachen Sie die Cache-Leistung:** Verfolgen Sie die Cache-Trefferquoten und optimieren Sie sie anhand von Nutzungsmustern

Optimierung der Inhaltsbereitstellung

- **Konsistente URL-Muster:** Verwenden Sie vorhersehbare URL-Strukturen für ein besseres Caching
- **Standardisierung von Metadaten:** Sorgen Sie für konsistente Inhaltsmetadaten für eine zuverlässige Anzeigenausrichtung
- **Positionierung von Werbeunterbrechungen:** Optimieren Sie die Platzierung von Werbeunterbrechungen für natürliche Inhaltsübergänge
- **Qualitätsvarianten:** Stellen Sie sicher, dass SGAI für alle Bitratenvarianten Ihrer VOD-Inhalte funktioniert

MediaTailor Anforderungen an die Ad-Server-Integration

Um Ihren Ad-Server zu integrieren AWS Elemental MediaTailor, muss Ihr Ad-Server XML senden, das den IAB-Spezifikationen für die unterstützten Versionen von VAST und VMAP entspricht. Sie

können mithilfe einer öffentlichen VAST-Validierung sicherstellen, dass Ihre Tags ein gültiges Format aufweisen.

AWS Elemental MediaTailor unterstützt VAST- und VMAP-Antworten von Anzeigenentscheidungsservern. AWS Elemental MediaTailor unterstützt auch die Weiterleitung von VPAID-Metadaten über unsere clientseitige Berichts-API für die clientseitige Anzeigeneinblendung. Weitere Informationen zur clientseitigen Berichterstellung finden Sie unter [Kundenseitiges Anzeigen-Tracking](#).

MediaTailor unterstützt die folgenden Versionen von VAST, VMAP und VPAID:

- [Bis zu VAST 4.3](#)

MediaTailor akzeptiert Antwortversionen bis VAST 4.3, aber einige erweiterte Funktionen ab VAST 4.0 werden nicht unterstützt.

- [VMAP 1.0](#)

- [VPAID 2.0](#)

RIESIGE Anforderungen

Die VAST-Antwort Ihres Webservers muss IAB-konforme TrackingEvents-Elemente und Standard-Ereignistypen, wie z. B. impression, enthalten. Wenn Sie nicht standardmäßige Tracking-Ereignisse einbeziehen, AWS Elemental MediaTailor lehnt die VAST-Antwort ab und stellt keine Werbeanzeige für den Erfolg bereit.

Mit VAST 3.0 wurde die Unterstützung von Werbe-Pods eingeführt. Dabei handelt es sich um die Bereitstellung einer Reihe von sequenziellen linearen Werbespots. Wenn eine bestimmte Anzeige in einem Ad-Pod nicht verfügbar ist, AWS Elemental MediaTailor wird im Interaktionsprotokoll des ADS ein Fehler protokolliert. CloudWatch Anschließend versucht es, die nächste Werbung im Pod einzufügen. Auf diese Weise werden die Anzeigen im Pod MediaTailor wiederholt, bis eine gefunden wird, die verwendet werden kann.

Targeting

Um bestimmte Spieler für Ihre Anzeigen anzusprechen, können Sie Vorlagen für Ihre Anzeigen-Tags und URLs erstellen. Weitere Informationen finden Sie unter [MediaTailor dynamische Anzeigenvariablen für ADS-Anfragen](#).

AWS Elemental MediaTailor leitet die Werte `user-agent` und `x-forwarded-for` Header des Players weiter, wenn er die VAST-Anfrage des Anzeigenservers sendet und wenn er serverseitige Tracking-Aufrufe tätigt. Stellen Sie sicher, dass Ihre Werbeserver diese Header verarbeiten können. Alternativ können Sie `[session.user_agent]` oder `[session.client_ip]` verwenden und diese Werte in Abfragezeichenfolgen für das Werbe-Tag und die Werbe-URL übergeben. Weitere Informationen finden Sie unter [MediaTailor Sitzungsvariablen für ADS-Anfragen](#).

Ad-Aufrufe

AWS Elemental MediaTailor ruft Ihre VAST-Anzeigen-URL wie in Ihrer Konfiguration definiert auf. Sie ersetzt beim Anzeigenaufruf alle spieler- oder sitzungsspezifischen Parameter. MediaTailor folgt bis zu sieben Ebenen von VAST-Wrappern und leitet in der VAST-Antwort um. Führt in Live-Streaming-Szenarien zu Beginn der Werbeanzeige für verbundene Spieler gleichzeitig Werbeanrufe durch. MediaTailor In der Praxis können sich diese Werbeaufrufe aufgrund von Jitter über mehrere Sekunden erstrecken. Stellen Sie sicher, dass Ihr Anzeigenserver die Anzahl gleichzeitiger Verbindungen verarbeiten kann, die für diese Art von Anrufen erforderlich sind. MediaTailor unterstützt das Vorabrufen von VAST-Antworten für Live-Workflows. Weitere Informationen finden Sie unter [Anzeigen vorab abrufen](#).

Kreativer Umgang

Beim AWS Elemental MediaTailor Empfang der ADS VAST-Antwort wird für jedes Creative die höchste Bitrate MediaFile für die Transcodierung ermittelt und diese als Quelle verwendet. Diese Datei wird an den on-the-fly Transcoder gesendet, wo sie in Wiedergabeversionen umgewandelt wird, die den Bitraten und Auflösungen der multivarianten Playlist des Players entsprechen. Um optimale Ergebnisse zu erzielen, sollten Sie sicherstellen, dass es sich bei Ihrer Mediendatei mit der höchsten Bitrate um ein qualitativ hochwertiges Asset mit gültigen Manifest-Voreinstellungen handelt. MP4 Wenn die Manifest-Voreinstellungen nicht gültig sind, schlagen die transcodierten Aufträge fehl und wird keine Werbung angezeigt. Zu den ungültigen Voreinstellungen gehören beispielsweise nicht unterstützte Eingabedateiformate und bestimmte Wiedergabespezifikationen wie ProRes die Auflösung 855X481.

Eine Liste der unterstützten Formate für Mediendateieingaben finden Sie in der MP4Zeile [Unterstützte Eingabeformate](#) im Benutzerhandbuch.AWS Elemental MediaConvert

Kreative Indizierung

AWS Elemental MediaTailor indiziert jedes Creative eindeutig anhand des Werts des im Element angegebenen `id` Attributs. `<Creative>` Wenn die ID eines Kreativen nicht angegeben ist, wird die URL der Mediendatei für den Index MediaTailor verwendet.

Die folgende Beispieldeklaration zeigt die Werbemittel-ID.

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

Wenn Sie Ihr eigenes Creative definieren IDs, verwenden Sie für jedes Creative eine neue, eindeutige ID. Kreatives Material nicht wiederverwenden IDs. AWS Elemental MediaTailor speichert kreative Inhalte zur wiederholten Verwendung und findet sie anhand ihrer indizierten ID. Wenn ein neues Werbemittel eingeht, überprüft der Service zunächst den Index auf seine ID. Wenn die ID vorhanden ist, verwendet MediaTailor die gespeicherten Inhalte, ohne die eingehenden Inhalte erneut zu verarbeiten. Wenn Sie eine Creative-ID wiederverwenden MediaTailor, die ältere, gespeicherte Anzeige verwenden und Ihre neue Anzeige nicht wiedergeben.

VAST-Erweiterungen, die von Ad-Serving-Partnern bereitgestellt werden

Um Kollisionen mit Werbematerial zu vermeiden IDs, können Sie Erweiterungen verwenden, die von Werbepartnern für die VAST-Antwort bereitgestellt werden. MediaTailor unterstützt Erweiterungen von SpringServe, Publica und. FreeWheel Wenn Sie VAST-Erweiterungsüberschreibungen aktivieren, wird die standardmäßige Creative-ID durch den Erweiterungswert MediaTailor ersetzt.

Um diese Funktion zu aktivieren, [reichen Sie ein AWS Support-Ticket ein](#) und beantragen Sie, dass auf VAST-Erweiterungen basierende IDs Werbemittel aktiviert werden. Nehmen Sie die folgenden Informationen in das Support-Ticket auf:

- AWS Region
- AWS Konto-ID
- MediaTailor Namen der Wiedergabekonfigurationen

Um zu überprüfen, ob Creative, die auf VAST-Erweiterungen basieren, in Ihrem Konto aktiviert IDs sind, empfehlen wir Ihnen, auch die RAW_ADS_RESPONSE Protokollierung für eine Staging- oder Testkonfiguration für die Wiedergabe zu aktivieren. Mithilfe der Protokollierung können Sie sich die ursprüngliche VAST-Antwort ansehen, die das ADS erhält, und sich vergewissern, dass die richtigen Werbemittel IDs verwendet wurden.

VPAID-Anforderungen

Mithilfe von VPAID können Herausgeber äußerst interaktive Werbevideos aufnehmen und Sichtbarkeitsmetriken für ihre monetarisierte Streams bereitstellen. Informationen zu VPAID finden Sie in der [VPAID-Spezifikation](#).

AWS Elemental MediaTailor unterstützt eine Mischung aus MP4 linearen server-side-stitched VAST-Anzeigen und interaktiven client-side-inserted VPAID-Kreativen in derselben Anzeigenversion. Dabei wird die Reihenfolge beibehalten, in der sie in der VAST-Antwort erscheinen. MediaTailor folgt VPAID-Weiterleitungen durch maximal sieben Ebenen von Wrappern. Die Antwort des clientseitigen Berichts enthält die entpackten VPAID-Metadaten.

Beachten Sie zur Verwendung von VPAID die folgenden Richtlinien:

- Konfigurieren Sie einen MP4 Slate für Ihre VPAID-Creatives. AWS Elemental MediaTailor füllt die VPAID-Anzeigenfelder mit Ihrem konfigurierten Slate und stellt dem Kundenplayer VPAID-Anzeigenmetadaten zur Verfügung, die er zur Schaltung der interaktiven Anzeigen verwenden kann. Wenn bei Erscheinen einer VPAID-Werbung kein konfigurierter Pausenfüller vorhanden ist, stellt MediaTailor die Werbe-Metadaten wie üblich über die clientseitige Berichtserstellung bereit. Außerdem wird eine Fehlermeldung CloudWatch bezüglich des fehlenden Slate protokolliert. Weitere Informationen erhalten Sie unter [MediaTailor Schiefer und Einfügung](#) und [Eine MediaTailor Wiedergabekonfiguration erstellen](#).
- Verwenden Sie die clientseitige Berichterstattung. AWS Elemental MediaTailor unterstützt VPAID über unsere clientseitige Berichts-API. Weitere Informationen finden Sie unter [Kundenseitiges Anzeigen-Tracking](#).

Es ist theoretisch auch möglich, den serverseitigen Standard-Berichterstellungsmodus mit VPAID zu verwenden. Wenn Sie die serverseitige Berichterstellung verwenden, verlieren Sie jedoch alle Informationen über das Vorhandensein der VPAID-Werbung und der zugehörigen Metadaten, da diese Informationen nur über die clientseitige API verfügbar sind.

- Stellen Sie in Live-Szenarien sicher, dass Ihre mit gekennzeichneten Werbeanzeigen lang genug sind **EXT-X-CUE-OUT: Duration**, um jegliche Nutzerinteraktivität auf VPAID zu ermöglichen. Wenn das VAST-XML beispielsweise eine VPAID-Anzeige mit einer Länge von 30 Sekunden angibt, sollten Sie erwägen, Ihre Anzeigenverfügbarkeit auf mehr als 30 Sekunden zu konfigurieren. Diese zusätzliche Zeit gibt Nutzern mehr Möglichkeiten, mit der Anzeige zu interagieren. Wenn Sie keine Zeit hinzufügen, könnten Sie die VPAID-Metadaten verlieren, da die verbleibende Dauer der Anzeigenverfügbarkeit nicht lang genug ist, um die VPAID-Anzeige aufzunehmen.

MediaTailor Verwaltung der Wiedergabekonfiguration

In diesem Abschnitt werden die wichtigsten Aufgaben zur Verwaltung von AWS Elemental MediaTailor Wiedergabekonfigurationen behandelt. Sie erfahren, wie Sie eine neue Konfiguration erstellen, um Inhaltsstreams einzurichten und den Zugriff für Wiedergabegeräte zu ermöglichen, Details einer vorhandenen Konfiguration anzuzeigen, eine Konfiguration zu bearbeiten, um Einstellungen wie Ursprungsserver und Anzeigenentscheidungsserver zu aktualisieren, und eine Konfiguration zu löschen, die nicht mehr benötigt wird.

Themen

- [Eine MediaTailor Wiedergabekonfiguration erstellen](#)
- [MediaTailor Konfigurationsdetails anzeigen](#)
- [MediaTailor Konfigurationseinstellungen bearbeiten](#)
- [MediaTailor Konfigurationen werden gelöscht](#)

Eine MediaTailor Wiedergabekonfiguration erstellen

In diesem Thema wird gezeigt, wie Sie eine Konfiguration erstellen, mit der Sie beginnen können, Inhaltsstreams zu empfangen AWS Elemental MediaTailor. Außerdem wird gezeigt, wie ein Zugriffspunkt für nachgeschaltete Wiedergabegeräte bereitgestellt wird, um Inhalte anzufordern.

Sie können die AWS Elemental MediaTailor Konsole, AWS Command Line Interface (AWS CLI) > oder die MediaTailor API verwenden, um eine Konfiguration zu erstellen. Informationen zum Erstellen einer Konfiguration über die MediaTailor API AWS CLI oder finden Sie in der [AWS Elemental MediaTailor API-Referenz](#).

Wenn Sie eine Konfiguration erstellen, geben Sie keine vertraulichen Identifikationsinformationen in Freiformfelder wie das Feld für den Konfigurationsnamen ein. Zu den identifizierenden Informationen können Dinge wie Kundenkontonummern gehören. Verwenden Sie außerdem keine identifizierenden Informationen, wenn Sie in der MediaTailor Konsole, der REST-API AWS CLI, dem oder arbeiten AWS SDKs. Alle Daten, die Sie eingeben, werden MediaTailor möglicherweise für die Aufnahme in Diagnoseprotokolle oder Amazon CloudWatch Events aufgenommen.

So fügen Sie eine Konfiguration hinzu (Konsole)

1. Öffnen Sie die MediaTailor Konsole unter <https://console.aws.amazon.com/mediatailor/>.

2. Wählen Sie auf der Seite Configurations (Konfigurationen) die Option Create configuration (Konfiguration erstellen).
3. Füllen Sie die Konfigurationsfelder und zusätzlichen Konfigurationsfelder wie in den folgenden Themen beschrieben aus:
 - [Erforderliche Einstellungen](#)
 - [Optionale Konfigurationseinstellungen](#)
4. Wählen Sie Create configuration (Konfiguration erstellen).

AWS Elemental MediaTailor zeigt die neue Konfiguration in der Tabelle auf der Konfigurationsseite an.

5. (Empfohlen) Richten Sie ein CDN AWS Elemental MediaTailor für Manifest- und Berichtsanforderungen ein. Sie können die Konfigurationswiedergabe URLs für das CDN-Setup verwenden. Weitere Informationen zum Einrichten eines CDN für Manifest- und Berichterstellungs-Anforderungen finden Sie unter [Verwendung eines CDN zur Optimierung der MediaTailor Anzeigenpersonalisierung und Inhaltsbereitstellung](#).

Erforderliche Einstellungen

Wenn Sie eine Konfiguration erstellen, müssen Sie die folgenden erforderlichen Einstellungen angeben.

Name

Geben Sie einen eindeutigen Namen ein, der die Konfiguration beschreibt. Der Name ist der primäre Kennung für die Konfiguration. Die maximale zulässige Länge beträgt 512 Zeichen.

Quelle des Inhalts

Geben Sie das URL-Präfix für das Manifest für diesen Stream ein, ohne die Asset-ID. Die maximale Länge beträgt 512 Zeichen.

Das URL-Präfix `http://origin-server.com/a/` ist beispielsweise gültig für eine multivariante HLS-Playlist-URL von `http://origin-server.com/a/main.m3u8` und für eine DASH-MPD-URL von `http://origin-server.com/a/dash.mpd`. Alternativ können Sie ein kürzeres Präfix wie z. B. `http://origin-server.com` eingeben, /a/ muss dann aber in die Komponenten-ID der Player-Anforderung für Inhalte eingeschlossen werden.

 Note

Wenn der Ursprungs-Server Ihrer Inhalte HTTPS verwendet, muss sein Zertifikat von einer bekannten Zertifizierungsstelle stammen. Es kann kein selbstsigniertes Zertifikat sein. Wenn Sie ein selbstsigniertes Zertifikat verwenden, AWS Elemental MediaTailor schlägt die Verbindung zum Inhaltsursprung fehl und Sie können keine Manifeste als Antwort auf Spieleranfragen bereitstellen.

Und Entscheidungsserver

Geben Sie die URL für Ihren Ad Decision Server (ADS) ein. Dies ist entweder die unter [Schritt 3: Konfigurieren Sie die ADS-Anforderungs-URL und die Abfrageparameter](#) beschriebene URL mit Variablen oder die statische VAST-URL, die Sie für Testzwecke verwenden. Die maximale Länge beträgt 25.000 Zeichen.

 Note

Wenn der ADS HTTPS verwendet, muss sein Zertifikat von einer bekannten Zertifizierungsstelle stammen. Es kann kein selbstsigniertes Zertifikat sein. Das Gleiche gilt auch für Mezzanine-Anzeigen, die vom ADS URLs zurückgegeben werden. Wenn Sie ein selbstsigniertes Zertifikat verwenden, AWS Elemental MediaTailor können Sie keine Anzeigen vom Ursprung des Inhalts abrufen und in die Manifeste einbinden.

Optionale Konfigurationseinstellungen

Sie können optional Konfigurationsalias, Personalisierungsdetails und erweiterte Einstellungen in der MediaTailor Konsole, der MediaTailor API oder der AWS Command Line Interface () konfigurieren. AWS CLI

Aliase für die Konfiguration

Im Folgenden finden Sie optionale Konfigurationsalias, die Sie in der MediaTailor Konsole oder mit der MediaTailor API konfigurieren können.

Variable für den Spielerparameter

Fügen Sie für die dynamische Domänenkonfiguration während der Sitzungsinitialisierung eine oder mehrere Player-Parametervariablen hinzu.

Weitere Hinweise zur Verwendung von Player-Parametervariablen zur dynamischen Konfiguration von Domänen finden Sie unter [MediaTailor Domain-Variablen für mehrere Inhaltsquellen](#).

Konfiguration protokollieren

Im Folgenden sind die Einstellungen für die Protokollkonfiguration aufgeführt.

Prozent aktiviert

Legt den Prozentsatz der Sitzungsprotokolle zur Wiedergabekonfiguration fest, der in CloudWatch Protokolle MediaTailor geschrieben wird. Wenn Ihre Wiedergabekonfiguration beispielsweise 1000 Sitzungen umfasst und Sie die Prozentzahl auf 60 gesetzt haben, werden 600 Sitzungsprotokolle in CloudWatch Logs MediaTailor geschrieben.

Wenn Sie diese Option aktivieren, MediaTailor wird automatisch eine dienstbezogene Rolle erstellt, mit der Sie Sitzungsprotokolle in Ihrem CloudWatch Logs-Konto schreiben und verwalten können MediaTailor . Weitere Informationen finden Sie unter [Verwenden von serviceverknüpften Rollen für MediaTailor](#).

Strategien für die Protokollierung

Gibt die Methode an, die zum Sammeln von ausgegebenen Protokollen verwendet wird. MediaTailor Um Protokolle direkt an CloudWatch Logs zu senden, wählen Sie `LEGACY_CLOUDWATCH`. Um Logs an CloudWatch Logs zu senden, das die Logs dann an das Ziel Ihrer Wahl weiterleitet, wählen Sie `VENDED_LOGS`. Unterstützte Ziele sind eine CloudWatch Logs-Protokollgruppe, ein Amazon S3 S3-Bucket und ein Amazon Data Firehose-Stream.

Für verkaufte Protokolle ist eine zusätzliche Einrichtung erforderlich. Informationen zur Einrichtung finden Sie unter [Verwendung von verkauften Protokollen](#).

Opt-in-Ereignisse für das ADS-Interaktionsprotokoll

Gibt an, dass MediaTailor `RAW_ADS_RESPONSE` Protokolle für Sitzungen ausgegeben werden, die mit dieser Konfiguration initialisiert wurden.

Das `RAW_ADS_RESPONSE` Protokollereignis enthält die gesamte VAST- oder VMAP-Antwort des ADS. Daher können die Protokolle umfangreich sein und Ihre Protokollierungskosten erhöhen.

Das ADS-Interaktionsprotokoll schließt Ereignisse aus

Zeigt an, dass die ausgewählten Ereignisse in den Protokollen, die Interaktionen mit dem ADS beschreiben, MediaTailor nicht ausgegeben werden.

Eine Beschreibung der ADS-Protokollereignisse finden Sie unter [ADS-Protokolle](#).

Manifestes Dienstinteraktionsprotokoll schließt Ereignisse aus

Gibt an, dass die ausgewählten Ereignisse in den Protokollen, die Interaktionen mit dem Manifestdienst beschreiben, MediaTailor nicht ausgegeben werden.

Eine Beschreibung der Manifest-Dienstprotokollereignisse finden Sie unter [Manifest-Protokolle](#).

Und Konditionierung

Im Folgenden wird festgelegt, welche Maßnahmen ergriffen MediaTailor werden, um Anzeigen zu konditionieren, bevor sie in einen Inhaltsstream eingefügt werden.

Konditionierung von Streaming-Mediendateien

Legt die Logik fest, nach der entschieden MediaTailor wird, welche Anzeigen zusammengefügt werden sollen.

- Wenn die Konditionierung von Streaming-Mediendateien auf Transkodieren gesetzt ist, werden die Mediendateien bei der progressive Auslieferung MediaTailor transkodiert und in das Manifest zusammengefügt. Wenn nicht genügend Mediendateien für Werbeanzeigen mit progressive Versand zur Verfügung stehen, werden diese Dateien MediaTailor transkodiert und bei Lieferung verwendet. `streaming`
- Wenn die Konditionierung von Streaming-Mediendateien auf „Keine“ gesetzt ist, werden Anzeigen mit MediaTailor `streaming` Übertragungsmediendateien in das Manifest zusammengefügt, ohne sie zu transkodieren. Wenn nicht genügend Werbeanzeigen mit Mediendateien zur `streaming` Auslieferung vorhanden sind, werden diese Dateien MediaTailor transkodiert und bei der Auslieferung verwendet. `progressive`

Einzelheiten zur Personalisierung

Im Folgenden finden Sie Personalisierungsdetails, die Sie in der MediaTailor Konsole oder mit der MediaTailor API konfigurieren können.

Slate-Werbung

Geben Sie die URL für ein qualitativ hochwertiges MP4 Asset ein, das MediaTailor verwendet wird, um ungenutzte Werbezeiten zu füllen. Die Slate-Konfiguration ist für die meisten Workflows optional, für VPAID-Anzeigen jedoch obligatorisch. Umfassende Informationen zum Verhalten

von Slate, zu den Konfigurationsanforderungen und wann Slate angezeigt wird, finden Sie unter [MediaTailor Schiefer und Einfügung](#).

Stoßstange starten

Die URL des Speicherorts des Start-Bumper-Assets. Bumper sind kurze Video- oder Audioclips, die zu Beginn oder am Ende einer Werbepause abgespielt werden. Sie können auf Amazon S3 oder einem anderen Speicherdienst gespeichert werden. Weitere Informationen über Stoßstangen finden Sie unter [MediaTailor Stoßstange und Einlage](#).

Endstoßstange

Die URL des Asset-Standorts am Ende der Stoßstange. Bumper sind kurze Video- oder Audioclips, die zu Beginn oder am Ende einer Werbepause abgespielt werden. Sie können auf Amazon S3 oder einem anderen Speicherdienst gespeichert werden. Weitere Informationen über Stoßstangen finden Sie unter [MediaTailor Stoßstange und Einlage](#).

Personalisierungsschwelle

Gilt nur in Verbindung mit Slate Ad. Definiert die maximale Dauer der unterfüllten Werbezeit (in Sekunden), die in einer Werbepause zulässig ist, bevor die Personalisierung MediaTailor abgebrochen wird und der zugrunde liegende Inhalt eingeblendet wird. Diese Funktion gilt für das Ersetzen von Inhalten durch Werbung in Live- und VOD-Streams und nicht für das Einblenden von Werbung, da dies auf einem zugrunde liegenden Inhaltsstream beruht. Ausführliche Verhaltensszenarien und umfassende Informationen finden Sie unter [So funktioniert der Schwellenwert für Personalisierung](#)

Live-Pre-Roll-Werbeentscheidungsserver

Um Werbung am Anfang eines Live-Streams einzufügen, bevor der Hauptinhalt mit der Wiedergabe beginnt, geben Sie die URL für das Ad-Preroll vom Ad Decision Server (ADS) ein. Dies ist entweder die unter [Schritt 3: Konfigurieren Sie die ADS-Anforderungs-URL und die Abfrageparameter](#) beschriebene URL mit Variablen oder die statische VAST-URL, die Sie für Testzwecke verwenden. Die maximale Länge beträgt 25.000 Zeichen.

Note

Wenn der ADS HTTPS verwendet, muss sein Zertifikat von einer bekannten Zertifizierungsstelle stammen. Es kann kein selbstsigniertes Zertifikat sein. Das Gleiche gilt auch für Mezzanine-Anzeigen, die vom ADS URLs zurückgegeben werden. Wenn Sie

ein selbstsigniertes Zertifikat verwenden, AWS Elemental MediaTailor können Sie keine Anzeigen vom Ursprung des Inhalts abrufen und in die Manifeste einbinden.

Informationen zur Funktionsweise von Pre-Roll finden Sie unter [MediaTailor Einfügen von Pre-Roll-Anzeigen](#).

Maximal zulässige Live-Pre-Roll-Dauer

Wenn Sie Werbung am Anfang eines Livestreams einfügen, geben Sie die maximal zulässige Dauer für die verfügbaren Pre-Roll-Werbung ein. MediaTailor wird diese Dauer beim Einfügen von Werbung nicht überschreiten. Wenn die Antwort aus dem ADS mehr Anzeigen enthält, als in diese Dauer passen, wird das MediaTailor Angebot mit so vielen Anzeigen wie möglich aufgefüllt, ohne die Dauer zu überschreiten. Weitere Informationen darüber, wie MediaTailor Fills verfügbar sind, finden Sie unter [Verhalten beim Zusammenfügen von Live-Anzeigen](#)

Verwenden Sie den Unterdrückungsmodus

Legt den Modus für die Werbeplatzunterdrückung fest, wird auch als Werbeunterdrückung bezeichnet. Standardmäßig ist die Anzeigenunterdrückung ausgeschaltet und alles wird mit Werbung oder Slate MediaTailor gefüllt. Wenn der Modus auf eingestellt ist `BEHIND_LIVE_EDGE`, ist die Anzeigenunterdrückung aktiv und füllt MediaTailor keine Werbeunterbrechungen, die an oder hinter dem Wert für die Verfügbarkeitsunterdrückung im Manifest-Lookback-Fenster liegen. Wenn der Modus auf eingestellt ist `AFTER_LIVE_EDGE`, ist die Anzeigenunterdrückung aktiv. MediaTailor füllt keine Werbeunterbrechungen auf oder hinter dem Zeitraum der Verfügbarkeitsunterbindung, der sich aus dem Live-Rand plus dem Wert für die Verfügbarkeitsunterbindung plus Pufferzeit zusammensetzt.

Nutzt den Wert für die Unterdrückung

Der Wert für die Avail-Unterdrückung entspricht der Offsetzeit der Live-Kante in. HH:MM:SS MediaTailor Ich werde im Manifest-Lookback-Fenster keine Werbeunterbrechungen an oder hinter diesem Zeitpunkt auffüllen.

Einfügemodus

Der Einfügemodus steuert, ob Spieler das Einfügen von eingefügten oder geführten Anzeigen verwenden können. Die Standardeinstellung zwingt alle Spielsitzungen dazu `STITCHED_ONLY`, gestickte (serverseitige) Werbeanzeigen einzufügen. Bei der Einstellung `InsertionMode` auf `PLAYER_SELECT` können Spieler bei der Initialisierung der Sitzung zwischen gestickter oder geführter Anzeigeneinfügung wählen. Die Standardeinstellung für Player, die keinen Einfügemodus angeben, ist „Stitched“.

Erweiterte Einstellungen

Bei den folgenden optionalen Einstellungen handelt es sich um erweiterte Einstellungen. Sie können diese in der MediaTailor Konsole, mit der AWS Command Line Interface (AWS CLI) oder mithilfe der MediaTailor API konfigurieren.

CDN-Inhaltssegmentpräfix

Ermöglicht AWS Elemental MediaTailor die Erstellung von Manifesten mit URLs Ihrem CDN-Pfad für Inhaltssegmente. Bevor Sie diesen Schritt ausführen, richten Sie in Ihrem CDN eine Regel für den Abruf von Segmenten vom Ursprungs-Server ein. Geben Sie für CDN content segment prefix (CDN-Inhaltssegment-Präfix) den CDN-Präfixpfad ein.

Weitere Informationen zur Integration MediaTailor mit einem CDN finden Sie unter [Verwendung eines CDN zur Optimierung der MediaTailor Anzeigenpersonalisierung und Inhaltsbereitstellung](#)

CDN-Webesegmentpräfix

Ermöglicht AWS Elemental MediaTailor die Erstellung von Manifesten mit URLs Ihrem eigenen CDN-Pfad für Anzeigensegmente. Standardmäßig werden MediaTailor Anzeigensegmente aus einer internen CloudFront Amazon-Distribution mit Standard-Cache-Einstellungen bereitgestellt. Bevor Sie das Feld CDN ad segment prefix (CDN-Webesegment-Präfix) ausfüllen können, müssen Sie wie in dem folgenden Beispiel in Ihrem CDN eine Regel für den Abruf von Segmenten von dem folgenden Ursprungs-Server einrichten.

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

Geben Sie für CDN ad segment prefix (CDN-Webesegmentpräfix) den Namen Ihres CDN-Präfix in der Konfiguration ein.

Weitere Informationen zur Integration MediaTailor mit einem CDN finden Sie unter [Verwendung eines CDN zur Optimierung der MediaTailor Anzeigenpersonalisierung und Inhaltsbereitstellung](#).

DASH-Origin Manifesttyp

Wenn Ihr Ursprungsserver DASH-Manifeste mit einem einzigen Zeitraum produziert, öffnen Sie die Dropdown-Liste und wählen Sie SINGLE_PERIOD. Standardmäßig behandelt MediaTailor DASH-Manifeste wie Manifeste mit mehreren Zeiträumen. Weitere Informationen finden Sie unter [the section called “Integration einer MPEG-DASH-Quelle”](#).

DASH-mpd Speicherort

(Optional, je nach Bedarf für DASH) Der Speicherort der Beschreibung der Medienpräsentation (MPD). Wählen Sie DISABLED für die folgende Situation:

- Sie richten CDN-Routingregeln für den Zugriff auf MediaTailor Manifeste ein.
- Sie verwenden clientseitige Berichterstattung, oder Ihr Player unterstützt Sticky-HTTP-Weiterleitungen.

Weitere Informationen über die Funktion Location (Speicherort) finden Sie unter [the section called "Standort-Funktion"](#).

Transkodieren Sie den Profilnamen

Der Name, der diese Konfiguration einem benutzerdefinierten Transcodierungsprofil zuordnet. Dieser Name überschreibt die Standardeinstellungen für die dynamische Transcodierung von MediaTailor. Füllen Sie dieses Feld nur aus, wenn Sie mithilfe des AWS Supports bereits benutzerdefinierte Profile eingerichtet haben.

Passthrough für Werbemarkierungen

Aktiviert oder deaktiviert für HLS den Passthrough für Werbemarkierungen. Wenn der Passthrough für Werbemarkierungen aktiviert ist, werden MediaTailor EXT-X-CUE-IN, EXT-X-CUE-OUT, und EXT-X-SPLICEPOINT-SCDTE35 Anzeigenmarkierungen vom Ursprungsmanifest zum personalisierten Manifest weitergeleitet. MediaTailor wendet keine Logik auf die Werte der Werbemarkierung an. Sie werden unverändert vom Ursprungsmanifest an das personalisierte Manifest übergeben. Wenn es beispielsweise 60 im Ursprungsmanifest den Wert 0 EXT-X-CUE-OUT hat, aber keine Werbung geschaltet wird, wird der Wert 0 im personalisierten Manifest nicht geändert.

MediaTailor Konfigurationsdetails anzeigen

AWS Elemental MediaTailor zeigt zusätzlich zu den Werten, die bei der Erstellung der Konfiguration angegeben wurden, den Namen der Konfiguration, die Wiedergabeendpunkte und den entsprechenden Zugriff an. URLs Gehen Sie wie folgt vor, um eine Konfiguration anzuzeigen.

So zeigen Sie eine Konfiguration an

1. Öffnen Sie die MediaTailor Konsole unter <https://console.aws.amazon.com/mediatailor/>.

2. Wählen Sie auf der Seite Configurations (Konfigurationen) unter Configuration name (Konfigurationsname) den Namen der anzuzeigenden Konfiguration aus.

MediaTailor Konfigurationseinstellungen bearbeiten

Sie können eine Konfiguration bearbeiten, um die Zuordnung des Ursprungsservers und des Ad Decision Servers (ADS) zu aktualisieren oder die Art und Weise der AWS Elemental MediaTailor Interaktion mit einem Content Distribution Network (CDN) zu ändern.

So bearbeiten eine Konfiguration

1. Öffnen Sie die MediaTailor Konsole unter <https://console.aws.amazon.com/mediatailor/>
2. Wählen Sie auf der Seite Configurations (Konfigurationen) den Namen der Konfiguration aus, die Sie bearbeiten möchten.
3. Klicken Sie auf der Seite mit den Details auf Edit (Bearbeiten) und überarbeiten Sie dann die Konfigurationseinstellungen nach Bedarf. Der Konfigurationsname kann nicht bearbeitet werden. Weitere Informationen über Konfigurationsattribute finden Sie unter [Eine MediaTailor Wiedergabekonfiguration erstellen](#).
4. Wählen Sie Speichern.

MediaTailor Konfigurationen werden gelöscht

Sie können eine Konfiguration löschen, damit sie nicht mehr wiedergegeben werden kann AWS Elemental MediaTailor.

So löschen Sie eine Konfiguration

1. Öffnen Sie die MediaTailor Konsole unter <https://console.aws.amazon.com/mediatailor/>.
2. Führen Sie auf der Seite Configurations (Konfigurationen) einen der folgenden Schritte durch:
 - Wählen Sie den Namen der Konfiguration aus, die Sie löschen möchten.
 - Aktivieren Sie in der Spalte Configuration name (Konfigurationsname) das Optionsfeld neben dem Namen und wählen Sie dann Delete (Löschen).
3. Geben Sie im Bestätigungsdialegfeld Delete (Löschen) den Text **Delete** ein und wählen Sie dann Delete (Löschen) aus.

Integration einer Inhaltsquelle für die MediaTailor Anzeigeneinfügung

In diesem Thema wird die Integration verschiedener Arten von Videoinhaltsquellen mit behandelt MediaTailor. MediaTailorunterstützt sowohl HLS- als auch DASH-Streaming-Protokolle für Live- und On-Demand-Inhalte. Der Dienst kann das Einfügen oder Ersetzen von Anzeigen während bestimmter Werbepausen vornehmen und stellt spezielle Anforderungen an die Struktur und Formatierung der Eingabevideomanifeste, um diese Funktionen zu aktivieren. In den folgenden Themen finden Sie Einzelheiten zu den Anforderungen an die Eingabequelle und zu den Schritten für die Integration von HLS- und DASH-Inhalten MediaTailor , um personalisierte Werbeerlebnisse zu ermöglichen.

Themen

- [Anforderungen an die Eingabequelle für die MediaTailor Anzeigenschaltung](#)
- [Integration einer HLS-Quelle](#)
- [Integration einer MPEG-DASH-Quelle](#)
- [Sicherung der AWS Elemental MediaTailor ursprünglichen Interaktionen mit SigV4](#)

Anforderungen an die Eingabequelle für die MediaTailor Anzeigenschaltung

Eine Eingabequelle muss die folgenden Anforderungen erfüllen, um damit arbeiten zu können MediaTailor:

- Er muss Apple HLS (HTTP Live Streaming) oder MPEG DASH (Dynamic Adaptive Streaming über HTTP) verwenden.
- Er muss Live-Streaming oder Video-On-Demand (VOD) verwenden.
- Er muss über das öffentliche Internet zugänglich sein und über eine öffentliche IP-Adresse verfügen.
- Verwenden Sie Standard-HTTP-Ports (Port 80) oder HTTPS-Ports (Port 443). MediaTailor unterstützt keine benutzerdefinierten Ports für die Kommunikation mit dem Ursprungsserver.
- Enthält Werbemarkierungen in einem der Formate, die im [Tutorial Erste Schritte beim Einfügen von MediaTailor Anzeigen](#) beschrieben sind

Integration einer HLS-Quelle

AWS Elemental MediaTailor unterstützt .m3u8 HLS-Manifeste mit einem Wert EXT-X-VERSION von 3 oder höher für Live-Streaming und Video on Demand (VOD). Wenn MediaTailor auf eine Werbeunterbrechung gestoßen wird, wird je nach Art des Inhalts versucht, eine Anzeige einzufügen oder zu ersetzen. Wenn nicht genügend Anzeigen für die gesamte Dauer vorhanden sind, wird für den Rest der Werbeunterbrechung der zugrunde liegende Inhaltsstream oder der konfigurierte Slate MediaTailor angezeigt. Weitere Informationen zum Verhalten von HLS-Anzeigen je nach Inhaltstyp finden Sie unter [Das Verhalten bei der AWS Elemental MediaTailor Anzeigeneinblendung verstehen](#).

In den folgenden Abschnitten finden Sie weitere Informationen zum MediaTailor Umgang mit HLS-Manifesten.

Themen

- [Von HLS unterstützte Anzeigenmarkierungen](#)
- [Passthrough für Werbemarkierungen aktivieren](#)
- [Behandlung von HLS-Manifest-Tags](#)
- [Beispiele für HLS-Manifests](#)

Von HLS unterstützte Anzeigenmarkierungen

AWS Elemental MediaTailor identifiziert und nutzt Grenzen in einem HLS-Manifest, indem das Eingabemanifest nach unterstützten Anzeigenmarkierungen analysiert wird. Die folgenden Abschnitte beschreiben, welche Marker MediaTailor verwendet.

EXT-X-ANLAGE

Das EXT-X-ASSET-Tag enthält Metadaten, die vom Ad Decision Server (ADS) zum Personalisieren von Inhalten für die Anzeige verwendet werden. EXT-X-ASSET-Parameter sind durch Kommas getrennte Schlüssel-Wert-Paare.

Um diesen Tag verwenden zu können, müssen Sie die folgenden Anforderungen erfüllen:

- Sie müssen die EXT-X-ASSET-Werte im Ursprungsmanifest URL-kodieren. Es folgt das Beispiel eines EXT-X-ASSET-Tags mit Schlüsseln und URL-kodierten Werten.

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

- Sie müssen die dynamische [asset.] Variable und die Schlüssel in Ihre MediaTailor ADS-Konfiguration aufnehmen. Das folgende Beispiel zeigt eine MediaTailor ADS-Konfiguration mit der dynamischen [asset.] Variablen und den Schlüsseln.

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

Beispiel VAST-Anfrage

Es folgt das Beispiel einer VAST-GET-Anforderung an ein ADS.

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

EXT-X-CUE-OUT und EXT-X-CUE-IN

Diese Art von Werbe-Marker ist die geläufigste. Die folgenden Beispiele zeigen Optionen für diese Cue-Marker.

```
#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

Bei EXT-X-DATERANGE Werbemarkierungs-Tags verwenden Sie SCTE35-OUT-Attribute, um den Zeitpunkt der Werbeverfügbarkeit festzulegen.

Note

AWS Elemental MediaTailor ignoriert alle START-DATE Attribute, die für EXT-X-DATERANGE Anzeigenmarkierungen bereitgestellt werden.

Sie können den Ad-Avail auf eine der folgenden Arten angeben:

- EXT-X-DATERANGE-Tag mit SCTE35-OUT- und DURATION-Spezifikationen.

Beispiel

```
#EXT-X-DATERANGE:ID="splice-6FFFFFFF0",START-DATE="2019-01T00:15:00Z\n",DURATION=60.000,SCTE35-OUT=0xF
```

- Gepaarte EXT-X-DATERANGE-Tags, das erste mit einer SCTE35-OUT-Spezifikation und das zweite mit einer SCTE35-IN-Spezifikation.

Beispiel

```
#EXT-X-DATERANGE:ID="splice-6FFFFFFF0",START-DATE="2019-01T00:15:00Z\n",SCTE35-OUT=0xF
...
#EXT-X-DATERANGE:ID="splice-6FFFFFFF0",START-DATE="2019-01T00:15:00Z\n",SCTE35-IN=0xF
```

- Eine Kombination aus den vorherigen Optionen. Sie geben ein EXT-X-DATERANGE-Tag mit SCTE35-OUT- und DURATION-Spezifikationen gefolgt von einem EXT-X-DATERANGE-Tag mit einer SCTE35-IN-Spezifikation an. MediaTailor Verwendet in diesem Fall die früheste Cue-In-Einstellung aus den beiden Spezifikationen.

Beispiel

```
#EXT-X-DATERANGE:ID="splice-6FFFFFFF0",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\","SCTE35-
IN=0xF
```

EXT-X-SPLICEPOINT- SCTE35

Sie hängen das EXT-X-SPLICEPOINT-SCTE35-Werbemarker-Tag mit einer SCTE-35-Nutzlast im base64-kodierten binären Format an. Das dekodierte binäre Format muss einen SCTE-35-splice_info_section mit dem Cue-out-Marker 0x34 als Anfangspunkt der Platzierungsmöglichkeit für Anbieter und dem Cue-in-Marker 0x35 als Endpunkt der Platzierungsmöglichkeit für Anbieter enthalten.

Das folgende Beispiel zeigt die Splice-Punkt-Spezifikation mit base64-kodierten binären Nutzlasten, die die Cue-out- und Cue-in-Marker angeben.

```
#EXT-X-SPLICEPOINT-SCTE35:/DA9AAAAAAAAAAP/wBQb+uYbZqwAnAiVDVUVJAAAKqX//
AAEjW4AMEU1EU05CMDAxMTMyMjE5M190NAAAmXz5JA==
...
#EXT-X-SPLICEPOINT-SCTE35:/DA4AAAAAAAAAAP/wBQb+tTeaawAiAiBDVUVJAAAKqH+/
DBFNRFNOQjAwMTEzMjIx0TJfTjUAAIiGK1s=
```

Passthrough für Werbemarkierungen aktivieren

Standardmäßig enthalten MediaTailor personalisierte Manifeste für HLS nicht die SCTE-35-Werbemarkierungen aus den ursprünglichen Manifesten. Wenn das Passthrough für Werbemarkierungen aktiviert ist, MediaTailor werden die folgenden Werbemarkierungen aus den ursprünglichen Manifesten in personalisierte Manifeste weitergeleitet:

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

Der Passthrough für Werbemarkierungen ist eine optionale Einstellung. Verwenden Sie den Passthrough für Werbemarkierungen, wenn Sie möchten, dass die SCTE-Werbemarkierungen in das MediaTailor personalisierte Manifest aufgenommen werden. Zu den häufigsten Anwendungsfällen gehören die folgenden:

- Austausch von Inhalten — Führen Sie eine Inhaltsersetzung oder eine Inhaltsbeschränkung durch.
- Ad-Tracking — Geben Sie Informationen zur Anzeigenverfolgung auf Grundlage des Vorhandenseins oder Fehlens einer oder mehrerer Werbemarkierungen an.
- Player-Einstellungen — Aktiviere die Scrubbing- oder Countdown-Timer-Funktionen in der Benutzeroberfläche des Players, je nachdem, ob Werbemarkierungen vorhanden sind oder nicht.

Note

MediaTailor ändert die Werte für diese Markierungen nicht. Wenn EXT-X-CUE-OUT beispielsweise 60 im Ursprungsmanifest der Wert 0 steht, aber keine Werbung geschaltet MediaTailor wird, wird der Wert 0 im personalisierten Manifest nicht geändert.

Aktivieren Sie den Passthrough für Werbemarkierungen

Sie können den Passthrough für Werbemarkierungen mit dem AWS-Managementkonsole oder dem AWS Command Line Interface (AWS CLI) aktivieren.

So aktivieren Sie den Passthrough für Werbemarkierungen mithilfe der Konsole

1. Öffnen Sie die MediaTailor Konsole unter <https://console.aws.amazon.com/mediatailor/>
2. Wählen Sie entweder Neue Konfiguration oder Konfiguration bearbeiten aus.
3. Wählen Sie im Bereich Erweiterte Einstellungen im Dropdownmenü die Option Aktivieren aus.

Um den Passthrough für Werbemarkierungen mit dem AWS Command Line Interface (AWS CLI) zu aktivieren

Verwenden Sie den Befehl [put-playback-configuration](#).

Behandlung von HLS-Manifest-Tags

In diesem Abschnitt wird beschrieben, wie Tags im personalisierten Ausgabemanifest AWS Elemental MediaTailor verwaltet werden.

EXT-X-CUE Tags

MediaTailor ersetzt EXT-X-CUE-OUT, EXT-X-CUE-OUT-CONT, und EXT-X-CUE-IN -Tags im Eingabemanifest durch EXT-X-DISCONTINUITY Tags im Ausgabemanifest. Die DISCONTINUITY-Tags markieren die folgenden Grenzen:

- Wo der Hauptinhalte zu einer Werbung wechselt
- Übergang von einer Werbung zur anderen
- Wo eine Werbung zum Hauptinhalt zurückkehrt

EXT-X-DATERANGE-Tags

MediaTailor übergibt EXT-X-DATERANGE Tags vom Eingabemanifest zum Ausgabemanifest. MediaTailor fügt außerdem EXT-X-DISCONTINUITY Tags ein, die den DATERANGE Tags entsprechen. Die DISCONTINUITY-Tags markieren die folgenden Grenzen:

- Wo der Hauptinhalte zu einer Werbung wechselt
- Übergang von einer Werbung zur anderen
- Wo eine Werbung zum Hauptinhalt zurückkehrt

EXT-X-KEY-Tags

MediaTailor durchläuft EXT-X-KEY Tags aus dem Eingabemanifest. Diese Tags zeigen an, dass der Hauptinhalt verschlüsselt ist. Da Werbung nicht verschlüsselt sind, fügt MediaTailor EXT-X-KEY:METHOD=NONE zu Beginn einer Werbung ein. Wenn die Wiedergabe zum Hauptinhalt zurückkehrt, wird die Verschlüsselung MediaTailor wieder aktiviert, indem das EXT-X-KEY Tag mit dem als METHOD Verschlüsselungstyp definierten Wert eingefügt wird.

Unbekannte Tags

MediaTailor durchläuft alle unbekannten und benutzerdefinierten Tags vom Eingabemanifest zum Ausgabemanifest.

Beispiele für HLS-Manifests

Die folgenden Abschnitte enthalten Beispiele für HLS-Manifeste und personalisierte Manifeste. Wenn Sie diese Beispiele verstehen, können Sie Ihre MediaTailor Workflows konfigurieren und Fehler beheben.

Informationen darüber, wie Abfrageparameter auf HLS-Manifeste und -Segmente angewendet werden, finden Sie unter [MediaTailor Implizite HLS-Sitzungsinitialisierung](#).

Grundlegendes zu den Typen von HLS-Wiedergabelisten

HTTP Live Streaming (HLS) verwendet zwei Haupttypen von Playlisten:

Multivariante Wiedergabeliste

Eine multivariante Playlist ist die Indexdatei auf oberster Ebene, in der alle verfügbaren Wiedergabeversionen des Inhalts aufgeführt sind. Sie enthält Verweise auf Medien-Playlisten, enthält aber selbst keine Mediensegmente. Diese Playlist ermöglicht es Spielern, je nach Netzwerkbedingungen, Gerätefunktionen oder Benutzereinstellungen die am besten geeignete Wiedergabeversion auszuwählen.

Dieser Playlist-Typ ist in verschiedenen Kontexten auch unter verschiedenen anderen Namen bekannt, darunter Master-Playlist, Master-Manifest, primäre Playlist, Haupt-Playlist, Indexdatei oder Master-M3U8.

In MediaTailor Workflows ist die multivariante Playlist der Einstiegspunkt für Wiedergabeanfragen und der Ausgangspunkt für die Personalisierung von Anzeigen.

Medien-Playlist

Eine Medien-Playlist enthält die aktuellen Mediensegmentinformationen für eine bestimmte Wiedergabe (Qualitätsstufe) des Inhalts. Sie enthält Zeitinformationen, Segmente und andere Metadaten URLs, die für die Wiedergabe einer einzelnen Wiedergabe erforderlich sind.

Dieser Playlist-Typ wird auch als Medien-Playlist, Child-Manifest, Chunklist, Media-M3U8 oder Wiedergabe-Playlist bezeichnet.

In MediaTailor Workflows werden Medien-Playlisten so personalisiert, dass sie sowohl Inhaltssegmente als auch Anzeigensegmente in der richtigen Reihenfolge enthalten.

Ausführlichere Informationen zu HLS-Playlist-Typen finden Sie unter [Typen von HLS-Wiedergabelisten](#).

Beispiele für HLS-Ursprungsmanifeste

Das folgende Beispiel zeigt eine multivariante HLS-Wiedergabeliste, die HLS vom AWS Elemental MediaTailor Inhaltsursprung empfangen hat.

```
#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_
```

In diesem Beispiel für eine multivariante Playlist:

- Die #EXT-X-STREAM-INF Tags definieren verschiedene Wiedergabeversionen mit unterschiedlichen Auflösungen und Bitraten
- Jede Wiedergabeversion verweist auf eine Medien-Playlist (z. B.) index_1.m3u8
- Das #EXT-X-MEDIA Tag definiert einen Untertiteltrack

Das folgende Beispiel zeigt eine HLS-Medien-Playlist, die HLS von der Quelle des Inhalts AWS Elemental MediaTailor empfangen hat. In diesem Beispiel werden die Tags EXT-X-CUE-OUT und EXT-X-CUE-IN verwendet, um Ad-Avail-Gelegenheiten zu beschreiben.

```
#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
```

In diesem Beispiel für eine Medien-Playlist:

- Die #EXTINF Tags geben die Dauer jedes Segments an
- Das #EXT-X-CUE-OUT Tag markiert den Beginn einer Werbepause
- Die #EXT-X-CUE-OUT-CONT Tags geben Aufschluss über die laufende Werbepause
- Das #EXT-X-CUE-IN Tag markiert das Ende der Werbepause

Beispiele für personalisierte HLS-Manifeste

Das folgende Beispiel zeigt eine multivariante HLS-Wiedergabeliste, die personalisiert wurde. AWS Elemental MediaTailor

```
#EXTM3U
```

```
#EXT-X-VERSION:3
#EXT-X-MEDIA:LANGUAGE="eng",AUTOSELECT=YES,FORCED=NO,TYPE=SUBTITLES,URI="../../../../manifest/43f3e412052f2808dd84ea1da90e92e914edddee/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/43f3e412052f2808dd84ea1da90e92e914edddee/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/43f3e412052f2808dd84ea1da90e92e914edddee/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/43f3e412052f2808dd84ea1da90e92e914edddee/external-canary-hls/ee1696a8-4f7f-4c4c-99de-9821131847e8/2.m3u8
```

Beachten Sie, dass die Medien-Playlist so geändert MediaTailor wurde, dass sie sitzungsspezifische Informationen enthält URLs , die das Einfügen personalisierter Anzeigen ermöglichen.

Das folgende Beispiel zeigt eine Medien-Playlist, die AWS Elemental MediaTailor personalisiert wurde.

```
#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/e309ffd02ba8498d864dcaacff7a5ad9/index_1_8779957.ts?m=1566416212
#EXTINF:6.006,
https://777788889999.mediapackage.us-west-2.amazonaws.com/out/v1/e309ffd02ba8498d864dcaacff7a5ad9/index_1_8779958.ts?m=1566416212
#EXTINF:5.372,
https://777788889999.mediapackage.us-west-2.amazonaws.com/out/v1/e309ffd02ba8498d864dcaacff7a5ad9/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

```

In diesem Beispiel für eine personalisierte Medien-Playlist:

- MediaTailor hat Anzeigensegmente zwischen den Inhaltssegmenten eingefügt
- Die #EXT-X-DISCONTINUITY Tags markieren die Übergänge zwischen Inhalten und Anzeigen
- Inhaltssegmente werden vom Ursprungsserver aus bereitgestellt (MediaPackage in diesem Beispiel)
- Anzeigensegmente werden aus dem MediaTailor Anzeigensegmentspeicher bereitgestellt

Hauptunterschiede bei personalisierten Manifesten

Bei der MediaTailor Personalisierung von HLS-Manifesten werden mehrere wichtige Änderungen vorgenommen:

Änderungen an der Multivarianten-Playlist

- Varianten-Playlisten URLs werden so umgeschrieben, dass sie auf MediaTailor -managed URLs verweisen und Sitzungsinformationen enthalten
- Die Reihenfolge der Tags kann für eine optimale Wiedergabe neu organisiert werden

Änderungen der Varianten-Playlist

- Anzeigenmarkierungen (EXT-X-CUE-OUT, EXT-X-CUE-IN) werden durch tatsächliche Anzeigensegmente ersetzt
- Diskontinuitätsmarkierungen (EXT-X-DISCONTINUITY) werden an content/ad Grenzen hinzugefügt
- Inhaltssegmente URLs werden so umgeschrieben, dass sie auf den Ursprung oder das CDN verweisen
- Anzeigensegmente URLs werden zum Anzeigensegmentspeicher MediaTailor von Point To hinzugefügt

Wenn Sie sich mit diesen Änderungen vertraut machen, können Sie Probleme in Ihren MediaTailor Workflows beheben und die korrekte Konfiguration Ihres CDN und Players sicherstellen.

Verwandte Themen

Weitere Informationen zu HLS-Manifesten und MediaTailor finden Sie in den folgenden Themen:

- [Typen von HLS-Wiedergabelisten](#)— Ausführliche Erläuterung der HLS-Playlist-Typen
- [Verwendung eines CDN zur Optimierung der MediaTailor Anzeigenpersonalisierung und Inhaltsbereitstellung](#)- Informationen zur Verwendung eines CDN mit MediaTailor
- [Wie funktioniert das Einfügen von MediaTailor Anzeigen](#)- Überblick darüber, wie das Einfügen von MediaTailor Anzeigen funktioniert

Integration einer MPEG-DASH-Quelle

AWS Elemental MediaTailor unterstützt .mpd Live- und Video-on-Demand (VOD) -Manifeste, die den Richtlinien für das dynamische DASH-Profil entsprechen. MediaTailor akzeptiert DASH-

konforme Manifesteingaben für mehrere Perioden und einzelne Perioden und liefert DASH-konforme Manifestausgaben für mehrere Perioden.

Eingabemanifeste müssen Folgendes aufweisen:

- SCTE-35 Event-Streams mit Splice-Info-Einstellungen für `splice insert` oder `time signal`. Die Einstellungen können im transparenten XML- oder im base64-kodierten, binären Format angegeben werden.
- `Segment templates` mit `segment timelines`.

MediaTailor erfordert für veröffentlichte Manifeste, dass bei Aktualisierungen vom Ursprungsserver Folgendes unverändert bleibt:

- Die im Attribut `start` angegebenen Startzeiten von Zeiträumen.
- Werte von `presentationTimeOffset` in den Segmentvorlagen der Zeitraumdarstellungen.

Es hat sich bewährt, für die Werbeanzeige dieselben `AdaptationSet Representation` Einstellungen wie für die Zeiträume des Content-Streams festzulegen. AWS Elemental MediaTailor verwendet diese Einstellungen, um die Anzeigen so zu transkodieren, dass sie dem Inhaltsstream entsprechen, sodass ein reibungsloser Wechsel zwischen beiden möglich ist.

In den folgenden Abschnitten finden Sie weitere Informationen zum MediaTailor Umgang mit Anzeigen in DASH-Manifesten.

Themen

- [DASH-Werbemarkierungen](#)
- [DASH und Verfügbarkeitsdauer](#)
- [Segmentnummerierung des DASH-Manifests](#)
- [DASH MPD-Beispiele](#)
- [DASH-Ortungsfunktion](#)

DASH-Werbemarkierungen

MediaTailor

AWS Elemental MediaTailor verwendet SCTE-35-Cue-Out-Markierungen, um verfügbare Werbeanzeigen im DASH-Manifest anhand der folgenden Logik zu identifizieren:

- DASH mit mehreren Perioden: MediaTailor fügt Anzeigen für die jeweils erste Anzeige ein, die entweder oder Event Cue-Out-Markierungen Period enthält. `SpliceInsert` `TimeSignal` MediaTailor ignoriert zusätzliche Event Markierungen in der. Period
- Einzeiliges DASH: MediaTailor Fügt jeweils Anzeigen Event in die einPeriod, die entweder `SpliceInsert` oder `TimeSignal` Cue-Out-Markierungen enthalten.

AWS Elemental MediaTailor Verwaltet DASH-Manifeste standardmäßig als Manifeste mit mehreren Perioden. Sie können die Konfiguration so ändern, dass sie DASH-Manifeste mit einem einzigen Zeitraum von Ihrem Ursprungs-Server verarbeitet. Weitere Informationen finden Sie unter [the section called “Konfiguration erstellen”](#).

Die folgenden Abschnitte enthalten zusätzliche Informationen zur Handhabung von DASH-Werbemarkern und enthalten dekorierte Manifeste von Anfang an.

XML-Anforderungen für das DASH-Ursprungsmanifest

Anzeigenmarkierungen in DASH-Manifesten aus dem Ursprungsland müssen korrekt formatiert sein, MediaTailor damit Werbeunterbrechungen identifiziert werden können. In den folgenden Themen werden diese Formatierungsanforderungen in klarem XML beschrieben.

SpliceInsert in klarem XML

`SpliceInsert` Werbemarkierungen in klarem XML müssen Folgendes enthalten:

- EventStream muss das folgende Attribut haben: `schemeIdUri=urn:scte:scte35:2013:xml`
- Event muss halten `scte35:SpliceInfoSection`
- `scte35:SpliceInfoSection` muss halten `scte35:SpliceInsert`
- `scte35:SpliceInsert` muss das folgende Attribut haben: `outOfNetworkIndicator="true"`

Example **SpliceInsert** in XML

Im folgenden Beispiel sind die erforderlichen SCTE-Markierungen fett gedruckt.

```
<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>
.
.
.
</Period>

```

TimeSignal in klarem XML

TimeSignalWerbemarkierungen in klarem XML müssen Folgendes enthalten:

- EventStream muss das folgende Attribut haben: `schemeIdUri=urn:scte:scte35:2013:xml`
- Event muss halten `scte35:SpliceInfoSection`
- `scte35:SpliceInfoSection` muss halten `scte35:TimeSignal`
- `scte35:SpliceInfoSection` muss auch halten `scte35:SegmentationDescriptor`
- `scte35:SegmentationDescriptor` muss das folgende Attribut haben, wobei der Wert gültig ist [Cue-Out-Zahlen](#): `segmentationTypeId="xx"`
- `scte35:SegmentationDescriptor` muss gelten `scte35:SegmentationUpid`

Cue-Out-Zahlen

Die folgenden Cue-Out-Nummern werden für die unterstützt. `segmentationTypeId`

Segmentierungsnachricht	segmentationTypeId Wert	Hexadezimaler Wert
Ende der Werbung für den Vertriebspartner	51	0x51
Beginn der Werbung für Vertriebspartner	50	0x32

Segmentierungsnachricht	segmentationTypeld Wert	Hexadezimaler Wert
Ende der Gelegenheit zur Vermittlung eines Vertriebspartners	55	0x37
Beginn der Platzierungsmöglichkeit für Vertriebspartner	54	0x36
Pause beenden	35	0x23
Ende der Anbieter-Werbung	49	0 x 31
Beginn der Anbieterwerbung	48	0x30
Ende der Platzierungsmöglichkeit im Provider-Overlay	57	0x39
Beginn der Platzierungsmöglichkeit im Provider-Overlay	56	0x38
Ende der Gelegenheit zur Anbietervermittlung	53	0x35
Beginn der Platzierungsmöglichkeit für Anbieter	52	0x34
Pause starten	34	0 x 22

Example **TimeSignal** in XML

Im folgenden Beispiel sind die erforderlichen SCTE-Markierungen fett gedruckt.

```
<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:SpliceInfoSection>
    </Event>
  </EventStream>
</Period>
```

```

    <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>

```

Base64-kodierte Binäranforderungen für das DASH-Ursprungsmanifest

Anzeigenmarkierungen in DASH-Manifesten aus dem Ursprungsland müssen korrekt formatiert sein, MediaTailor damit Werbeunterbrechungen identifiziert werden können. In den folgenden Themen werden diese Formatierungsanforderungen in einer Base64-codierten Binärdatei beschrieben.

TimeSignalSowohl als auch SpliceInsert Werbemarkierungen in Base64-codierten Manifesten müssen Folgendes enthalten:

- EventStreammuss das folgende Attribut haben: `urn:scte:scte35:2014:xml+bin`
- Eventmuss halten `scte35:Signal`
- `scte35:Signal`muss gelten `scte35:Binary`, das eine Base64-kodierte Binärdatei enthält.

Die dekodierte Binärdatei muss dieselben Informationen enthalten wie die, die für klare XML-Werbemarkierungen erforderlich sind. `splice_info_section`

- Der Befehlstyp muss entweder `splice_insert()` sein `time_signal()`
- Die zusätzlichen Einstellungen müssen den unter [TimeSignalin klarem XML](#) und beschriebenen Einstellungen entsprechen [SpliceInsertin klarem XML](#).

Das dekodierte binäre Format muss einen `splice_info_section` mit dem gleichen Satz von Informationen bereitstellen, der im transparenten XML-Format in einem `scte35:SpliceInfoSection`-Element ausgegeben würde. Der Befehlstyp muss entweder

`splice_insert()` oder `time_signal()` sein, und die zusätzlichen Einstellungen müssen denen entsprechen, die zuvor für eine Bereitstellung im transparenten XML-Format beschrieben wurden.

Das folgende Beispiel zeigt diese Option, wobei die erforderlichen Marker fettgedruckt sind.

```
<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>QW5vdGhlciB0ZXN0IHN0cmVuZyBmb3IgaWZlZjB2RpbmcgdG8gQmFzZTY0IGVuY29kZWQgYmluYXJ5Lg=
scte35:Binary>
      </scte35:Signal>
    </Event>
    .
    .
    .
  </Period>
```

Nachstehend wird das dekodierte binäre Format für das erste, im voranstehenden Beispiel aufgeführte Ereignis angegeben. Die Einstellung für `splice_command_type` lautet 5, wodurch `splice_insert` angegeben wird.

```
{
  "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": "00:00:24:00000"
        }
    },
    "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 und Verfügbarkeitsdauer

Wenn bei der Wiedergabe AWS Elemental MediaTailor auf eine Anzeige gestoßen wird, wird die Anzeige ganz oder teilweise durch Werbung ersetzt. MediaTailor Startet das Ersetzen von Anzeigen zu Beginn der Verfügbarkeit der Anzeige und umfasst Anzeigen wie folgt:

- Wenn für die Anzeigenverfügbarkeit eine Dauer angegeben ist, werden so viele Anzeigen MediaTailor eingeschlossen, wie innerhalb der Zeitbegrenzung Platz finden, ohne dass der nachfolgende Inhalt überschrieben wird.

- Wenn keine Dauer angegeben ist, werden Anzeigen MediaTailor eingeschlossen, bis das Ende der verfügbaren Anzeige erreicht ist. Für Manifeste mit mehreren Zeiträumen ist dies das Ende des Zeitraums. Bei Manifesten mit einer Periode ist dies das Ende der Veranstaltung. MediaTailor spielt keine Werbung nach dem Ende der verfügbaren Anzeige ab und kürzt, wenn das Ende erreicht ist, die aktuelle Anzeige, anstatt den darauffolgenden Inhalt zu überschreiben.

Wie AWS Elemental MediaTailor sieht die Dauer der Anzeigenverfügbarkeit aus

AWS Elemental MediaTailor sucht in der folgenden Reihenfolge nach einer Einstellung für die Dauer:

1. Event duration
2. Für Splice-Insert-Marker unter `scte35:BreakDuration` duration
3. Für Zeitsignal-Marker unter `scte35:SegmentationDescriptor` segmentationDuration

Wenn AWS Elemental MediaTailor keine dieser Einstellungen gefunden wird, wird die Anzeigeneinbindung ohne Dauer verwaltet.

Das folgende Beispiel zeigt ein Event mit einer duration.

```
<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>
```

Das folgende Beispiel zeigt ein Ad-Avail ohne angegebene Dauer. Das Event besitzt keine duration und das Element `scte35:SpliceInsert` enthält kein untergeordnetes `scte35:BreakDuration`-Element.

```

<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>
    ...
  </EventStream>
</Period>

```

Segmentnummerierung des DASH-Manifests

MediaTailor unterstützt Mediensegmente `<SegmentTemplate>`, die mit `<SegmentTimeline>` und dem `media` Attribut definiert sind. Sie können die Liste der Mediensegmente entweder mit dem Bezeichner `$Number$` oder dem Bezeichner `$Time$` im Attribut `media` angeben.

Das folgende Beispiel zeigt eine `SegmentTemplate` mit einer `media`-Attributeinstellung, die den Bezeichner `$Number$` verwendet.

```

<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>

```

Das folgende Beispiel zeigt eine `SegmentTemplate` mit einer `media`-Attributeinstellung, die den Bezeichner `$Time$` verwendet.

```

<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"/>
  </SegmentTimeline>
</SegmentTemplate>

```

```
<S d="147000" t="540000"/>
</SegmentTimeline>
</SegmentTemplate>
```

DASH MPD-Beispiele

In den folgenden Abschnitten finden Sie Beispiele für die Herkunft MPDs und Personalisierung MPDs von DASH. Wenn Sie diese Beispiele verstehen, können Sie Ihre MediaTailor Workflows konfigurieren und Fehler beheben.

Informationen darüber, wie Abfrageparameter auf DASH-Manifeste und -Segmente angewendet werden, finden Sie unter [MediaTailor Implizite Initialisierung der DASH-Sitzung](#).

Grundlegendes zur DASH-MPD-Struktur

Dynamic Adaptive Streaming over HTTP (DASH) verwendet ein MPD-Manifest (Media Presentation Description), um Streaming-Inhalte bereitzustellen. Das MPD ist ein XML-Dokument, das die Struktur und Verfügbarkeit von Medieninhalten beschreibt.

MPD (Beschreibung der Medienpräsentation)

Die MPD ist die primäre Manifestdatei beim DASH-Streaming, die die Struktur und Verfügbarkeit von Medieninhalten beschreibt. Sie enthält Informationen über Zeiträume, Adaptionssätze, Repräsentationen und Segmente, aus denen sich die Streaming-Inhalte zusammensetzen.

Dieser Manifesttyp ist in verschiedenen Kontexten auch unter verschiedenen anderen Namen bekannt, darunter DASH-Manifest, DASH-MPD, Master-Manifest (im Vergleich zu HLS) oder Präsentationsmanifest.

In MediaTailor Workflows ist die MPD der Einstiegspunkt für Wiedergabeanfragen und der Ausgangspunkt für die Personalisierung von Anzeigen.

Zeitraum

Eine Periode ist ein zeitlicher Abschnitt einer DASH-Präsentation. Jede Periode enthält ein oder mehrere Adaptionssätze und steht für einen Zeitraum der Medienzeit. In Workflows zum Einfügen von Anzeigen werden in der Regel separate Perioden verwendet, um zwischen Inhalten und Anzeigen zu unterscheiden.

In MediaTailor Workflows werden Perioden verwendet, um den Hauptinhalt vom Anzeigeninhalt zu trennen, wobei jede Anzeige in der Regel durch einen eigenen Zeitraum dargestellt wird.

AdaptationSet

Eine AdaptationSet Gruppe von austauschbaren codierten Versionen einer oder mehrerer Medieninhaltskomponenten gruppiert. Eine AdaptationSet kann beispielsweise mehrere Videoqualitätsstufen enthalten, während eine andere mehrere Audiosprachenoptionen enthalten kann.

In MediaTailor Workflows AdaptationSets werden diese beim Einfügen von Anzeigen beibehalten, um konsistente Medientypen zwischen Inhalten und Anzeigen zu gewährleisten.

Darstellung

Eine Darstellung ist eine bestimmte kodierte Version des Medieninhalts innerhalb eines AdaptationSet. Jede Darstellung unterscheidet sich in der Regel in der Bitrate, Auflösung oder anderen Kodierungsparametern, sodass Kunden je nach Netzwerkbedingungen und Gerätefunktionen die am besten geeignete Version auswählen können.

In MediaTailor Workflows werden Darstellungen in Anzeigenperioden so genau wie möglich mit den Darstellungen in Inhaltsperioden abgeglichen, um ein reibungsloses Seherlebnis zu gewährleisten.

Ausführlichere Informationen zu DASH-Manifesttypen finden Sie unter [DASH-Manifesttypen](#).

Informationen zur Struktur und Konfiguration des DASH-Manifests finden Sie im AWS Elemental MediaPackage Abschnitt „Überblick über DASH“ im MediaPackage Benutzerhandbuch.

Live-Beispiele für DASH MPD

Dieser Abschnitt enthält Beispiele für Live-DASH MPDs. In jedem Beispiel wird eine MPD so aufgeführt, wie sie vom Ursprungsserver empfangen wurde. Danach MediaTailor wurde die MPD mit Werbung personalisiert.

Beispiel für das Einfügen eines DASH-MPD-Spleißes

Beispiel für MPD mit Ursprung in DASH für das Einfügen von Spleißverbindungen

Das folgende Beispiel aus einer MPD zeigt eine verfügbare Anzeige in einem Manifest, das DASH von der Quelle des Inhalts erhalten hat. In diesem Beispiel wird die `SpliceInsert` Markierung verwendet, um darauf hinzuweisen, dass eine Anzeige verfügbar ist.

```
<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: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>

```

In diesem Origin-MPD-Beispiel:

- Das <EventStream> Element enthält SCTE-35-Markierungen, die auf Anzeigenverfügbarkeit hinweisen
- Das <scte35:SpliceInsert> Element enthält Einzelheiten zur Verfügbarkeit der Anzeige
- Das <scte35:BreakDuration> Element gibt die Dauer der Werbeunterbrechung an
- Die <AdaptationSet> Elemente definieren die verfügbaren Video- und Audiostreams

Personalisiertes DASH-MPD-Beispiel für das Einfügen von Spleißverbindungen

AWS Elemental MediaTailor personalisiert die Anzeige mit Werbespezifikationen. Die Personalisierungen spiegeln die vom Player empfangenen Daten zum Betrachter wider sowie die Werbekampagnen, die derzeit ausgeführt werden.

Das folgende Beispiel zeigt eine Anzeige, die nach AWS Elemental MediaTailor der Personalisierung verfügbar ist.

```

<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>

```

In diesem personalisierten MPD-Beispiel:

- MediaTailor hat einen neuen Zeitraum für den Anzeigeninhalt erstellt
- Das <BaseURL> Element verweist auf den Speicherort des Anzeigeninhalts
- Die <AdaptationSet> Elemente haben eine ähnliche Struktur wie der Inhalt
- Die <Representation> Elemente bieten unterschiedliche Qualitätsstufen für den Anzeigeninhalt

Beispiel für ein DASH MPD-Zeitsignal

MPD-Beispiel für ein Zeitsignal mit DASH-Ursprung

Das folgende Beispiel aus einer MPD zeigt eine verfügbare Anzeige in einem Manifest, das DASH von der Quelle des Inhalts erhalten hat. In diesem Beispiel wird die `TimeSignal` Markierung verwendet, um darauf hinzuweisen, dass eine Anzeige verfügbar ist.

```
<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: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>
  </Period>

```

In diesem Origin-MPD-Beispiel:

- Das `<scte35:TimeSignal>` Element wird anstelle von verwendet `<scte35:SpliceInsert>`
- Das `<scte35:SegmentationDescriptor>` bietet zusätzliche Informationen zur Verfügbarkeit der Anzeige

Personalisiertes DASH-MPD-Beispiel für ein Zeitsignal

AWS Elemental MediaTailor personalisiert die Anzeige mit Werbespezifikationen. Die Personalisierungen spiegeln die vom Player empfangenen Daten zum Betrachter wider sowie die Werbekampagnen, die derzeit ausgeführt werden.

Das folgende Beispiel zeigt eine Anzeige, die nach AWS Elemental MediaTailor der Personalisierung verfügbar ist.

```

<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>
  </AdaptationSet>
</Period>

```

Die personalisierte MPD für das Zeitsignal ähnelt der für das Einfügen von Splices, wobei ein neuer Zeitraum für den Anzeigeninhalt MediaTailor erstellt wird.

VOD DASH MPD-Beispiele

Dieser Abschnitt enthält Beispiele für Video-on-Demand (VOD) DASH. MPDs In jedem Beispiel wird eine MPD so aufgeführt, wie sie vom Ursprungsserver empfangen wurde. Danach MediaTailor wurde die MPD mit Werbung personalisiert.

VOD DASH MPDs folgen derselben Struktur wie Live MPDs, haben jedoch in der Regel ein `type="static"` Attribut im MPD-Element und können mehrere Perioden für verschiedene Inhaltssegmente enthalten.

Beispiele für VOD DASH MPDs finden Sie in der MediaTailor Dokumentation unter. [DASH-Werbemarkierungen](#)

Die wichtigsten Unterschiede im Bereich Personalisierung MPDs

Bei der MediaTailor Personalisierung von DASH MPDs werden mehrere wichtige Änderungen vorgenommen:

Umgang mit Perioden

- Für Anzeigeninhalte werden neue Perioden erstellt
- Die Startzeiten der Perioden werden angepasst, um die Kontinuität des Zeitplans zu gewährleisten
- EventStream Elemente mit SCTE-35-Markern werden verarbeitet und entfernt

AdaptationSet und Umgang mit Repräsentationen

- AdaptationSets In der Anzeige werden Perioden so erstellt, dass sie dem Inhalt entsprechen AdaptationSets
- Repräsentationen werden für unterschiedliche Qualitätsstufen des Anzeigeninhalts erstellt
- SegmentTemplate Elemente werden aktualisiert, sodass sie auf den Anzeigeninhalt verweisen

Wenn Sie sich mit diesen Änderungen vertraut machen, können Sie Probleme in Ihren MediaTailor Workflows beheben und sicherstellen, dass Ihr CDN und Ihr Player ordnungsgemäß konfiguriert sind.

Verwandte Themen

Weitere Informationen zu DASH MPDs und MediaTailor finden Sie in den folgenden Themen:

- [DASH-Manifesttypen](#)- Ausführliche Erläuterung der DASH-Manifesttypen

- [Verwendung eines CDN zur Optimierung der MediaTailor Anzeigenpersonalisierung und Inhaltsbereitstellung](#)- Informationen zur Verwendung eines CDN mit MediaTailor
- [Wie funktioniert das Einfügen von MediaTailor Anzeigen](#)- Überblick darüber, wie das Einfügen von MediaTailor Anzeigen funktioniert
- Umfassende Informationen zur Struktur und MediaPackage Konfiguration des DASH-Manifests finden Sie im Abschnitt mit der DASH-Übersicht im MediaPackage Benutzerhandbuch.

DASH-Ortungsfunktion

Dieser Abschnitt enthält Informationen über die Speicherort-Funktion für DASH, die in AWS Elemental MediaTailor standardmäßig aktiviert ist. Lesen Sie diesen Abschnitt, wenn Sie CDN-Routingregeln (Content Delivery Network) für den Zugriff auf MediaTailor Manifeste erstellen. Informieren Sie sich zudem in diesem Abschnitt, wenn Sie die serverseitige Berichterstellung bei Playern verwenden, die keine Sticky-HTTP-Umleitungen unterstützen.

Was ist die Speicherort-Funktion?

Mit der Speicherort-Funktion können Player, die keine Sticky-HTTP-Umleitungen unterstützen, in ihren Manifest-Aktualisierungsanforderungen Sticky-Verhalten ermöglichen.

AWS Elemental MediaTailor verwendet eine Sitzungslose Initialisierung und verlangt von seinen Playern ein Sticky-HTTP-Umleitungsverhalten. Bei serverseitiger Berichterstattung gibt der Dienst, wenn der Spieler eine Anfrage für eine Aktualisierung des Manifests stellt MediaTailor, eine temporäre 302-Weiterleitung aus, um den Spieler an einen Endpunkt für das personalisierte Manifest weiterzuleiten. MediaTailor schließt eine Sitzungs-ID als Abfrageparameter in die Antwort ein. Beabsichtigt ist, dass der Player die URL über die gesamte Sitzung hinweg nachverfolgt, aber Player, die keine Sticky-HTTP-Umleitungen unterstützen, geben die Umleitung auf und kehren zur ursprünglichen URL zurück. Wenn ein Spieler zur ursprünglichen URL zurückkehrt, wird für jede neue Anfrage eine neue Sitzung MediaTailor erstellt, anstatt bei der ursprünglichen Sitzung zu bleiben. Dies kann zu einer Beschädigung des Manifests führen.

Die DASH-Spezifikation bietet eine Lösung für dieses Problem in der Ortungsfunktion, die in AWS Elemental MediaTailor Konfigurationen standardmäßig aktiviert ist. Wenn diese Funktion aktiviert ist, wird MediaTailor die absolute URL in das <Location> Manifest-Tag eingefügt. Player, die keine Sticky-HTTP-Umleitungen unterstützen, können mit der in <Location> bereitgestellten URL Aktualisierungen am Manifest anfordern.

Muss ich die Speicherort-Funktion in meiner Konfiguration deaktivieren?

Die Speicherort-Funktion überschreibt alle CDN Routing-Regeln, die Sie für den Zugriff auf AWS Elemental MediaTailor -Manifeste einrichten. Daher müssen Sie sie möglicherweise deaktivieren. Der Speicherort-Funktion wirkt sich nicht auf das CDN-Caching von Inhalten oder Werbe-Segmenten aus.

Bestimmen Sie anhand Ihrer Situation in der folgenden Liste, ob Sie die Speicherort-Funktion für Ihre Konfiguration deaktivieren müssen und wie dabei vorzugehen ist:

- Wenn keine CDN-Routing-Regeln für den Zugriff auf AWS Elemental MediaTailor -Manifeste eingerichtet sind, lassen Sie die Speicherort-Einstellung aktiviert.
- Verwenden Sie andernfalls die folgende Regeln:
 - Wenn Sie keine serverseitige Berichterstellung verwenden oder Ihre Player alle Sticky-HTTP-Umleitungen unterstützen, deaktivieren Sie die Speicherort-Funktion. Weitere Informationen, wie dies über die Konsole möglich ist, finden Sie unter [the section called “Konfiguration erstellen”](#).
 - Andernfalls wenden Sie sich an den [AWS-Support](#).

Muss die Speicherortfunktion verwendet werden?

Sie müssen die Standortfunktion für Spieler verwenden, die keine Sticky-HTTP-Weiterleitungen unterstützen. Verwenden Sie die angegebene URL in dem <Location>-Tag für alle Ihre Manifest-Aktualisierungsanfragen.

Beispiel

Beispiel URLs und <Location> Beispiel-Tag.

- Example Beispiel: Anfängliche Anforderungs-URL

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

- Example Beispiel: Umgeleiteten 302-Antwort

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

- Example Beispiel: Speicherort-Tag in einem Manifest

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

Sicherung der AWS Elemental MediaTailor ursprünglichen Interaktionen mit SigV4

Signature Version 4 (Sigv4) ist ein Signaturprotokoll, das zur Authentifizierung von MediaTailor Anfragen an unterstützte Quellen über HTTPS verwendet wird. MediaTailor unterstützt nur HTTPS-Kommunikation und erlaubt keine HTTP-Verbindungen. Schließt bei der Sigv4-Signatur einen signierten Autorisierungsheader in die HTTPS-Ursprungsanfrage an MediaTailor Channel Assembly, Amazon S3 und AWS Elemental MediaPackage Version 2 ein. MediaTailor

Sie können SigV4 an Ihrem Ursprung verwenden, um sicherzustellen, dass Manifestanfragen nur erfüllt werden, wenn sie von einem signierten Autorisierungsheader stammen MediaTailor und einen signierten Autorisierungsheader enthalten. Auf diese Weise wird verhindert, dass nicht autorisierte MediaTailor Wiedergabekonfigurationen auf Ihre ursprünglichen Inhalte zugreifen können. Wenn der signierte Autorisierungsheader gültig ist, erfüllt Ihr Absender die Anfrage. Wenn er nicht gültig ist, schlägt die Anfrage fehl.

In den folgenden Abschnitten werden die Anforderungen für die Verwendung der MediaTailor SigV4-Signatur für unterstützte Ursprünge beschrieben.

MediaTailor Anforderungen an die Kanalmontage

Wenn Sie SigV4 verwenden, um Ihren MediaTailor Channel Assembly-Ursprung zu schützen, müssen die folgenden Anforderungen erfüllt sein, MediaTailor um auf das Manifest zugreifen zu können:

- Die ursprüngliche Basis-URL in Ihrer MediaTailor Konfiguration muss ein Channel Assembly-Channel im folgenden Format sein: `channel-assembly.mediatailor.region.amazonaws.com`
- Ihr Ursprung muss für die Verwendung von HTTPS konfiguriert sein. MediaTailor unterstützt nur HTTPS-Kommunikation und erlaubt keine HTTP-Verbindungen. Wenn HTTPS am Ursprung nicht aktiviert ist, MediaTailor wird die Anfrage nicht signiert.

- Für deinen Kanal muss eine Ursprungszugriffsrichtlinie gelten, die Folgendes beinhaltet:
 - Hauptzugriff für MediaTailor den Zugriff auf deinen Kanal. Gewähren Sie Zugriff auf `mediatailor.amazonaws.com`.
 - IAM berechtigt `mediatailor`: zum Lesen aller multivarianten Playlisten, auf die in der Konfiguration verwiesen wird. `GetManifest` MediaTailor

Informationen zum Einrichten einer Richtlinie für den Kanal finden Sie unter. [Erstellen Sie mit der MediaTailor Konsole einen Kanal](#)

Example Ursprungszugriffsrichtlinie für Channel Assembly, bezogen auf das MediaTailor Konfigurationskonto

```
{
  "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 Ursprungszugriffsrichtlinie für Channel Assembly, bezogen auf die Wiedergabekonfiguration MediaTailor

```
{
  "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 S3-Anforderungen

Wenn Sie SigV4 verwenden, um Ihren Amazon S3 S3-Ursprung zu schützen, müssen die folgenden Voraussetzungen erfüllt sein, MediaTailor um auf das Manifest zugreifen zu können:

- Die ursprüngliche Basis-URL in Ihrer MediaTailor Konfiguration muss ein S3-Bucket im folgenden Format sein: `s3.region.amazonaws.com`
- Ihr Ursprung muss für die Verwendung von HTTPS konfiguriert sein. MediaTailor unterstützt nur HTTPS-Kommunikation und erlaubt keine HTTP-Verbindungen. Wenn HTTPS am Ursprung nicht aktiviert ist, MediaTailor wird die Anfrage nicht signiert.
- Für deinen Kanal muss eine Ursprungszugriffsrichtlinie gelten, die Folgendes beinhaltet:
 - Hauptzugriff für den MediaTailor Zugriff auf deinen Bucket. Gewähren Sie Zugriff auf `mediatailor.amazonaws.com`.

Informationen zur Konfiguration des Zugriffs in IAM finden Sie unter [Zugriffsverwaltung](#) im AWS Identity and Access Management-Benutzerhandbuch.

- IAM-Berechtigungen `s3:GetObject` zum Lesen aller Manifeste der obersten Ebene, auf die in der Konfiguration verwiesen wird. MediaTailor

Allgemeine Informationen zu SigV4 für Amazon S3 finden Sie im Thema [Authenticating Requests \(AWS Signature Version 4\)](#) in der Amazon S3 S3-API-Referenz.

Example Ursprungszugriffsrichtlinie für Amazon S3, bezogen auf das Konto MediaTailor

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

Example Origin-Zugriffsrichtlinie für Amazon S3, abhängig von der Wiedergabekonfiguration MediaTailor

```
{
  "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 Anforderungen

Wenn Sie SigV4 verwenden, um Ihren MediaPackage v2-Ursprung zu schützen, müssen die folgenden Voraussetzungen erfüllt sein, MediaTailor um auf das Manifest zugreifen zu können:

- Die ursprüngliche Basis-URL in Ihrer MediaTailor Konfiguration muss ein MediaPackage v2-Endpunkt im folgenden Format sein: `mediapackagev2.region.amazonaws.com`
- Ihr Ursprung muss für die Verwendung von HTTPS konfiguriert sein. MediaTailor unterstützt nur HTTPS-Kommunikation und erlaubt keine HTTP-Verbindungen. Wenn HTTPS am Ursprung nicht aktiviert ist, MediaTailor wird die Anfrage nicht signiert.
- Für deinen Kanal muss eine Ursprungszugriffsrichtlinie gelten, die Folgendes beinhaltet:
 - Hauptzugriff für den MediaTailor Zugriff auf deinen Endpunkt. Gewähren Sie Zugriff auf `mediatailor.amazonaws.com`.
 - IAM-Berechtigungen `mediapackagev2`: zum Lesen aller multivarianten Playlisten, auf die in der Konfiguration verwiesen wird. `GetObject` MediaTailor

Allgemeine Informationen zu SigV4 für MediaPackage v2 finden Sie im Thema [Authenticating Requests \(AWS Signature Version 4\)](#) in der MediaPackage v2-API-Referenz.

Example Origin-Zugriffsrichtlinie für MediaPackage Version 2, bezogen auf das Konto 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:SourceAccount": "444455556666"}
  }
}

```

Example Origin-Zugriffsrichtlinie für MediaPackage Version 2, bezogen auf die Wiedergabekonfiguration 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"}
  }
}
```

Integration AWS Elemental MediaTailor mit Google Ad Manager

Durch die Integration MediaTailor mit [Google Ad Manager](#) (Ad Manager) erhalten Sie programmatischen Zugriff auf einen auktionsgesteuerten Online-Marktplatz, auf dem Anzeigenimpressionen in Echtzeit gekauft und verkauft werden können. Sie müssen ein Konto bei Ad Manager eingerichtet haben. Anschließend können Sie die Integration mit Ad Manager auf folgende Weise durchführen:

- Eine serverseitige Integration mit einem SSL-Zertifikat.
- Eine clientseitige Player-Integration unter Verwendung des Programmatic Access Libraries (PAL) - SDK. Diese Integration ist erforderlich, wenn Sie den Transaktionstyp „Offene Auktion“ verwenden möchten.

Die Ad Manager-Unterstützung für programmatische Transaktionstypen hängt von der Art der Integration ab, die Sie verwenden. Eine Liste der verfügbaren Optionen finden Sie unter [Transaktionstypen](#) oder wenden Sie sich an Ihr Google-Kontoteam.

In den folgenden Abschnitten werden diese Integrationen detailliert beschrieben.

Themen

- [Serverseitige AWS Elemental MediaTailor Integration mit Google Ad Manager](#)
- [Kundenseitige AWS Elemental MediaTailor Integration mit Google Ad Manager](#)

Serverseitige AWS Elemental MediaTailor Integration mit Google Ad Manager

Serverseitige Anzeigenanfragen an Google Ad Manager (Ad Manager) müssen das SSL-Zertifikat enthalten, für das Ad Manager ausgestellt hat, um programmatische Transaktionen MediaTailor zu autorisieren.

Um serverseitige Anzeigenanfragen an Ad Manager zu stellen

1. [Reichen Sie ein AWS Support-Ticket](#) ein, um die Aktivierung von SSL-Zertifikaten anzufordern. Nehmen Sie die folgenden Informationen in das Support-Ticket auf:
 - AWS Region
 - AWS Konto-ID
 - MediaTailor Name der Wiedergabekonfiguration

Wenn Sie keine SSL-Zertifikate aktivieren, reagiert Ad Manager auf MediaTailor Anzeigenanfragen mit HTTP 401-Fehlercodes im Ereignistyp `ERROR_ADS_INVALID_RESPONSE` ADS-Interaktionsprotokoll.

2. Nachdem die SSL-Zertifikate aktiviert wurden, aktualisieren Sie die URL und die Parameter für Ihre ADS- und Preroll-ADS in der Wiedergabekonfiguration. Informationen zum Aktualisieren oder Erstellen einer Wiedergabekonfiguration finden Sie unter [MediaTailor Verwaltung der Wiedergabekonfiguration](#).

Offizielle Hinweise zu URL-Parametern für VAST-Anzeigenanfragen für Ad Manager finden Sie im [Leitfaden zur serverseitigen Implementierung](#) von Ad Manager. Die Aktualisierung beinhaltet die folgenden Änderungen:

- Ändern Sie die Basis-URL von `pubads.g.doubleclick.net` zu `zuserside.doubleclick.net`.
- Fügen Sie den `ssss=mediatailor`-Parameter hinzu. Dies weist darauf hin, dass MediaTailor es sich um die serverseitige Stitching-Quelle handelt.
- Entfernen Sie den Parameter `IP`. MediaTailor übergibt automatisch die IP-Adresse des Endbenutzers mithilfe des `X-Forwarded-For` Headers.
- Entfernen Sie den `ss_req=1` Parameter.

Aktualisierte und vollständige VAST-URL-Anleitungen finden Sie im [Leitfaden zur serverseitigen Implementierung](#) oder wenden Sie sich an Ihr Google-Kontoteam.

Kundenseitige AWS Elemental MediaTailor Integration mit Google Ad Manager

Für die Verwendung der Google Ad Manager Programmatic Access Libraries (PAL) ist eine MediaTailor clientseitige Integration erforderlich. SDKs Diese Integration ist erforderlich, wenn Sie den Transaktionstyp „Offene Auktion“ von Ad Manager verwenden möchten.

Die PAL SDKs enthalten Informationen über den Inhalt, das Gerät und die Benutzerdaten für eine Wiedergabesitzung. Über das PAL-SDK können Sie diese Informationen an Google Ad Manager weitergeben, der dann besser bestimmen kann, welche zielgerichteten Anzeigen geschaltet werden sollen. SDKs sind für Android HTML5, iOS und Cast verfügbar. Informationen zur Verwendung von PAL SDKs finden Sie unter [Google Ad Manager PAL SDK](#).

So erstellen Sie eine clientseitige Integration mit Ad Manager

1. Verwenden Sie das PAL-SDK, um eine Nonce zu generieren.

Die Nonce ist eine verschlüsselte Zeichenfolge, die PAL für Stream-Anfragen generiert. Jede Anfrage muss eine eindeutige Nonce haben. Informationen zum Einrichten einer Nonce erhalten Sie, wenn Sie Ihr SDK aus dem [Google Ad Manager PAL](#) SDK auswählen.

2. Verwenden Sie den `givn` Parameter in Ihrer ADS-Anfrage, um den Nonce-Wert zu übergeben. Aktualisieren Sie dazu Ihre ADS-URL so, dass sie Folgendes enthält:
&givn=[*player_params.givn*] Detaillierte Anweisungen finden Sie unter [Aktivierung von clientseitigem Tracking](#).

Datazoom-Player SDKs

MediaTailor hat sich mit Datazoom zusammengetan, um einen kostenlosen Player zur Verfügung zu stellen, mit SDKs der Integrationen, SDKs wie sie beispielsweise im Ad Manager PAL angeboten werden, vereinfacht werden können. Informationen zum Datazoom und zur Partnerschaft finden Sie unter MediaTailor [Kostenloser Datazoom-Player SDKs](#)

[Um auf den Datazoom Player zuzugreifen SDKs, verwenden Sie die Kontaktinformationen auf der Datazoom with -Website. AWS](#)

Anpassen des Verhaltens von Werbeunterbrechungen mit der Unterdrückung von Werbeunterbrechungen

Wenn Sie eine Konfiguration in erstellen AWS Elemental MediaTailor, können Sie optionale Konfigurationseinstellungen für Werbeunterbrechungen angeben, die das Verhalten von Werbeunterbrechungen steuern, einschließlich der Möglichkeit, die Unterdrückung von Werbeunterbrechungen zu konfigurieren. Auf diese Weise können Sie die Werbeblock-Erlebnisse für Ihre Videoinhalte an Ihre spezifischen Anforderungen anpassen.

Einschränkungen bei der Kompatibilität

In folgenden Fällen können Sie die Unterdrückung von Werbeunterbrechungen nicht verwenden:

- VOD und live-to-VOD Workflows. Es werden nur Live-Workflows unterstützt.
- Methoden zur servergesteuerten Anzeigeneinfügung (SGAI). Servergestützte Methoden behandeln die Entscheidungsfindung bei Werbeanzeigen unterschiedlich und erfordern keine Konfiguration zur Unterdrückung.

Themen

- [Konfiguration der Unterdrückung von Werbeunterbrechungen](#)

Konfiguration der Unterdrückung von Werbeunterbrechungen

Sie können die Personalisierung von MediaTailor Live-Inhalten so konfigurieren, dass die Personalisierung von Werbeunterbrechungen übersprungen wird. Dies wird als Unterdrückung von Werbeunterbrechungen oder Avail-Unterdrückung bezeichnet. In diesem Thema erfahren Sie, wie das geht, und es wird auch erklärt, wie die Konfiguration der Unterdrückung von Werbeunterbrechungen funktioniert.

Die Unterdrückung von Werbeunterbrechungen kann für die folgenden Anwendungsfälle verwendet werden:

- Betrachtungszeitraum für große Manifeste – Wenn ein Betrachter die Wiedergabe an der Live-Grenze eines Manifests startet, der Betrachtungszeitraum aber groß ist, empfiehlt es sich,

Werbung erst ab dem Zeitpunkt einzufügen, an dem der Betrachter mit der Wiedergabe begonnen hat. Oder fügen Sie Werbeanzeigen für einen Teil des gesamten Betrachtungszeitraum im Manifest ein. Sie können die Anzeigenunterdrückung so konfigurieren, dass Werbeunterbrechungen innerhalb oder innerhalb eines bestimmten Zeitraums hinter dem Live-Edge MediaTailor personalisiert werden.

- Teilnahme mitten in der Pause – Wenn der Betrachter mit der Ansicht eines Live-Videostream mitten in einer Werbepause beginnt, ist die Wahrscheinlichkeit groß, dass dieser Benutzer den Kanal wechselt und sich die Werbung nicht ansieht. Mit der Anzeigenunterdrückungsfunktion können Sie die Personalisierung von Werbeunterbrechungen überspringen, wenn die Werbeunterbrechung begann, bevor der Zuschauer dem Stream beigetreten ist.

Anzeigenunterdrückung konfigurieren

Um die Anzeigenunterdrückungsfunktion zu verwenden, konfigurieren Sie auf folgende Weise einen Verfügbarkeitsunterdrückungsmodus, einen Verfügbarkeitsunterdrückungswert und eine Richtlinie zum Ausfüllen der Unterdrückung:

- In der Konsole MediaTailor
- Mit dem AWS Command Line Interface (AWS CLI)
- Verwenden Sie die MediaTailor API oder als Parameter in der Anfrage für die Wiedergabe-Sitzung Ihres Clients

Hinweise zur Konfiguration mit Parametern finden Sie unter [Konfigurieren von Werbeunterdrückungsparametern – Wiedergabe-Sitzungsanforderung](#).

Konfigurationsparameter für die Anzeigenunterdrückung

Sie können wählen, ob Sie die Werbeunterdrückung ein- oder ausschalten möchten. Wenn Sie die Werbeunterdrückung aktivieren, geben Sie an, ob diese Unterdrückung nach dem Rand der Live-Wiedergabe oder vor dem Rand der Live-Wiedergabe eines Live-Streams erfolgt. In beiden Fällen geben Sie auch eine Zeit an, die sich auf den Live-Rand bezieht und in der Werbung MediaTailor nicht personalisiert wird. Wenn Sie die Unterdrückung von Inhalten aktivieren, können Sie eine Richtlinie zur Unterdrückung von Inhalten angeben, die dafür MediaTailor verwendet wird, dass Werbeunterbrechungen nur teilweise aufgefüllt werden, wenn eine Sitzung mitten in der Pause beginnt.

Im Folgenden sind die Konfigurationsparameter für die Werbeunterdrückung aufgeführt:

- Avail suppression mode (Werbeplatz-Unterdrückungsmodus) – Legt den Werbeunterdrückungsmodus fest. Standardmäßig ist die Werbeunterdrückung deaktiviert. Zulässige Werte: OFF,, oder. BEHIND_LIVE_EDGE AFTER_LIVE_EDGE
- OFF: Es gibt keine Werbeunterbrechung und alle Werbeunterbrechungen werden MediaTailor personalisiert.
- BEHIND_LIVE_EDGE: personalisiert MediaTailor keine Werbeunterbrechungen, die vor dem Live-Edge beginnen, abzüglich des Werts für die Avail-Unterdrückung. Dies wirkt sich auf die gesamte Werbeunterbrechung aus, nicht nur auf einzelne Werbeanzeigen.
- AFTER_LIVE_EDGE: Personalisiert MediaTailor keine Werbeunterbrechungen, die innerhalb des Live-Randes liegen, plus den Avail-Unterdrückungswert. Dieser Wert kann so konfiguriert werden, dass entweder ganze Werbeunterbrechungen oder nur ein teilweises Ausfüllen von Werbeunterbrechungen möglich ist.
- Avail suppression value (Werbeplatz-Unterdrückungswert) – Ein Zeitpunkt relativ zur Live-Grenze in einem Livestream. Akzeptierter Wert: Ein Zeitwert in HH:MM:SS.
- Richtlinie zur Unterdrückung von Verfügbarkeiten — Definiert die Richtlinie, die für den Unterdrückungsmodus Avail MediaTailor gilt. Akzeptierte Werte: PARTIAL_AVAIL, FULL_AVAIL_ONLY.
 - BEHIND_LIVE_EDGE Der Modus verwendet immer die FULL_AVAIL_ONLY Unterdrückungsrichtlinie.
 - AFTER_LIVE_EDGE Der Modus kann verwendet werden, um das Auffüllen von PARTIAL_AVAIL Werbeunterbrechungen aufzurufen, wenn eine Sitzung mitten in der Pause beginnt.

Beispiele für Einstellungen zur Anzeigenunterdrückung

Durch die Art und Weise, wie die [Konfigurationsparameter für die Anzeigenunterdrückung](#) miteinander interagieren, können Sie verschiedene Methoden für die Anzeigenunterdrückung angeben und das Ausfüllen vor, am oder nach dem Live-Rand des Livestreams nutzen. In diesem Abschnitt finden Sie Beispiele, die Ihnen einige dieser Interaktionen zeigen. Verwenden Sie diese Beispiele, um Ihnen bei der Einrichtung der Konfigurationsparameter für Ihre spezielle Situation zu helfen.

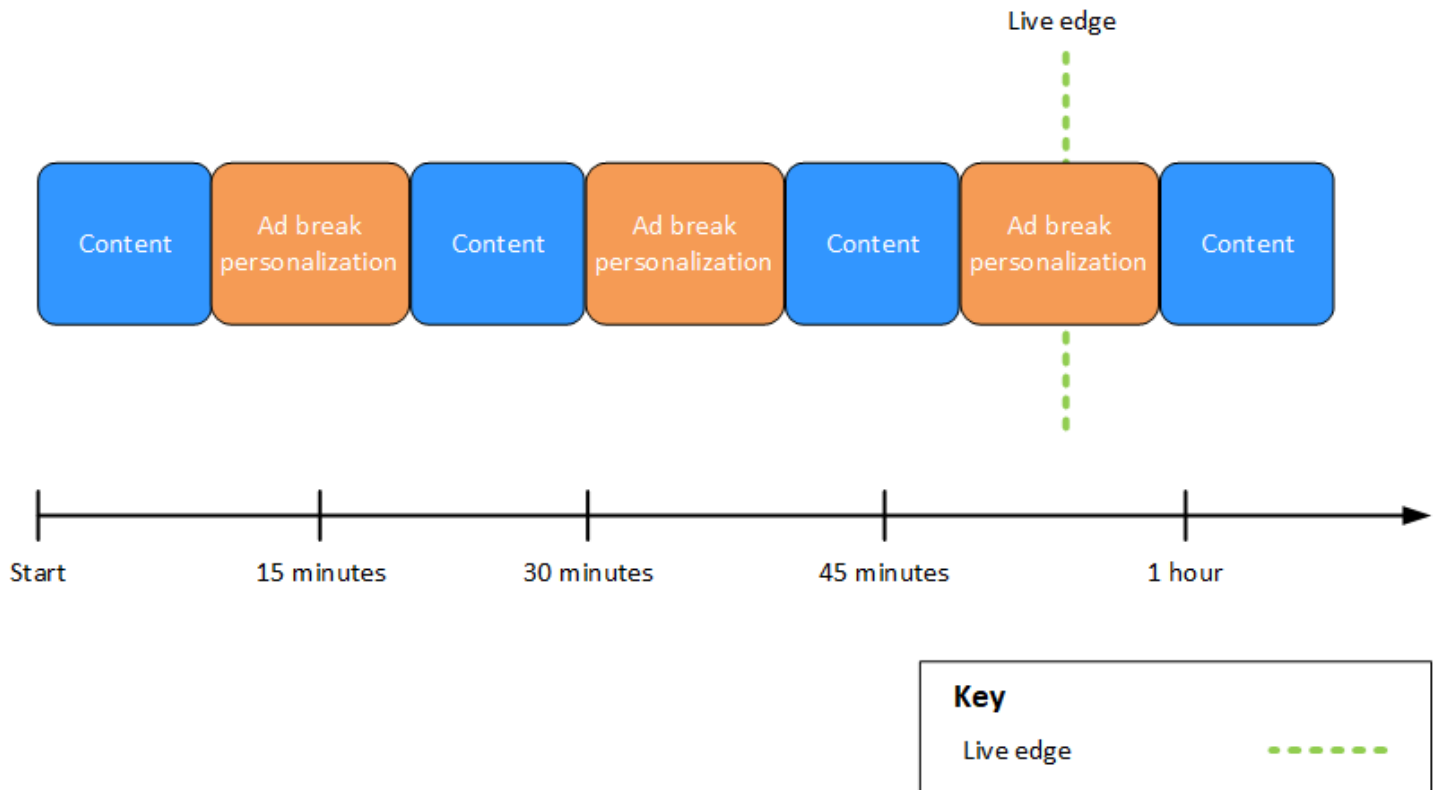
Im Folgenden finden Sie Beispiele für Einstellungen zur Anzeigenunterdrückung:

Example 1: Keine Werbeunterdrückung

Wenn der Avail-Unterdrückungsmodus aktiviert ist OFF, gibt es keine Werbeunterbrechung und alle Werbeunterbrechungen werden MediaTailor personalisiert.

In der folgenden Abbildung sind verschiedene Blöcke horizontal entlang einer Zeitachse angeordnet, die von links nach rechts verläuft. Jeder Block steht für einen Zeitraum, in dem der Inhalt des Livestreams oder einer personalisierten Werbeunterbrechung wiedergegeben wird. Eine gepunktete Linie steht für den aktuellen Live-Rand des Live-Streams. Vor dem Live-Edge finden zwei Werbeunterbrechungen statt, und am Live-Edge wird gerade eine weitere Werbeunterbrechung ausgeführt. Wie in der Abbildung dargestellt, werden bei aktiviertem Verfügbarkeitsunterdrückungsmodus alle Werbeunterbrechungen OFF, die vor dem Live-Edge stattfinden, auf der Timeline MediaTailor personalisiert. MediaTailor personalisiert auch die Werbeunterbrechung, die gerade am Live-Edge läuft.

Avail suppression mode (default): OFF



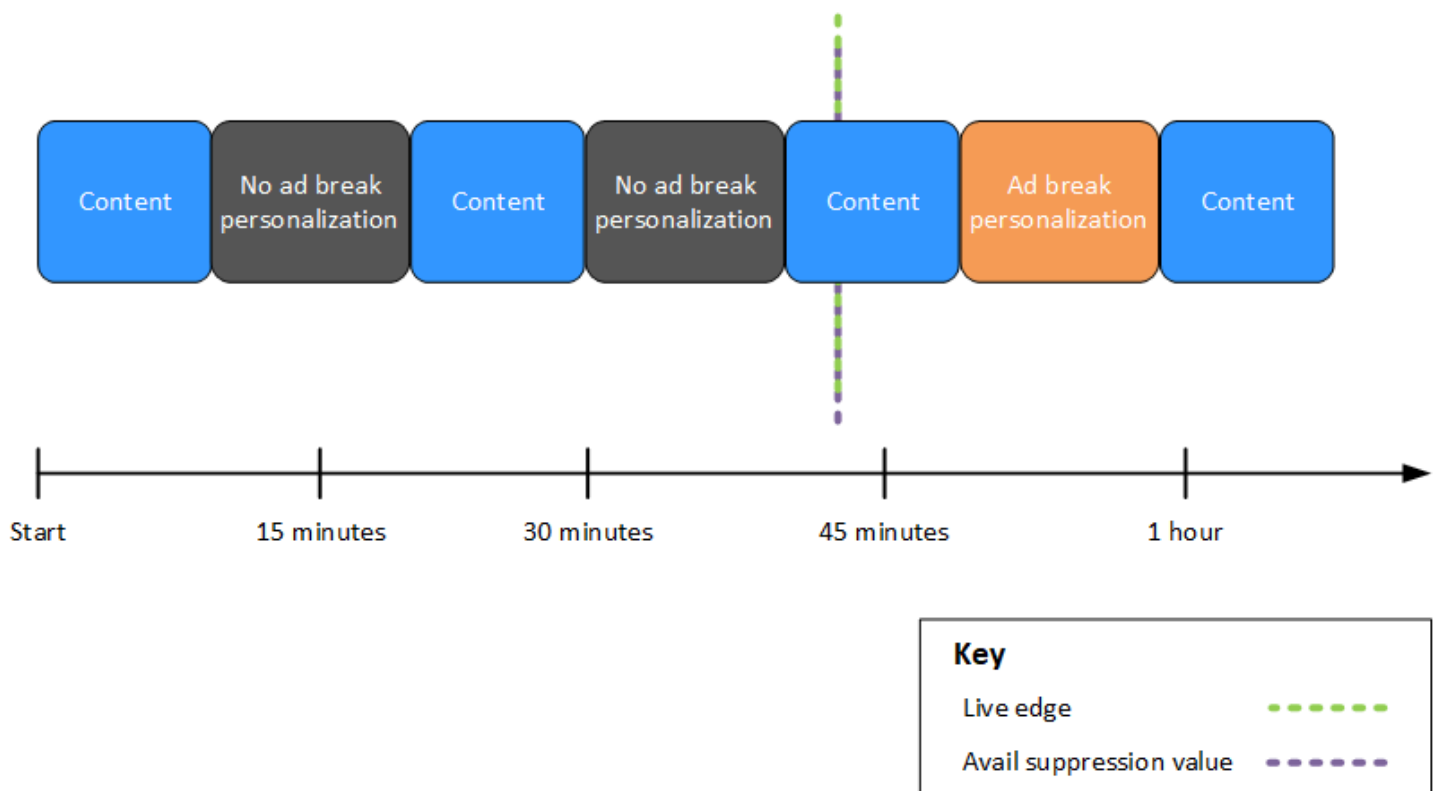
Example 2: **BEHIND_LIVE_EDGE** Werbeunterdrückung, deren Wert mit Live Edge synchron ist

Wenn der Verfügbarkeitsunterdrückungsmodus auf **BEHIND_LIVE_EDGE** und der Wert für die Verfügbarkeitsunterdrückung auf eingestellt ist `00:00:00`, ist der Verfügbarkeitsunterdrückungswert mit dem Live-Edge synchronisiert. MediaTailor personalisiert keine Werbeunterbrechungen, die am oder vor dem Live-Edge beginnen.

In der folgenden Abbildung sind verschiedene Blöcke horizontal entlang einer Zeitleiste angeordnet, die von links nach rechts verläuft. Jeder Block steht für einen Zeitraum, in dem der Inhalt eines Live-Streams, einer personalisierten Werbepause oder einer nicht personalisierten Werbepause abgespielt wird. Eine gepunktete Linie steht für den aktuellen Live-Rand des Live-Streams. Eine weitere gepunktete Linie, die den auf eingestellten Wert für die Funktionsunterdrückung darstellt 00:00:00, überlappt die gepunktete Linie für den Live-Rand. Zwei Werbeunterbrechungen treten vor der Live-Kante auf, und eine weitere Werbeunterbrechung erfolgt nach der Live-Kante. Wie in der Abbildung dargestellt, werden Werbeunterbrechungen, die vor dem Live-Edge auftreten **BEHIND_LIVE_EDGE**, auf der Timeline MediaTailor nicht personalisiert, wenn der Avail-Unterdrückungsmodus auf eingestellt ist und der Avail-Unterdrückungswert 00:00:00 so eingestellt ist, dass er mit dem Live-Edge synchronisiert ist. MediaTailor personalisiert die Werbeunterbrechung, die nach dem Live-Edge erfolgt.

Avail suppression mode: **BEHIND_LIVE_EDGE**

Avail suppression value: 00:00:00



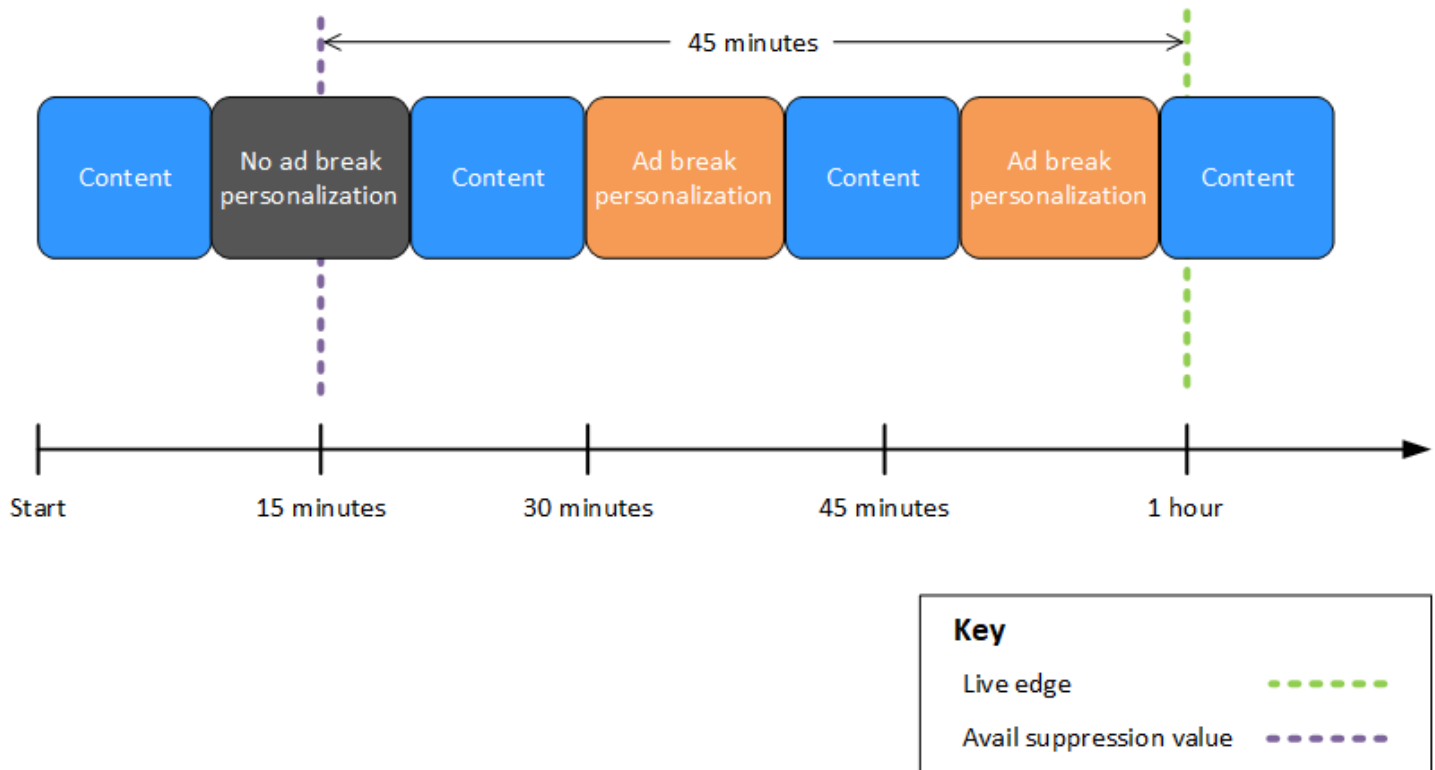
Example 3: **BEHIND_LIVE_EDGE** Werbeunterdrückung mit einem Wert hinter dem Live-Edge

Wenn der Verfügbarkeitsunterdrückungsmodus aktiviert ist **BEHIND_LIVE_EDGE**, werden MediaTailor keine Werbeunterbrechungen an oder vor diesem Zeitpunkt personalisiert. MediaTailor personalisiert in diesem Beispiel Werbeunterbrechungen, die bis zu 45 Minuten hinter dem Live-Edge beginnen. MediaTailor personalisiert keine Werbeunterbrechungen, die mehr als 45 Minuten hinter dem Live-Edge beginnen.

In der folgenden Abbildung sind verschiedene Blöcke horizontal entlang einer Zeitachse angeordnet, die von links nach rechts verläuft. Jeder Block steht für einen Zeitraum, in dem der Inhalt eines Live-Streams, einer personalisierten Werbepause oder einer nicht personalisierten Werbepause abgespielt wird. Eine gepunktete Linie steht für den aktuellen Live-Rand des Live-Streams. Eine weitere gepunktete Linie, die den auf eingestellten Wert für die Funktionsunterdrückung darstellt `00:45:00`, befindet sich 45 Minuten früher in der Timeline als die gepunktete Linie für den Live-Rand. Der Zeitraum von 45 Minuten zwischen den gepunkteten Linien steht für den Zeitraum der Verfügbarkeitsunterbindung. Zu Beginn des Zeitraums zur Verfügbarkeitsunterbrechung ist eine Werbeunterbrechung im Gange. Zwei weitere Werbeunterbrechungen treten während des Zeitraums der Verfügbarkeitsunterbrechung auf. Wie in der Abbildung dargestellt, werden alle Werbeunterbrechungen, die innerhalb des Verfügbarkeitsunterdrückungszeitraums auftreten **BEHIND_LIVE_EDGE**, MediaTailor personalisiert, wenn der Verfügbarkeitsunterdrückungsmodus auf und der Wert für die Verfügbarkeitsunterbindung auf `00:45:00` hinter dem Live-Edge eingestellt ist. MediaTailor personalisiert nicht die Werbeunterbrechung, die zu Beginn des Zeitraums zur Verfügbarkeitsunterbindung gerade läuft.

Avail suppression mode: **BEHIND_LIVE_EDGE**

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



Example 4: **AFTER_LIVE_EDGE** Werbeunterbrechung ohne Werbeunterbrechungen während der Dauer der Verfügbarkeitsunterbrechung

Wenn der Verfügbarkeitsunterdrückungsmodus auf eingestellt ist **AFTER_LIVE_EDGE** und der Wert für die Verfügbarkeitsunterbindung größer als Null ist, werden Werbeunterbrechungen erst personalisiert, wenn die verstrichene Zeit der Sitzung diesen Wert erreicht MediaTailor hat.

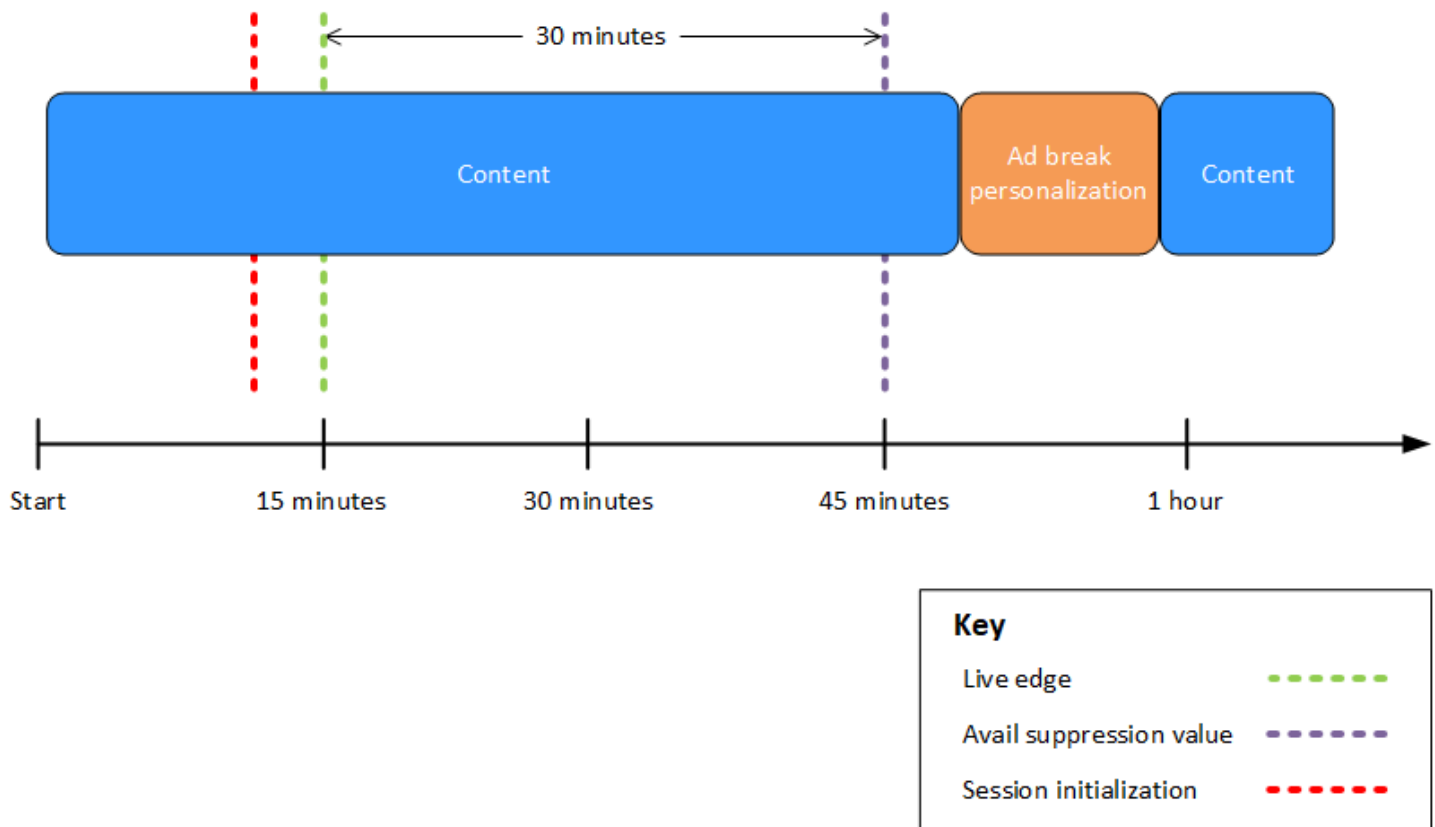
In der folgenden Abbildung sind verschiedene Blöcke horizontal entlang einer Zeitleiste angeordnet, die von links nach rechts verläuft. Jeder Block steht für einen Zeitraum, in dem der Inhalt des Livestreams oder einer personalisierten Werbepause wiedergegeben wird. Eine gepunktete Linie steht für den aktuellen Live-Rand des Live-Streams. Eine weitere gepunktete Linie, die den eingestellten Wert für die Verfügbarkeitsunterdrückung darstellt **00:30:00**, erscheint 30 Minuten später in der Timeline in Bezug auf die gepunktete Linie für den Live-Rand. Eine dritte gepunktete Linie, die für die Sitzungsinitialisierung steht, befindet sich in der Timeline an einem anderen Punkt als die gepunktete Linie für die Live-Kante. Der Zeitraum von 30 Minuten zwischen der Live-Edge-Zeit und der Uhrzeit stellt den Zeitraum für die Unterdrückung der avail-suppression-value

Nutzung dar. Nach Ablauf der Verfügbarkeitsunterbrechung erfolgt eine Werbeunterbrechung.

Wie in der Abbildung dargestellt, wird, wenn der Avail-Unterdrückungsmodus auf eingestellt ist `AFTER_LIVE_EDGE`, der Avail-Unterdrückungswert auf „00:30:00 nach dem Live Edge“ gesetzt, und die Sitzungsinitialisierung erfolgt vor dem Live-Edge, wodurch alle Werbeunterbrechungen MediaTailor personalisiert werden, die nach dem Verfügbarkeitsunterdrückungszeitraum auftreten.

Avail suppression mode: `AFTER_LIVE_EDGE`

Avail suppression value: 00:30:00



Example 5: `AFTER_LIVE_EDGE` Werbeunterbrechung mit der Option „`PARTIAL_AVAIL`Ausfüllen“ und einer aktiven Werbeunterbrechung am Ende des Zeitraums

Wenn der Modus zur Verfügungsunterbindung auf eingestellt ist `AFTER_LIVE_EDGE` und der Wert für die Verfügbarkeitsunterbindung größer als Null ist, werden Werbeunterbrechungen erst personalisiert, wenn die verstrichene Zeit der Sitzung diesen Wert erreicht MediaTailor hat.

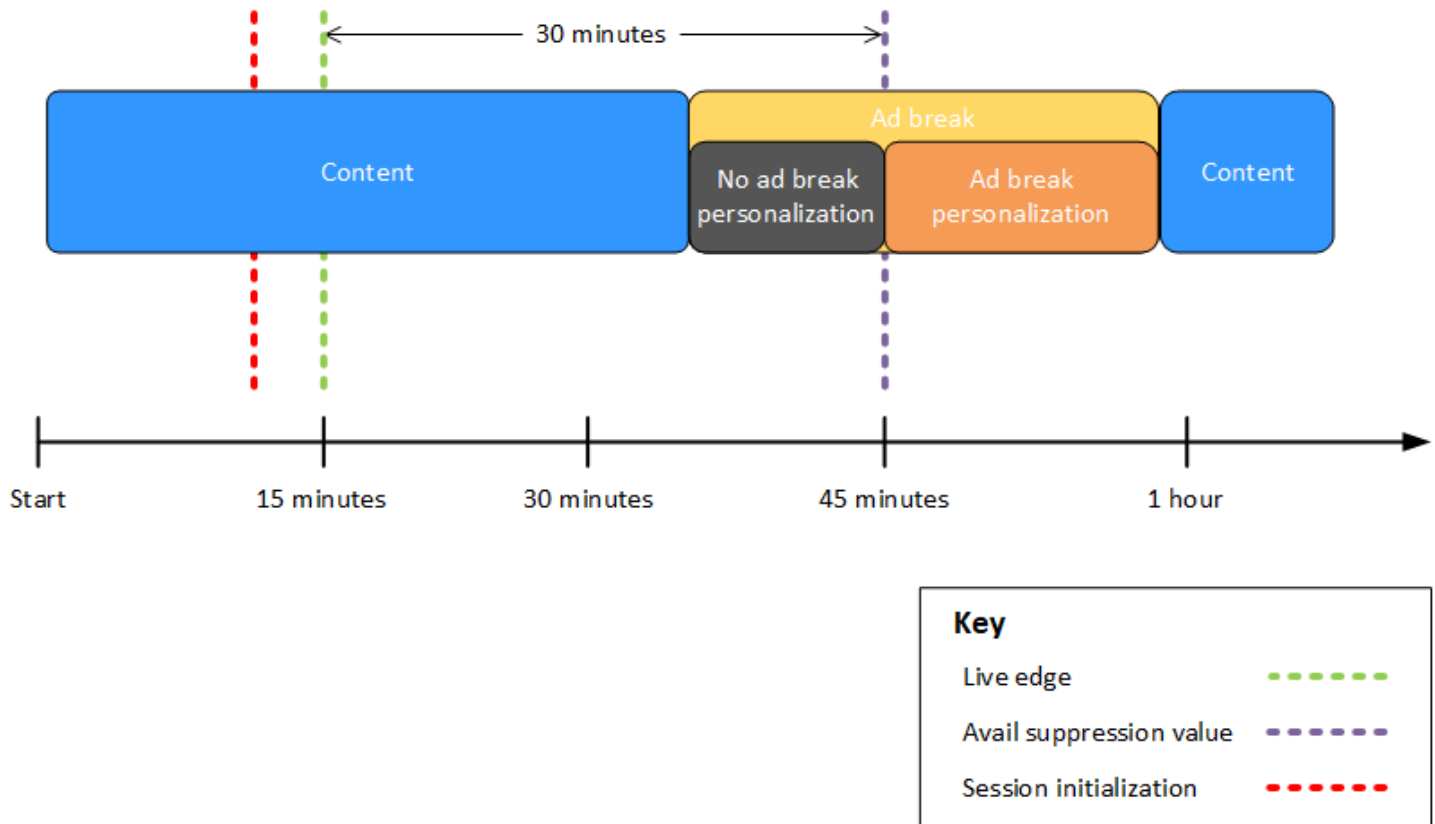
In der folgenden Abbildung sind verschiedene Blöcke horizontal entlang einer Zeitleiste angeordnet, die von links nach rechts verläuft. Jeder Block steht für einen Zeitraum, in dem der Inhalt eines

Live-Streams, einer personalisierten Werbepause oder einer nicht personalisierten Werbepause abgespielt wird. Eine gepunktete Linie steht für den aktuellen Live-Rand des Live-Streams. Eine weitere gepunktete Linie, die den eingestellten Wert für die Verfügbarkeitsunterdrückung darstellt 00:30:00, erscheint 30 Minuten später in der Timeline in Bezug auf die gepunktete Linie für den Live-Rand. Eine dritte gepunktete Linie, die für die Sitzungsinitialisierung steht, befindet sich in der Timeline an einem anderen Punkt als die gepunktete Linie für die Live-Kante. Der Zeitraum von 30 Minuten zwischen der Live-Edge-Zeit und der Uhrzeit stellt den Zeitraum für die Unterdrückung der avail-suppression-value Nutzung dar. Am Ende der Verfügbarkeitsunterbrechung wird eine Werbeunterbrechung ausgeführt. Wie in der Abbildung dargestellt, ist bei aktiviertem Verfügbarkeitsunterdrückungsmodus der Wert für die Verfügbarkeitsunterdrückung auf „00:30:00 nach dem Live-Edge“ eingestellt, die Richtlinie zum Füllen der Verfügbarkeitsunterdrückung ist auf eingestellt und die Sitzungsinitialisierung erfolgt vor dem Live-Edge *PARTIAL_AVAIL*, wodurch alle Werbeunterbrechungen MediaTailor personalisiert werden, die nach dem Verfügbarkeitsunterdrückungszeitraum auftreten. *AFTER_LIVE_EDGE* MediaTailor Personalisiert für die laufende Werbeunterbrechung am Ende des Verfügbarkeitsunterdrückungszeitraums den Teil dieser Werbeunterbrechung, der nach dem Verfügbarkeitsunterdrückungszeitraum erfolgt, personalisiert jedoch nicht den Teil dieser Werbeunterbrechung, der während des Verfügbarkeitsunterdrückungszeitraums stattfindet.

Avail suppression mode: **AFTER_LIVE_EDGE**

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

Avail suppression fill policy: **PARTIAL_AVAIL**



Example 6: **AFTER_LIVE_EDGE** Werbeunterbrechung mit der Richtlinie „**PARTIAL_AVAIL**Ausfüllen“ und einer aktiven Werbeunterbrechung von vor der Sitzungsinitialisierung bis nach dem Ende der Nutzungsunterbrechung

Wenn der Verfügbarkeitsunterdrückungsmodus auf eingestellt ist **AFTER_LIVE_EDGE** und der Wert für die Verfügbarkeitsunterbindung größer als Null ist, werden Werbeunterbrechungen erst personalisiert, wenn die Dauer der Sitzung diesen Wert erreicht MediaTailor hat.

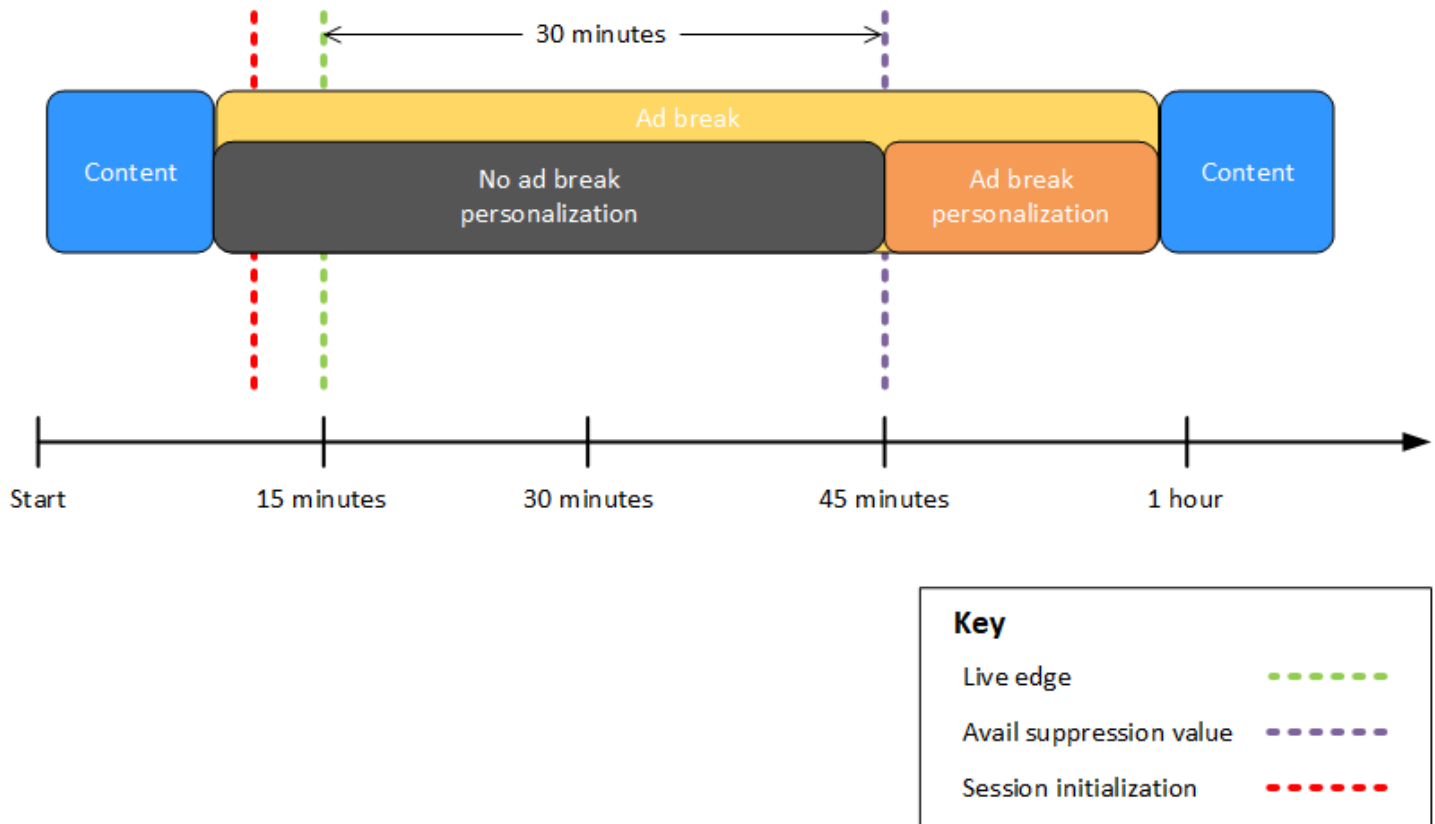
In der folgenden Abbildung sind verschiedene Blöcke horizontal entlang einer Zeitleiste angeordnet, die von links nach rechts verläuft. Jeder Block steht für einen Zeitraum, in dem der Inhalt eines Live-Streams, einer personalisierten Werbepause oder einer nicht personalisierten Werbepause abgespielt wird. Eine gepunktete Linie steht für den aktuellen Live-Rand des Live-Streams. Eine weitere gepunktete Linie, die den eingestellten Wert für die Verfügbarkeitsunterdrückung darstellt **00:30:00**, erscheint 30 Minuten später in der Timeline

in Bezug auf die gepunktete Linie für den Live-Rand. Eine dritte gepunktete Linie, die für die Sitzungsinitialisierung steht, befindet sich in der Timeline an einem anderen Punkt als die gepunktete Linie für die Live-Kante. Der Zeitraum von 30 Minuten zwischen der Live-Edge-Zeit und der Uhrzeit stellt den Zeitraum für die Unterdrückung der avail-suppression-value Nutzung dar. Eine Werbeunterbrechung wird von einem Zeitpunkt vor der Sitzungsinitialisierung bis zu einem Zeitpunkt nach dem Zeitraum nach dem Verfügbarkeitsunterdrückungszeitraum ausgeführt. Wie in der Abbildung dargestellt, wird, wenn der Avail-Unterdrückungsmodus auf eingestellt ist `AFTER_LIVE_EDGE`, der Wert für die Avail-Unterdrückung auf die Zeit `00:30:00` nach dem Live-Edge gesetzt, die Richtlinie zum Füllen der Verfügbarkeitsunterdrückung auf aktiviert ist und die Sitzungsinitialisierung vor dem Live-Edge erfolgt `PARTIAL_AVAIL`, MediaTailor personalisiert alle Werbeunterbrechungen, die nach dem Zeitraum der Verfügbarkeitsunterdrückung auftreten. MediaTailor Personalisiert für die Werbeunterbrechung, die vor, während und nach der Verfügbarkeitsunterdrückungszeit ausgeführt wird, den Teil dieser Werbeunterbrechung, der nach dem Verfügbarkeitsunterdrückungszeitraum erfolgt, personalisiert jedoch nicht den Teil dieser Werbeunterbrechung, der vor oder während des Verfügbarkeitsunterdrückungszeitraums stattfindet.

Avail suppression mode: **AFTER_LIVE_EDGE**

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

Avail suppression fill policy: **PARTIAL_AVAIL**



Example 7: **AFTER_LIVE_EDGE** Werbeunterbrechung, wenn zu Beginn des Zeitraums zur Unterdrückung der Verfügbarkeit eine Werbeunterbrechung im Gange ist

Wenn der Verfügbarkeitsunterdrückungsmodus auf eingestellt ist **AFTER_LIVE_EDGE** und der Wert für die Verfügbarkeitsunterbindung größer als Null ist, werden Werbeunterbrechungen erst personalisiert, wenn die verstrichene Zeit der Sitzung diesen Wert erreicht MediaTailor hat.

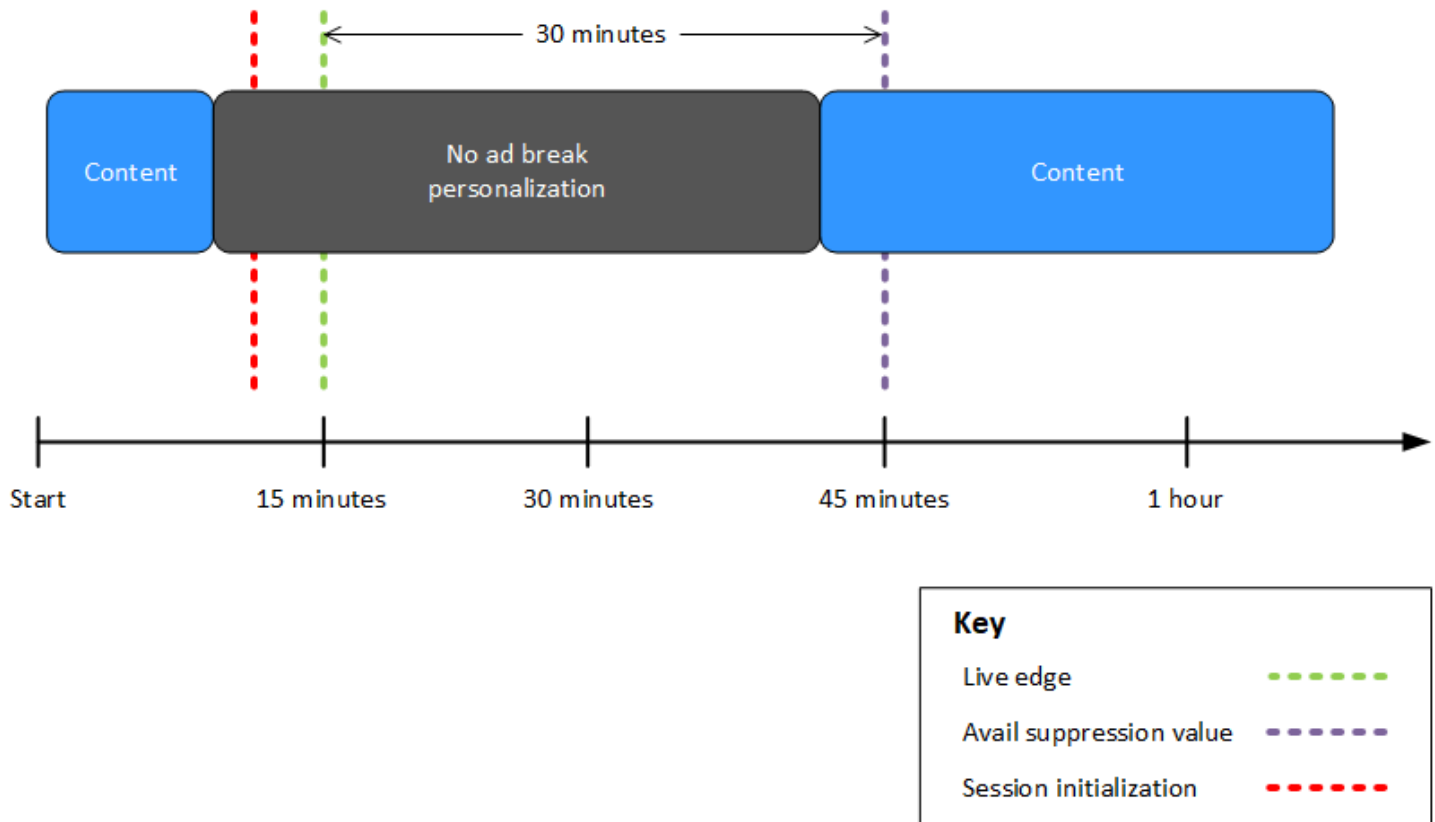
In der folgenden Abbildung sind verschiedene Blöcke horizontal entlang einer Zeitleiste angeordnet, die von links nach rechts verläuft. Jeder Block steht für einen Zeitraum, in dem der Inhalt des Live-Streams oder einer nicht personalisierten Werbepause wiedergegeben wird. Eine gepunktete Linie steht für den aktuellen Live-Rand des Live-Streams. Eine weitere gepunktete Linie, die den eingestellten Wert für die Verfügbarkeitsunterdrückung darstellt **00:30:00**, erscheint 30 Minuten später in der Timeline in Bezug auf die gepunktete Linie für den Live-Rand. Eine dritte gepunktete Linie, die für die Sitzungsinitialisierung steht, befindet sich in der Timeline an einem anderen Punkt

als die gepunktete Linie für die Live-Kante. Der Zeitraum von 30 Minuten zwischen der Live-Edge-Zeit und der Uhrzeit stellt den Zeitraum für die Unterdrückung der avail-suppression-value Nutzung dar. Eine Werbeunterbrechung wird von einem Zeitpunkt vor der Sitzungsinitialisierung bis zu einem Zeitpunkt innerhalb des Zeitraums der Verfügbarkeitsunterbrechung ausgeführt.

Wie in der Abbildung dargestellt, wird, wenn der Avail-Unterdrückungsmodus auf eingestellt ist `AFTER_LIVE_EDGE`, der Wert für die Avail-Unterdrückung auf `00:30:00` nach dem Live-Edge gesetzt, und die Sitzungsinitialisierung erfolgt vor der Live-Edge-Zeit, aber nach dem Beginn der Werbeunterbrechung, MediaTailor wodurch diese Werbeunterbrechung nicht personalisiert wird.

Avail suppression mode: `AFTER_LIVE_EDGE`

Avail suppression value: `00:30:00`



Konfigurieren von Werbeunterdrückungsparametern – Wiedergabe-Sitzungsanforderung

Sie können die Einstellungen für die Anzeigenunterdrückung über Parameter in Ihrer ersten server- oder clientseitigen Wiedergabe-Sitzungsanforderung an konfigurieren. MediaTailor Wenn Sie die Einstellungen zur Werbeunterdrückung bereits über die MediaTailor Konsole oder AWS Elemental MediaTailor API konfiguriert haben, haben diese Parameter Vorrang vor diesen Einstellungen.

Sowohl der Werbeplatz-Unterdrückungsmodus als auch der Werbeplatz-Unterdrückungswert sind erforderlich, damit die Werbeunterdrückung funktioniert. Diese Parameter können nicht aus verschiedenen Quellen konfiguriert werden. Sie können beispielsweise nicht einen Parameter mit der MediaTailor Konsole und einen anderen mit einem Abfrageparameter konfigurieren.

MediaTailor unterstützt die folgenden Parameter zur Anzeigenunterdrückung.

Name	Description	Akzeptierte Werte
<code>availSuppressionMode</code>	Legt den Modus für die Werbeunterdrückung fest. Die Standardeinstellung für die Anzeigenunterdrückung ist <code>OFF</code> . Wenn diese Option auf <code>BEHIND_LIVE_EDGE</code> eingestellt ist, werden Werbeunterbrechungen nicht <code>aws.availSuppressionValue</code> pünktlich oder nachträglich aufgefüllt. Wenn diese Option auf <code>AFTER_LIVE_EDGE</code> eingestellt ist, werden Werbeunterbrechungen während oder nach Ablauf der Sperrzeit nicht gefüllt. Der Zeitraum für die Unterdrückung der Verfügbarkeit reicht von der Live-Edge-Zeit bis zu diesem Zeitpunkt, zuzüglich zusätzlicher <code>aws.availSuppressionValue</code> Pufferzeit.	<code>OFF</code> <code>BEHIND_LIVE_EDGE</code> <code>AFTER_LIVE_EDGE</code>
<code>availSuppressionValue</code>	Ein Zeitpunkt relativ zur Live-Grenze in einem Livestream.	Ein URL-codierter UTF-8-Zeitcode in. <code>HH:MM:SS</code> Zum Beispiel wären 1 Stunde und 30 Minuten <code>01%3A30%3A00</code> .
<code>availSuppressionFullPolicy</code>	Definiert die Richtlinie, die auf den Avail-Unterdrückungsmodus angewendet werden soll. <code>BEHIND_LIVE_EDGE</code> verwendet immer die Richtlinie zur Unterdrückung des vollen Nutzens. <code>AFTER_LIVE_EDGE</code> kann verwendet werden, um das teilweise Auffüllen von Werbeunterbrechungen	<code>PARTIAL_Avail</code> - nicht verfügbar für den <code>BEFORE_LIVE_EDGE</code>

Name	Description	Akzeptierte Werte
	aufzurufen, wenn eine Sitzung mitten in der Pause beginnt.	Unterdrückungsmodus FULL_AVAI L_ONLY - der Standardwert für den AFTER_LIVE_EDGE Unterdrückungsmodus

Serverseitige Konfiguration

Der Basis-Abfrageparameter ist `aws.availSuppression`, gefolgt von optionalen Parameternamen/Werte-Paaren. Um die Abfrage zu erstellen, fügen Sie am Ende `aws.availSuppression=` der Wiedergabesitzung eine Anfrage an MediaTailor, gefolgt von Parameternamen und -werten. Weitere Hinweise zum Erstellen einer serverseitigen Wiedergabesitzungsanforderung finden Sie unter [MediaTailor Serverseitiges Ad-Tracking und Reporting](#).

Beispiel: HLS

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

Die serverseitige Abfragesyntax ist in der folgenden Tabelle aufgeführt.

Abfragezeichenfolgekomponente	Description
?	Ein eingeschränktes Zeichen, das den Anfang einer Abfrage markiert.
aws.	Die Basisabfrage, gefolgt von Parametern aus Name- und Wert-Paaren. Eine Liste aller verfügbaren Parameter finden Sie unter Konfigurieren von Werbeunterdrückungsparametern – Wiedergabe-Sitzungsanforderung .

Abfragezeichenfolge enkomponente	Description
=	Ordnet den Parameternamen einem Wert zu. Beispiel, <code>aws.availSuppressionMode= BEHIND_LIVE_EDGE</code> .
&	Verkettet Abfrageparameter. Beispiel: <code>aws.availSuppressionMode= BEHIND_LIVE_EDGE</code> und <code>aws.availSuppressionValue= 00:30:00&aws.availSuppressionFillPolicy= FULL_AVAIL_ONLY</code> >.

Clientseitige Konfiguration

Nehmen Sie `availSuppression` Parameter in die POST-Anfrage Ihres Kunden auf MediaTailor. Weitere Hinweise zum Erstellen einer clientseitigen Wiedergabebesitzungsanforderung finden Sie unter [Kundenseitiges Anzeigen-Tracking](#).

Beispiel: HLS

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

MediaTailor Stoßstange und Einlage

Bumper sind kurze, nicht überspringbare Video- oder Audioclips, die zu Beginn oder vor dem Ende einer Werbepause mit abgespielt werden. AWS Elemental MediaTailor

Für Bumper gelten die folgenden Bedingungen:

- Die Länge der Stoßstangen darf 10 Sekunden oder weniger betragen.
- Bumper können zu Beginn einer Werbeunterbrechung, direkt vor dem Ende einer Werbeunterbrechung oder an beiden Stellen eingefügt werden.

- Bumper werden bei jeder Werbeunterbrechung in einer Playback-Sitzung abgespielt, sofern Pre-Roll nicht konfiguriert ist. Wenn Pre-Roll konfiguriert ist, werden Bumper während der Pre-Roll-Pause nicht abgespielt. Stattdessen werden sie in jeder nachfolgenden Pause nach dem Pre-Roll abgespielt.
- Für HLS müssen Sie das `duration` Attribut jedem EXT-X-CUE-OUT SCTE-35-Tag beifügen.
- Bumper werden so transcodiert, dass sie dem Quellinhalt entsprechen.
- Für Bumper werden Ihnen keine Gebühren berechnet.

Stoßstangen konfigurieren

Um Bumper zu verwenden, konfigurieren Sie den Bumper URLs mit der MediaTailor Konsole, der MediaTailor API oder der AWS Command Line Interface ().AWS CLI Sie können einen Start-Bumper, einen End-Bumper oder beides konfigurieren. Bumper werden auf einem Server wie Amazon Simple Storage Service (Amazon S3) gespeichert. Die Bumper URLs geben den Standort der gespeicherten Bumper-Asset (s) an.

Beispiel für Start- und Endstoßstange: URLs

URL für den Start-Bumper: `https://s3.amazonaws.com/startbumperad`

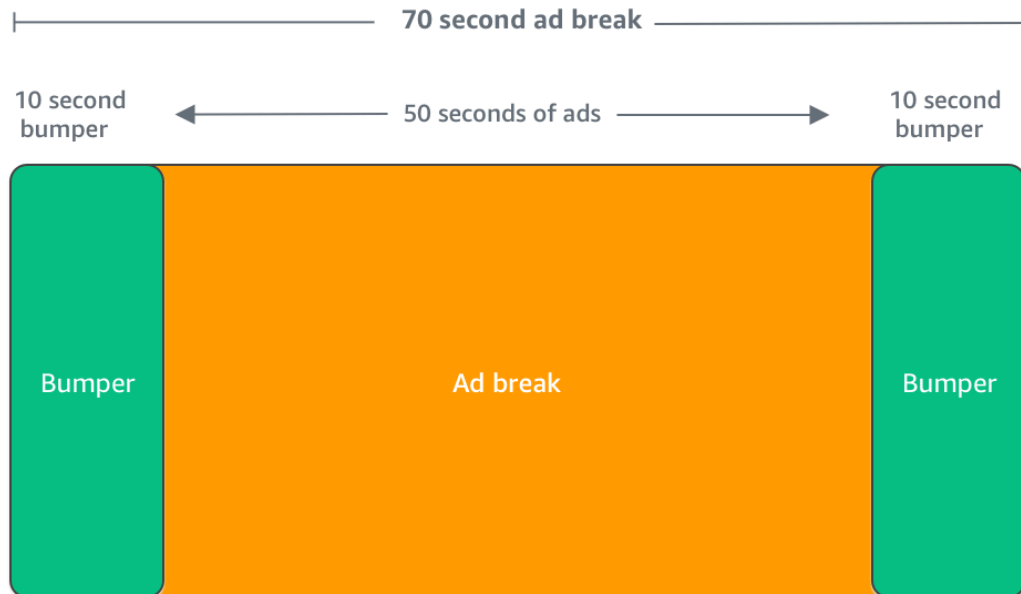
URL für das Ende des Bumpers: `https://s3.amazonaws.com/endbumperad`

Beispiel

Im Folgenden finden Sie ein Beispiel für das Verhalten von Bumper-Anzeigen.

Example Beispiel 1: Bumper am Anfang und Ende

In diesem Beispiel sind die Start- und Endstoßstangen aktiviert. Der Ad Decision Server verfügt über 50 Sekunden mit personalisierten Anzeigen, um eine 70-sekündige Werbepause zu füllen. Der 10-Sekunden-Start-Bumper wird zu Beginn der Werbepause abgespielt, 50 Sekunden lang werden Anzeigen abgespielt, danach der 10-Sekunden-End-Bumper.



MediaTailor Einfügen von Pre-Roll-Anzeigen

Note

Konfigurierbare Pre-Roll-Anzeigen sind nur für Live-Workflows verfügbar. Einzelheiten dazu, wie das Einfügen von Anzeigen (einschließlich Pre-Roll) für VOD funktioniert, finden Sie unter [Verhalten beim Zusammenfügen von Anzeigen bei VOD](#)

AWS Elemental MediaTailor kann Anzeigen zu Beginn einer Playback-Sitzung einfügen, bevor der Hauptinhalt beginnt. Dies sind Pre-Roll-Werbespots.

Um Pre-Roll-Werbung einzufügen, füllen Sie die Felder Live pre-roll ad decision server (Live Pre-Roll-Ad Decision Server) und Live pre-roll maximum allowed duration (Maximal zulässige Live-Pre-Roll-Dauer) in den Einstellungen unter Additional (Zusätzlich) für Ihre Konfiguration aus, wie unter [Optionale Konfigurationseinstellungen](#) beschrieben.

1. Wenn es eine Wiedergabeanforderung MediaTailor erhält, sendet es eine Anfrage an das ADS für Pre-Roll-Anzeigen, die auf den folgenden Feldern in der MediaTailor Wiedergabekonfiguration basiert:
 - Der Live-Entscheidungsserver für Pre-Roll-Anzeigen ist die URL des Ad Decision Servers (ADS), über den die Anfrage für Pre-Roll-Anzeigen MediaTailor gesendet wird.
 - Die maximal zulässige Dauer von Live-Pre-Roll-Anzeigen ist die maximale Gesamtdauer der Pre-Roll-Anzeigen. MediaTailor ergreift auf der Grundlage der maximal zulässigen Dauer die folgenden Maßnahmen:
 - Wenn die Gesamtdauer der Anzeigen in der ADS-Antwort unter dem Wert liegt, den Sie unter Höchstdauer des Live-Pre-Rolls angegeben haben, werden alle Anzeigen MediaTailor eingefügt. Wenn die letzte Anzeige fertig ist, wird MediaTailor sofort zum zugrunde liegenden Inhalt zurückgekehrt.
 - Wenn die Gesamtdauer der Anzeigen in der ADS-Antwort den Wert übersteigt, den Sie in der Live-Pre-Roll-Höchstdauer angegeben haben, MediaTailor wählt eine Gruppe von Anzeigen aus, die in die Dauer passen, ohne sie zu überschreiten. MediaTailor fügt diese Anzeigen ohne Kürzung oder Kürzung ein. MediaTailor kehrt zum zugrunde liegenden Inhalt zurück, wenn die zuletzt ausgewählte Anzeige abgeschlossen ist.
2. Wenn die Pre-Roll-Antwort vom ADS MediaTailor erhält, manipuliert es das Manifest, um Links zu den Pre-Roll-Anzeigen hinzuzufügen. MediaTailor berechnet die Startzeit der Pre-Roll-Werbepause wie folgt:
 - Für DASH lautet die Formel $(publishTime - availabilityStartTime) - \max(suggestedPresentationDelay, minBufferTime)$.
 - Für HLS lautet die Formel $\max(2 * EXT-X-TARGETDURATION, EXT-X-START:TIMEOFFSET)$.
3. MediaTailor bestimmt, welche Maßnahmen bei Werbeunterbrechungen ergriffen werden sollen, bei denen es sich nicht um Pre-Rolls handelt. Wenn sich die Pre-Roll mit einer anderen Werbepause überschneidet, MediaTailor wird der überlappende Teil der Werbepause nicht personalisiert.

MediaTailor Schiefer und Einfügung

Note

Slate ist nur für Live- und live-to-VOD Workflows verfügbar.

Mit AWS Elemental MediaTailor können Sie eine Slate-Anzeige für Werbeunterbrechungen festlegen. Ein Slate ist ein MP4 Standard-Asset, das in einen Stream eingefügt wird, z. B. ein Standbild oder ein Video in einer Endlosschleife, der anstelle des Live-Inhalts abgespielt wird.

AWS Elemental MediaTailor zeigt Slate während Werbeunterbrechungen in den folgenden spezifischen Situationen an:

- Um die verbleibende Zeit in einer Pause aufzufüllen, die nicht durch personalisierte Werbung ersetzt wurde
- Wenn der Ad Decision Server (ADS) mit einer leeren VAST- oder VMAP-Antwort antwortet
- Bei Fehlerbedingungen wie ADS-Timeout oder HTTP 500-Fehler vom ADS
- Wenn eine Anzeige nicht zum Einfügen verfügbar ist MediaTailor (z. B. wenn die Transcodierung noch nicht abgeschlossen ist)

Wenn Sie kein Slate konfigurieren, MediaTailor wird standardmäßig der zugrunde liegende Inhaltsstream verwendet, wenn eine der oben genannten Bedingungen erfüllt ist.

Das Slate konfigurieren

Sie bestimmen das Slate im Bereich für zusätzliche Konfigurationen in der [MediaTailor Konsole](#). MediaTailor lädt das Slate von der von Ihnen angegebenen URL herunter und transkodiert es in dieselben Wiedergabeversionen wie Ihr Inhalt. Über die optionale Konfiguration des Schwellenwerts für Personalisierung in der Konsole können Sie festlegen, wie lange ein Slate maximal angezeigt wird. MediaTailor Weitere Informationen finden Sie unter [the section called “So funktioniert der Schwellenwert für Personalisierung”](#).

Bei dem Slate muss es sich um ein qualitativ hochwertiges MP4 Asset handeln, das sowohl Audio als auch Video enthält. Die Konfiguration des Slate ist für Konfigurationen ohne VPaid optional, für VPAID-Workflows jedoch obligatorisch.

Note

Wenn der Server, auf dem der Pausenfüller gehostet wird, HTTPS verwendet, muss sein Zertifikat von einer bekannten Zertifizierungsstelle stammen. Es kann kein selbstsigniertes Zertifikat sein. Wenn Sie ein selbstsigniertes Zertifikat verwenden, AWS Elemental MediaTailor können Sie das Slate aus dem Inhaltsursprung nicht abrufen und in die Manifeste einbinden.

So funktioniert der Schwellenwert für Personalisierung

Der Schwellenwert für die Personalisierung definiert die maximale Dauer einer unterfüllten Werbezeit (in Sekunden), die in einer Werbepause zulässig ist. Diese Funktion bezieht sich speziell auf das Ersetzen von Anzeigen in Live- und VOD-Streams und nicht auf das Einfügen von Anzeigen, da sie auf einem zugrunde liegenden Inhaltsstream basiert.

Das Verhalten variiert je nach drei Szenarien:

1. Wenn die Personalisierung deaktiviert ist:
 - a. Slate wird für die gesamte Dauer der nicht ausgefüllten Zeit eingefügt
 - b. Bei der Konfiguration werden Start-/Endpuffer eingefügt (weitere Informationen finden Sie unter [MediaTailor Stoßstange und Einlage](#))
 - c. Werbeanzeigen werden wie gewohnt eingefügt
2. Wenn die Personalisierung aktiviert ist und der Schwellenwert unter der Pausendauer liegt:
 - a. Wenn die ungenutzte Zeit den Schwellenwert für die Personalisierung überschreitet:
 - i. MediaTailor gibt die Personalisierung der Werbepause auf
 - ii. Der zugrundeliegende Inhalt wird angezeigt
 - iii. Es sind keine Werbung, Schiefer oder Stoßstangen eingefügt
 - b. Wenn die ungenutzte Zeit unter dem Schwellenwert für die Personalisierung liegt:
 - i. Werbung und Slate werden eingefügt
 - ii. Stoßstangen werden eingefügt, sofern konfiguriert
3. Wenn die Personalisierung aktiviert ist und der Schwellenwert größer als die Pausendauer ist:
 - a. Werbeanzeigen werden eingefügt
 - b. Slate wird für die verbleibende Zeit in der Werbepause eingefügt
 - c. Bumper werden eingefügt, sofern sie konfiguriert sind

Die folgende Tabelle enthält detaillierte Verhaltensmuster für bestimmte Szenarien in den Protokollen HLS und DASH:

Detaillierte Verhaltensmatrix

Szenario	Personalisierung deaktiviert	Personalisierung wurde für weniger Zeit als für die Dauer der Pause aktiviert	Personalisierung länger als Pausendauer aktiviert
Leeres VAST oder VMAP	Schiefer eingefügt	Keine Schiefer eingefügt	Schiefer eingefügt
ADS-Timeout	Schiefer eingefügt	Keine Schiefer eingefügt	Schiefer eingefügt
Anzeige ist nicht verfügbar (Vast 404)	Schiefer eingefügt	Keine Schiefer eingefügt	Schiefer eingefügt
Die Dauer der Anzeigen ist länger als die Werbepause	Eingefügte Anzeigen	Keine Werbung eingefügt	Anzeigen eingefügt
Geben Sie die Zeit ein, die durch eine Ersatzanzeige nicht vollständig genutzt wurde	Schiefer eingefügt	Wenn der Schwellenwert für die Personalisierung höher ist als die Leerlaufzeit: Nein ads/slides eingefügt, sonst: Werbeanzeigen und Schieferlisten eingefügt	Schiefertafel eingefügt
Preroll mit leerem VAST	Schieferplatte eingefügt, kein Preroll eingefügt	Keine Schiefer eingelegt, keine Vorwalze eingelegt	Schiefer eingelegt, keine Vorwalze eingelegt

Szenario	Personalisierung deaktiviert	Personalisierung wurde für weniger Zeit als für die Dauer der Pause aktiviert	Personalisierung länger als Pausendauer aktiviert
Preroll mit ADS-Timeout	Slate eingefügt , kein Preroll eingefügt	Keine Schiefer eingelegt, keine Vorwalze eingelegt	Schiefer eingelegt, keine Vorwalze eingelegt
Preroll, wenn die Anzeige nicht verfügbar ist (Vast 404)	Slate eingefügt , kein Preroll eingefügt	Keine Schiefer eingelegt, keine Vorwalze eingelegt	Schiefer eingelegt, keine Vorwalze eingelegt
Stoßstangen mit leerem VAST	Schiefer eingesetzt, Stoßstange eingesetzt	Keine Schiefer eingesetzt, keine Stoßstangen eingesetzt	Schiefer eingesetzt, Stoßstange eingesetzt
Stoßstangen mit ADS-Timeout	Schiefer eingesetzt, Stoßstange eingesetzt	Keine Schiefer eingesetzt, keine Stoßstangen eingesetzt	Schiefer eingesetzt, Stoßstange eingesetzt
Bumpers, wenn die Anzeige nicht verfügbar ist (Vast 404)	Schiefer eingesetzt, Stoßstange eingesetzt	Keine Schiefer eingesetzt, keine Stoßstangen eingesetzt	Schiefer eingesetzt, Stoßstange eingesetzt

Szenario	Personalisierung deaktiviert	Personalisierung wurde für weniger Zeit als für die Dauer der Pause aktiviert	Personalisierung länger als Pausendauer aktiviert
Stoßstangen zum Einfüllen von Zeit, die beim Austausch nicht vollständig genutzt wird	Schiefer eingefügt, Stoßstange eingesetzt, Anzeige eingefügt	Wenn der Schwellenwert für die Personalisierung höher ist als die ungenutzte Zeit: Nein eingefügt bumpers/ads/slates inserted, else: No bumpers/ads/slates	Schiefer eingefügt, Stoßstange eingefügt, Anzeige eingefügt

Wichtige Überlegungen

Beachten Sie Folgendes, wenn Sie mit Slate und Personalization Threshold arbeiten.

- Dieses Verhalten ist sowohl für das HLS- als auch für das DASH-Protokoll einheitlich
- Die Funktion für den Personalisierungsschwellenwert gilt nur, wenn Slate konfiguriert ist
- Wenn Slate konfiguriert ist und der Schwellenwert für Personalisierung nicht konfiguriert ist, wird Slate entweder so lange abgespielt, wie lange die Originalversion verfügbar ist oder wie lange es nach dem Ausfüllen der Anzeigen noch übrig ist
- MediaTailor Fügt bei VPAID-Anzeigen Slate für die Dauer der VPAID-Anzeige ein, um Platz für Anzeigen zu bieten, die der Videoplayer einfügt

Slate-Konfiguration und VPAID

Important

Bei der Verwendung von VPAID-Anzeigen ist eine Slate-Konfiguration erforderlich. MediaTailor fügt Slate ein, um Platz für VPAID-Anzeigen zu bieten, die der Videoplayer einfügt. Die Slate-Dauer kann etwas länger sein als die Dauer der VPAID-Anzeige, um der Nutzerinteraktivität Rechnung zu tragen.

Der Videoplayer verarbeitet die VPAID-Anzeige dann auf der Grundlage der MediaTailor zurückgegebenen clientseitigen Berichtsmetadaten, wie unter beschrieben. [VPAID-Anforderungen](#) Weitere Informationen zur clientseitigen Berichterstellung finden Sie unter [the section called “Kundenseitiges Tracking”](#).

Anzeigen vorab abrufen

Verwenden Sie den AWS Elemental MediaTailor Anzeigenvorabruf für Livestreams, um die Spitzenlast auf Ad Decision Servern (ADS) zu reduzieren und die Latenz bei der Manifestzustellung zu Beginn jeder Werbeunterbrechung zu verringern. Wenn Sie einen Zeitplan für den Vorabruf definieren, MediaTailor folgt dieser Zeitplan, um Anzeigen aus dem ADS abzurufen und sie für die Anzeigenschaltung vorzubereiten, bevor sie für eine Werbeunterbrechung benötigt werden. Bei Livestreams kann das Prefetching dazu beitragen, verringerte Anzeigenfüllraten und verpasste Monetarisierungsmöglichkeiten aufgrund von Timeouts bei Anzeigenanfragen und Transcodierung oder anderen Netzwerkverzögerungen zu vermeiden.

Note

Der Vorabruf von Anzeigen funktioniert nicht mit Methoden zur servergesteuerten Anzeigeneinfügung (SGAI), einschließlich herkömmlicher servergeführter Methoden und HLS-Interstitials. Bei SGAI-Methoden ist kein Prefetching erforderlich, da Spieler nur Anzeigen abrufen, die sie auch abspielen werden, und Manifeste können bereitgestellt werden, ohne dass einzelne Sitzungsanfragen angezeigt werden. CDNs MediaTailor

Um den Vorabruf von Anzeigen einzurichten, erstellen Sie in Ihrer Wiedergabekonfiguration einen oder mehrere Prefetch-Zeitpläne. Ein Prefetch-Zeitplan gibt an, MediaTailor wie und wann Werbeanzeigen abgerufen und für eine bevorstehende Werbepause vorbereitet werden müssen.

- Wenn für eine Veranstaltung Werbeanzeigen verfügbar sind, die nach einem vorhersehbaren Zeitplan verfügbar sind, verwenden Sie einen einzigen Prefetch-Zeitplan. Jeder einzelne Prefetch-Zeitplan definiert eine einzelne Gruppe von Anzeigen, die in einer einzigen Anzeigenauswahl platziert MediaTailor werden sollen. Um Anzeigen für mehrere verfügbare Anzeigen vorab abzurufen, wenn Sie einzelne Prefetch-Zeitpläne verwenden, müssen Sie mehrere Prefetch-Zeitpläne (bis zu 24 Stunden vor der Anzeigenbereitstellung) erstellen, die für jede verfügbare Anzeige relevant sind.
- Wenn für eine Veranstaltung Werbeanzeigen verfügbar sind, die nicht in einem vorhersehbaren Zeitplan verfügbar sind, verwenden Sie einen wiederkehrenden Prefetch-Zeitplan. Ein wiederkehrender Prefetch-Zeitplan erstellt automatisch einen Zeitplan und ruft Anzeigen vor jeder Werbeunterbrechung in einem Event vorab ab. Im Zeitplan für wiederkehrende Prefetches werden Anzeigen für jede verfügbare Anzeige innerhalb eines bestimmten Zeitraums abgerufen (bis zu 24 Stunden vor Ende der Veranstaltung). Sie müssen nicht für jede verfügbare Anzeige einen Zeitplan erstellen, verlieren jedoch einen Teil der zeitlichen Kontrolle, die ein einzelner Prefetch-Vorgang bietet.

In den folgenden Themen wird der Vorabruf von Anzeigen näher beschrieben.

Themen

- [So funktioniert der Vorabruf](#)
- [Prefetch-Zeitpläne erstellen](#)
- [TPS-basiertes Traffic-Shaping](#)
- [Prefetch-Zeitpläne werden gelöscht](#)

So funktioniert der Vorabruf

Wenn Ihr Client eine Manifestanforderung an stellt MediaTailor, wertet der Service alle Prefetch-Zeitpläne aus, die mit der Wiedergabekonfiguration verknüpft sind. Wenn MediaTailor kein passender Prefetch-Zeitplan gefunden wird, kehrt der Dienst zur normalen Anzeigeneinfügung zurück und ruft keine Anzeigen vorab ab.

Wenn ein MediaTailor passender Prefetch-Zeitplan gefunden wird, bewertet der Dienst den Zeitplan anhand von zwei Komponenten: Abruf und Nutzung. Die Konfiguration der einzelnen Komponenten variiert zwischen einzelnen Prefetch-Zeitplänen und wiederkehrenden Prefetch-Zeitplänen, wie in den folgenden Abschnitten beschrieben.

Ablauf eines einzelnen Prefetch-Zeitplans

Abruf

Dadurch wird das Abrufenster definiert. Dabei handelt es sich um den Zeitraum, in dem Anzeigen MediaTailor vorab aus dem ADS abgerufen werden. Achten Sie darauf, dieses Fenster für einen Zeitraum vor der Werbeunterbrechung einzuplanen. Im Folgenden finden Sie einen Überblick darüber, wie ein einzelner Prefetch-Zeitplan MediaTailor verarbeitet wird.

Die Schritte zum Erstellen eines einzelnen Prefetch-Zeitplans in der Konsole finden Sie unter [Prefetch-Zeitpläne erstellen](#). Eine API-Anleitung finden Sie [PrefetchSchedules](#) in der AWS Elemental MediaTailor API-Referenz.

MediaTailor Sendet während des angegebenen Abrufensters Anfragen an das ADS, um Anzeigen abzurufen und für die spätere Einfügung in Wiedergabe-Sitzungen vorzubereiten.

- MediaTailor verwendet optional Traffic Shaping, um die Anzahl der Anfragen an das ADS gleichzeitig zu begrenzen. Sie können zwischen zwei Ansätzen wählen:

Traffic Shaping im Zeitfenster — MediaTailor verteilt Anfragen über die angegebene Anzahl von Sekunden, anstatt Anfragen für alle Sitzungen gleichzeitig zu senden. Diese verteilte Verteilung des Datenverkehrs trägt dazu bei, dass das ADS nicht überlastet wird, was zu Timeouts und niedrigen Ausfüllraten von Anzeigen führt.

TPS-basiertes Traffic-Shaping — MediaTailor begrenzt Anfragen auf der Grundlage von Transaktionen pro Sekunde (TPS) und gleichzeitigen Benutzern. Dieser Ansatz ermöglicht eine intuitivere Konfiguration auf der Grundlage Ihrer ADS-Kapazitätsgrenzen. Weitere Informationen finden Sie unter [TPS-basiertes Traffic-Shaping](#).

- Wenn Sie dynamische Variablen einrichten, werden MediaTailor diese Variablen in die Anfragen an das ADS aufgenommen. MediaTailor verwendet diese Variablen, um Zeitpläne für den Vorabruf während des Nutzungsfensters abzugleichen und zu nutzen. Weitere Informationen finden Sie im folgenden Abschnitt zum Verbrauch.

Example

Eine Live-Veranstaltung dauert von 7:45 Uhr bis 10 Uhr, mit einer Werbepause um 8:15 Uhr. Sie konfigurieren MediaTailor den Abruf von Werbeanzeigen von 7:45 Uhr bis 8 Uhr mit einem Traffic-Shaping-Zeitfenster von 60 Sekunden. Bei 500.000 gleichzeitigen Benutzern werden die ADS-Anfragen so MediaTailor verteilt, dass eine durchschnittliche Rate von etwa 8.333 Transaktionen

pro Sekunde für 60 Sekunden (500.000 Benutzer/60 Sekunden = 8.333 Anfragen pro Sekunde) erreicht wird, anstatt alle Anfragen gleichzeitig zu senden.

Die Abrufkonfiguration umfasst die dynamische Variable Schlüssel und Wert. `scte.event 1234` MediaTailor nimmt diese Variable in die Anfragen an die ADS auf, die dann verwendet werden können, um bestimmte Werbetreibende auf die Event-ID 1234 auszurichten.

Verbrauch

Wenn Sie während des Nutzungsfensters MediaTailor auf SCTE-35-Werbeunterbrechungsmarken stoßen, werden die vorab abgerufenen Anzeigen in einer Werbeunterbrechung platziert.

- Wenn Sie keine Auswahlkriterien festgelegt haben, werden Anzeigen in der ersten Pause des Werbefensters MediaTailor eingefügt.
- Wenn Sie einen dynamischen Variablenschlüssel für verfügbare Übereinstimmungskriterien festgelegt haben, werden diese Kriterien anhand der dynamischen Variablen MediaTailor bewertet, die Sie im Abruffenster festgelegt haben. Eine Werbeunterbrechung kann nur dann vorab abgerufen werden, wenn die Auswahlkriterien erfüllt sind. MediaTailor fügt Anzeigen in die erste Pause ein, die die Kriterien erfüllt.

Eine Liste der unterstützten Kriterien für den Abgleich von Verfügbarkeiten finden Sie in der Tabelle unter In der Spalte Verfügbar für Anzeigenvorabruf. [MediaTailor Sitzungsvariablen für ADS-Anfragen](#)

Example fortgesetzt

Sie haben die Startzeit für den Konsum auf 8:15 Uhr und die Endzeit auf 8:17 Uhr festgelegt. Sie geben `scte.event_id` für den Schlüssel in die Verfügbarkeit die passenden Kriterien an.

Für jede Werbeunterbrechung, MediaTailor die zwischen 8:15 Uhr und 8:17 Uhr stattfindet, wird die SCTE Event-ID für jede Werbeunterbrechung ausgewertet. MediaTailor Fügt in jeder Wiedergabe-Sitzung die vorab abgerufenen Anzeigen in die erste Werbeunterbrechung ein, die die Event-ID 1234 hat (wie in den dynamischen Variablen für den Abruf definiert). MediaTailorFührt bei Werbeunterbrechungen, die nicht die richtige Event-ID enthalten, die Standardanzeigeneinfügung durch.

Ablauf eines wiederkehrenden Prefetch-Vorabrufs

Abruf

Dadurch wird das Zeitfenster für wiederkehrende Abrufe definiert. Dabei handelt es sich um den Zeitraum, in dem Anzeigen für ein Live-Event MediaTailor vorab abgerufen und eingefügt werden (bis zu 24 Stunden). Im Folgenden finden Sie einen Überblick darüber, wie wiederkehrende MediaTailor Prefetch-Zeitpläne verarbeitet werden.

Schritte zum Erstellen eines wiederkehrenden Prefetch-Zeitplans in der Konsole finden Sie unter [Prefetch-Zeitpläne erstellen](#). Eine API-Anleitung finden Sie [PrefetchSchedules](#) in der AWS Elemental MediaTailor API-Referenz.

MediaTailor ruft während des angegebenen wiederkehrenden Prefetch-Fensters Anzeigen für ein Live-Event ab und fügt sie ein, das bis zu 24 Stunden dauert. Ruft nach jeder Werbeunterbrechung im Fenster MediaTailor automatisch Anzeigen für die nächste Werbeunterbrechung ab.

- Wenn Sie die Verzögerung nach Ende der Verfügbarkeit festlegen, wird die angegebene Zeit MediaTailor gewartet, bevor die nächste Anzeigengruppe für die nächste Werbepause abgerufen wird.
- MediaTailor verwendet optional Traffic Shaping, um die Anzahl der Anfragen an das ADS gleichzeitig zu begrenzen. Sie können zwischen zwei Ansätzen wählen:

Traffic Shaping im Zeitfenster — MediaTailor verteilt Anfragen über die angegebene Anzahl von Sekunden, anstatt Anfragen für alle Sitzungen gleichzeitig zu senden. Diese verteilte Verteilung des Datenverkehrs trägt dazu bei, dass das ADS nicht überlastet wird, was zu Timeouts und niedrigen Ausfüllraten von Anzeigen führt.

TPS-basiertes Traffic-Shaping — MediaTailor begrenzt Anfragen auf der Grundlage von Transaktionen pro Sekunde (TPS) und gleichzeitigen Benutzern. Dieser Ansatz ermöglicht eine intuitivere Konfiguration auf der Grundlage Ihrer ADS-Kapazitätsgrenzen. Weitere Informationen finden Sie unter [TPS-basiertes Traffic-Shaping](#).

- Wenn Sie dynamische Variablen einrichten, werden MediaTailor diese Variablen in die Anfragen an das ADS aufgenommen. MediaTailor verwendet diese Variablen, um Zeitpläne für den Vorabruf während des Nutzungsfensters abzugleichen und zu nutzen. Weitere Informationen finden Sie im folgenden Abschnitt zum Verbrauch.

Example

Eine Live-Veranstaltung dauert von 19 Uhr bis 20:45 Uhr, mit vier Werbeunterbrechungen in dieser Zeit. Die Werbeunterbrechungen haben keinen vorhersehbaren Zeitplan. Sie konfigurieren den wiederkehrenden Prefetch von 19:00 Uhr bis 20:45 Uhr mit einer Verzögerung von 10 Minuten und einem Traffic-Shaping-Fenster von 60 Sekunden. MediaTailor ruft nach jedem Zugriff Anzeigen für die nächste Werbepause ab. Zehn Minuten nach Ende der Verfügbarkeit beginnt MediaTailor das Senden von Abrufanfragen an das ADS. Bei einem Zeitfenster von 60 Sekunden und 500.000 gleichzeitigen Benutzern werden die ADS-Anfragen so MediaTailor verteilt, dass eine durchschnittliche Rate von etwa 8.333 Transaktionen pro Sekunde für 60 Sekunden ($500.000 \text{ Benutzer} / 60 \text{ Sekunden} = 8.333 \text{ Anfragen pro Sekunde}$) erreicht wird, anstatt alle Anfragen gleichzeitig zu senden.

Die `scte.event 1234` Abrufkonfiguration umfasst die dynamische Variable Schlüssel und Wert. MediaTailor nimmt diese Variable in die Anfragen an die ADS auf, die dann verwendet werden können, um bestimmte Werbetreibende auf die Event-ID 1234 auszurichten.

Verbrauch

Wenn MediaTailor SCTE-35-Markierungen für Werbeunterbrechungen gefunden werden, werden die vorab abgerufenen Anzeigen in einer Werbeunterbrechung platziert.

- Wenn Sie das Ablaufdatum für abgerufene Anzeigen festlegen, können vorab abgerufene Anzeigen bis zum angegebenen Ablaufdatum eingefügt werden.
- Wenn Sie keine Auswahlkriterien für die Verfügbarkeit festgelegt haben, werden Anzeigen in der ersten Pause des Nutzungsfensters MediaTailor eingefügt.
- Wenn Sie einen dynamischen Variablenschlüssel für verfügbare Übereinstimmungskriterien festgelegt haben, werden diese Kriterien anhand der dynamischen Variablen MediaTailor bewertet, die Sie im Abruffenster festgelegt haben. Eine Werbeunterbrechung kann nur dann vorab abgerufen werden, wenn die Auswahlkriterien erfüllt sind. MediaTailor fügt Anzeigen in die erste Pause ein, die die Kriterien erfüllt.

Eine Liste der unterstützten Kriterien für den Abgleich von Verfügbarkeiten finden Sie in der Tabelle unter In der Spalte Verfügbar für Anzeigenvorabruf. [MediaTailor Sitzungsvariablen für ADS-Anfragen](#)

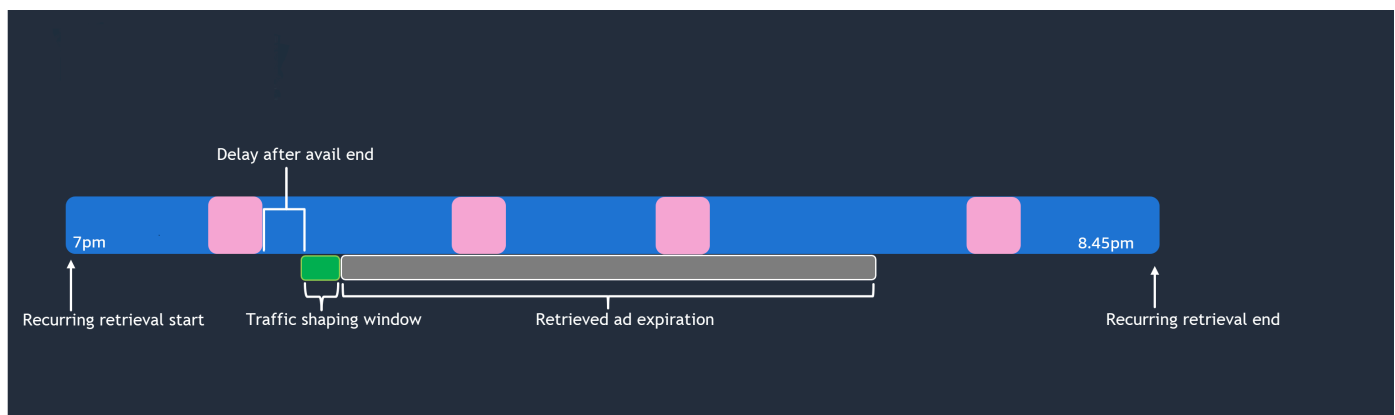
Example fortgesetzt

In den Verbrauch beziehen Sie `scte.event_id` für den Schlüssel in die Verfügbarkeit passende Kriterien mit ein.

Für jede Werbeunterbrechung, MediaTailor auf die Sie stoßen, wird die SCTE Event-ID für jede Werbeunterbrechung ausgewertet. MediaTailor fügt in jeder Wiedergabe-Sitzung die vorab abgerufenen Anzeigen in jede Werbeunterbrechung ein, die die Event-ID 1234 hat (wie in den dynamischen Variablen für den Abruf definiert). MediaTailor führt bei Werbeunterbrechungen, die nicht die richtige Event-ID enthalten, die Standardanzeigeneinfügung durch.

Sie haben den Ablauf der Anzeige auf 2700 Sekunden festgelegt, sodass die abgerufenen Anzeigen 45 Minuten lang eingeblendet werden können.

Die folgende Grafik veranschaulicht das Beispiel, wobei die kleinen Quadrate Werbeunterbrechungen darstellen. Die Einstellungen für den Zeitplan für wiederkehrende Prefetches werden entlang der Event-Zeitleiste dargestellt.



Grundlegendes zu den Kosten für das Prefetching

Für Anfragen zum Abrufen von Anzeigen fallen keine Kosten an. Für den Abruf von Prefetch-Anzeigen wird Ihnen jedoch der Standardtarif für die transkodierte Prefetch-Anzeigen berechnet. MediaTailor für die Nutzung von Prefetch-Anzeigen, die in Werbeunterbrechungen platziert werden, wird Ihnen der Standardtarif für die Anzeigeneinblendung berechnet. MediaTailor [Informationen zu den Kosten für Transcodierung und Anzeigenschaltung finden Sie unter Preise.AWS Elemental MediaTailor](#)

Prefetch-Zeitpläne erstellen

Im folgenden Verfahren wird erklärt, wie Sie mithilfe der Konsole einen Prefetch-Zeitplan erstellen. MediaTailor Informationen zum programmgesteuerten Erstellen und Verwalten von Prefetch-Zeitplänen mithilfe der MediaTailor API finden Sie in der API-Referenz. [PrefetchSchedulesAWS Elemental MediaTailor](#)

 Note

Bei der Konfiguration von Prefetch-Zeitplänen ist es wichtig zu verstehen MediaTailor, wie mit verschiedenen Variablentypen umgegangen wird.

Verwenden Sie die passenden Kriterien

Wenn Sie Avail-Abgleichskriterien in einem Zeitplan verwenden möchten, stellen Sie sicher, dass Sie zuerst die ADS-URL-Vorlage Ihrer Wiedergabekonfiguration mit [dynamischen Sitzungsvariablen](#) konfigurieren, da die Auswahlkriterien sonst keine Wirkung haben. Informationen zum Arbeiten mit dynamischen Sitzungsvariablen finden Sie unter [Schritt 3: Konfigurieren Sie die ADS-Anforderungs-URL und die Abfrageparameter](#) dem Thema Erste Schritte beim Einfügen von MediaTailor Anzeigen.

Spielervariablen in Prefetch-Zeitplänen

Wenn Sie einen Prefetch-Zeitplan erstellen, definieren Sie in Ihrer Prefetch-Konfiguration keine Spielervariablen als dynamische Variablen. Übergeben Sie die Spielervariablen stattdessen so, wie Sie es normalerweise zu Beginn der Sitzung tun würden. MediaTailor schließt diese Variablen automatisch in Prefetch-Anzeigenanfragen ein, wenn die Variablen in der ADS-Vorlagen-URL zugeordnet sind.

Um einen neuen Prefetch-Zeitplan mit der Konsole zu erstellen

1. Öffnen Sie die MediaTailor Konsole unter <https://console.aws.amazon.com/mediatailor/>
2. Wählen Sie im Navigationsbereich Konfigurationen aus. Wählen Sie die Wiedergabekonfiguration aus, für die Sie einen Prefetch-Zeitplan erstellen möchten.
3. Wählen Sie auf der Registerkarte Prefetch-Zeitpläne die Option Prefetch-Zeitplan hinzufügen aus.
4. Gehen Sie im Bereich mit den Details zum Prefetch-Zeitplan wie folgt vor:
 - Geben Sie unter Name eine Kennung für Ihren Prefetch-Zeitplan ein, z. B. my-prefetch-schedule
 - Geben Sie als Stream-ID optional eine eindeutige ID ein. Wenn Ihr Origin mehrere Playback-Streams enthält, können Sie diese ID verwenden, um anzuweisen MediaTailor, Anzeigen in einem bestimmten Stream zu schalten. Wenn Ihre Wiedergabekonfiguration beispielsweise einen Sport-Stream und einen TV-Show-Stream umfasst, können Sie die Stream-ID

verwenden, um Prefetch-Zeitpläne zu erstellen, um Werbeanzeigen einzufügen, die auf den Sport-Stream zugeschnitten sind. Sie übergeben den Stream-ID-Wert an MediaTailor in der Sitzungsinitialisierungs- oder Manifestanforderung Ihres Kunden. Weitere Informationen finden Sie im folgenden Beispiel.

- Für serverseitiges Tracking fügen Sie den `?aws.streamId` Abfrageparameter und den Wert in die GET HTTP Anfrage Ihres Clients an Ihren MediaTailor Endpunkt ein. Allgemeine Informationen zur serverseitigen Nachverfolgung finden Sie unter. [MediaTailor Serverseitiges Ad-Tracking und Reporting](#) Eine Manifestanforderung an einen HLS-Endpunkt, die eine Stream-ID enthält, sieht wie folgt aus, wobei der Name Ihrer Stream-ID *myStreamId* steht:

```
GET <mediatailorURL>/v1/master/<hashed-account-id>/<origin-id>/<asset-id>?
aws.streamId=myStreamId
```

- Für die clientseitige Nachverfolgung fügen Sie den `streamId` Schlüssel und den Wert in den Text der Anfrage zur POST HTTP Sitzungsinitialisierung Ihres Clients für den Endpunkt ein. `MediaTailor/v1/session` Allgemeine Informationen zur clientseitigen Nachverfolgung finden Sie unter. [Kundenseitiges Anzeigen-Tracking](#) Eine Anfrage zur Sitzungsinitialisierung, die eine Stream-ID enthält, sieht wie folgt aus, wobei der Name Ihrer Stream-ID *myStreamId* steht:

```
POST <mediatailorURL>/v1/session/<hashed-account-id>/<origin-id>/<asset-id>
{
  'streamId': 'myStreamId',
  'reportingMode': 'client'
}
```

5. Treffen Sie für den Typ Prefetch Ihre Auswahl und wählen Sie den entsprechenden Abschnitt aus, um Hilfe bei zusätzlichen Feldern zu erhalten:

- Wählen Sie Single, wenn Sie einen Prefetch-Zeitplan für eine Werbepause in einer Veranstaltung erstellen.
- Wählen Sie Wiederkehrend, wenn Sie einen Zeitplan erstellen, bei dem Anzeigen vor jeder Werbeunterbrechung einer Veranstaltung automatisch vorab abgerufen werden.

Einzelner Prefetch-Zeitplan

Um einen Zeitplan zu erstellen, bei dem Anzeigen vorab abgerufen werden, bevor eine Anzeige in einem Ereignis verfügbar ist.

1. Geben Sie im Bereich Abruf die Abrufeinstellungen an, die Sie verwenden möchten. Diese Einstellungen bestimmen, wann Anzeigen aus dem ADS MediaTailor vorab abgerufen werden. Sie legen auch fest, welche dynamischen Sitzungsvariablen gegebenenfalls in die Anfrage an das ADS aufgenommen werden sollen.
 - Geben Sie unter Startzeit die Uhrzeit ein, zu der Prefetch-Abrufe für diese Werbeunterbrechung gestartet werden MediaTailor können. MediaTailor versucht, Anzeigen für Manifestanfragen, die Ihr Kunde an oder nach diesem Zeitpunkt gestellt hat, vorab abzurufen. Der Standardwert ist die aktuelle Zeit. Wenn Sie keinen Wert angeben, beginnt der Dienst so bald wie möglich mit dem Prefetch-Abruf.
 - Geben Sie im Feld Endzeit den Zeitpunkt ein, MediaTailor zu dem Sie das Vorabrufen von Anzeigen für diese Werbepause beenden möchten. MediaTailor versucht, Anzeigen für Manifestanfragen, die zu oder vor diesem Zeitpunkt erfolgen, vorab abzurufen. Das Abrufenster kann sich mit dem Nutzungsfenster überschneiden.
 - Optional können Sie Traffic Shaping so konfigurieren, dass die Anzahl der Anfragen an das ADS gleichzeitig begrenzt wird. Wählen Sie einen der folgenden Ansätze:

Zeitfenster-Ansatz: Geben Sie für die Dauer des Traffic Shaping-Fensters die Anzahl der Sekunden ein, über die Anfragen an das ADS verteilt werden MediaTailor sollen. Weitere Informationen finden Sie unter Erläuterung des [Zeitplans für einen einzelnen Prefetch-Abruf](#).

TPS-basierter Ansatz: Konfigurieren Sie Peak TPS und Peak Concurrent Users so, dass Anfragen auf Basis von Transaktionen pro Sekunde und gleichzeitigen Benutzern begrenzt werden. Weitere Informationen finden Sie unter [TPS-basiertes Traffic-Shaping](#).

- Geben Sie im Abschnitt [Dynamische Variablen](#) bis zu 100 dynamische Sitzungsvariablen ein. MediaTailor verwendet diese Variablen als Ersatz in Prefetch-Anfragen, die es an das ADS sendet. [Wenn Sie keine dynamischen Sitzungsvariablen eingeben, versucht es nach besten MediaTailor Kräften, die Werte für die in Ihrer ADS-URL enthaltenen dynamischen Variablen zu interpolieren.](#)
 - Wählen Sie Dynamische Variable hinzufügen aus.
 - Geben Sie als Schlüssel einen dynamischen Sitzungsvariablenschlüssel ein, z. `scte.event_id` B. Sie können jede dynamische Variable verwenden, die MediaTailor

unterstützt. Hinweise zu dynamischen Sitzungsvariablen finden Sie unter [MediaTailor Sitzungsvariablen für ADS-Anfragen](#).

- Geben Sie unter Wert einen dynamischen Variablenwert ein, z. *my-event* B.
 - Um eine weitere dynamische Variable hinzuzufügen, wählen Sie „Dynamische Variable hinzufügen“.
2. Geben Sie im Bereich Verbrauch die Einstellungen an, die Sie für das Verbrauchsfenster verwenden möchten. Diese Einstellungen bestimmen, wann MediaTailor die Anzeigen in den Werbeblock aufgenommen werden. Sie bestimmen auch alle verfügbaren Übereinstimmungskriterien, die Sie verwenden möchten.
- Geben Sie unter Startzeit die Uhrzeit ein, zu der Sie beginnen MediaTailor möchten, vorab abgerufene Anzeigen in der Werbepause zu platzieren. Der Standardwert ist die aktuelle Uhrzeit. Wenn Sie keine Uhrzeit angeben, beginnt der Dienst so bald wie möglich mit dem Prefetch-Verbrauch.
 - Geben Sie unter Endzeit einen Zeitpunkt ein, MediaTailor zu dem Sie die Platzierung der vorab abgerufenen Anzeigen in der Werbepause beenden möchten. MediaTailor versucht, Anzeigen für die Manifestanfragen Ihres Kunden, die zu oder vor diesem Zeitpunkt eingehen, vorab abzurufen. Die Endzeit muss nach der Startzeit und in weniger als einem Tag liegen. Das Verbrauchsfenster kann sich mit dem Abruffenster überschneiden.
 - Wählen [Sie im Abschnitt Verfügbare Kriterien](#) die Option Verfügbarkeitskriterien hinzufügen aus und fügen Sie Ihrem Zeitplan bis zu fünf passende Verfügbarkeitskriterien hinzu. Fügen Sie dann unter Dynamischer Variablenschlüssel einen dynamischen Variablenschlüssel hinzu, z. B. `scte.event_id` MediaTailor platziert die vorab abgerufenen Anzeigen nur dann im Werbeblock, wenn sie die Kriterien erfüllen, die durch die dynamischen Variablenwerte definiert sind MediaTailor, an die Ihr Kunde entweder weiterleitet oder die MediaTailor aus Informationen wie Sitzungsdaten abgeleitet werden. Wenn eine Werbeunterbrechung die angegebenen Übereinstimmungskriterien nicht erfüllt, wird der Vorabruf für diese Werbepause MediaTailor übersprungen. Weitere Informationen finden Sie in der Erläuterung zum Verbrauch [eines einzelnen Prefetch-Zeitplans](#).
3. Wählen Sie „Verfügbarkeitskriterien hinzufügen“ aus.

Prefetch-Zeitpläne laufen automatisch ab, wenn das Nutzungsfenster abgelaufen ist. Zu Diagnosezwecken bleiben sie mindestens 7 Tage lang sichtbar und MediaTailor werden danach automatisch gelöscht. Alternativ können Sie einen Prefetch-Zeitplan jederzeit manuell löschen.

Informationen zum manuellen Löschen eines Prefetch-Zeitplans finden Sie im folgenden Abschnitt. [the section called “Prefetch-Zeitpläne werden gelöscht”](#)

Bestimmen Sie, wie oft Ihr Client die API aufrufen soll `CreatePrefetchSchedule`

Ihr Kunde kann die [CreatePrefetchSchedule](#) API einmal täglich programmgesteuert aufrufen, um den Abruf und die Nutzung einzurichten, sofern Sie genau wissen, wann Werbeunterbrechungen auftreten werden. Oder Ihr Kunde kann die API im Laufe des Tages viele Male aufrufen, um den Abruf und die Nutzung zu definieren. Berücksichtigen Sie bei der Auswahl der API-Aufrufhäufigkeit die [maximale Anzahl aktiver Prefetch-Zeitpläne](#) und die Wahrscheinlichkeit, dass sich Ihr Zeitplan für Werbeunterbrechungen ändert, nachdem Sie Ihre Prefetch-Zeitpläne erstellt haben. Wenn es wahrscheinlich ist, dass sich der Zeitplan für Werbeunterbrechungen ändert, nachdem Sie Ihre Prefetch-Zeitpläne erstellt haben, sollten Sie die API möglicherweise häufiger aufrufen.

Wiederkehrender Prefetch-Zeitplan

Um einen Zeitplan zu erstellen, bei dem Anzeigen vorab abgerufen werden, bevor jede Anzeige in einem Event verfügbar ist.

1. Geben Sie im Bereich Wiederkehrender Abruf die Abrufeinstellungen an, die Sie verwenden möchten. Diese Einstellungen bestimmen, wann Anzeigen aus dem ADS MediaTailor vorab abgerufen werden. Sie legen auch fest, welche dynamischen Sitzungsvariablen gegebenenfalls in die Anfrage an das ADS aufgenommen werden sollen.
 - Geben Sie unter Wiederkehrendes Prefetch-Fenster die Uhrzeit ein, zu der Prefetch-Abrufe für diese Werbepause gestartet werden MediaTailor können. MediaTailor versucht, Anzeigen für Manifestanfragen, die Ihr Kunde an oder nach diesem Zeitpunkt gestellt hat, vorab abzurufen. Der Standardwert ist die aktuelle Zeit. Wenn Sie keinen Wert angeben, beginnt der Dienst so bald wie möglich mit dem Prefetch-Abruf.
 - Geben Sie unter Verzögerung nach Ende der Verfügbarkeit die Anzahl der Sekunden ein, die nach Ablauf einer Verfügbarkeit warten MediaTailor sollen, bevor Anzeigen für die nächste Verfügbarkeit vorab abgerufen werden. Wenn Sie keinen Wert angeben, wird MediaTailor standardmäßig „Keine Verzögerung“ verwendet.
 - Optional können Sie Traffic Shaping so konfigurieren, dass die Anzahl der Anfragen an das ADS gleichzeitig begrenzt wird. Wählen Sie einen der folgenden Ansätze:

Zeitfenster-Ansatz: Geben Sie für die Dauer des Traffic Shaping-Fensters die Anzahl der Sekunden ein, über die Anfragen an das ADS verteilt werden MediaTailor sollen. Weitere

Informationen finden Sie unter Erläuterung des Zeitplans für den [wiederkehrenden Prefetch-Abruf](#)

TPS-basierter Ansatz: Konfigurieren Sie Peak TPS und Peak Concurrent Users so, dass Anfragen auf Basis von Transaktionen pro Sekunde und gleichzeitigen Benutzern begrenzt werden. Weitere Informationen finden Sie unter [TPS-basiertes Traffic-Shaping](#).

- Geben Sie im Abschnitt [Dynamische Variablen](#) bis zu 100 dynamische Sitzungsvariablen ein. MediaTailor verwendet diese Variablen als Ersatz in Prefetch-Anfragen, die es an das ADS sendet. [Wenn Sie keine dynamischen Sitzungsvariablen eingeben, versucht es nach besten MediaTailor Kräften, die Werte für die in Ihrer ADS-URL enthaltenen dynamischen Variablen zu interpolieren.](#)
 - Wählen Sie Dynamische Variable hinzufügen aus.
 - Geben Sie als Schlüssel einen dynamischen Sitzungsvariablenschlüssel ein, z. `scte.event_id` B. Sie können jede dynamische Variable verwenden, die MediaTailor unterstützt. Hinweise zu dynamischen Sitzungsvariablen finden Sie unter [MediaTailor Sitzungsvariablen für ADS-Anfragen](#).
 - Geben Sie unter Wert einen dynamischen Variablenwert ein, z. *my-event* B.
 - Um eine weitere dynamische Variable hinzuzufügen, wählen Sie „Dynamische Variable hinzufügen“.
2. Geben Sie im Bereich Verbrauch die Einstellungen an, die Sie für das Verbrauchsfenster verwenden möchten. Diese Einstellungen bestimmen, wann MediaTailor die Anzeigen in den Werbeblock aufgenommen werden. Sie bestimmen auch alle verfügbaren Übereinstimmungskriterien, die Sie verwenden möchten.
- Geben Sie unter Ablauf der abgerufenen Anzeige an, wie lange nach dem Abruf Anzeigen zum Einfügen verfügbar sind.
 - Wählen Sie im Abschnitt [Verfügbare Übereinstimmungskriterien](#) die Option Verfügbarkeitskriterien hinzufügen aus und fügen Sie Ihrem Zeitplan bis zu fünf verfügbare Kriterien hinzu. Fügen Sie dann unter Dynamischer Variablenschlüssel einen dynamischen Variablenschlüssel hinzu, z. B. `scte.event_id` MediaTailor platziert die vorab abgerufenen Anzeigen nur dann im Werbeblock, wenn sie die Kriterien erfüllen, die durch die dynamischen Variablenwerte definiert sind MediaTailor, an die Ihr Kunde entweder weiterleitet oder die MediaTailor aus Informationen wie Sitzungsdaten abgeleitet werden. Wenn eine Werbeunterbrechung die angegebenen Übereinstimmungskriterien nicht erfüllt, wird der Vorabruf für diese Werbepause MediaTailor übersprungen. Weitere Informationen finden Sie in der Erläuterung der Nutzung des Zeitplans für [wiederkehrende Prefetches](#).

3. Wählen Sie „Verfügbarkeitskriterien hinzufügen“ aus.

Prefetch-Zeitpläne laufen automatisch ab, wenn das Nutzungsfenster abgelaufen ist. Zu Diagnosezwecken bleiben sie mindestens 7 Tage lang sichtbar und MediaTailor werden danach automatisch gelöscht. Alternativ können Sie einen Prefetch-Zeitplan jederzeit manuell löschen. Informationen zum manuellen Löschen eines Prefetch-Zeitplans finden Sie im folgenden Abschnitt. [the section called “Prefetch-Zeitpläne werden gelöscht”](#)

TPS-basiertes Traffic-Shaping

AWS Elemental MediaTailor bietet zwei optionale Traffic-Shaping-Ansätze, um die Anzahl der Anfragen an das ADS gleichzeitig zu begrenzen. TPS-basiertes Traffic Shaping bietet eine Alternative zu Traffic Shaping auf Zeitfensterbasis für Prefetch-Zeitpläne. Dieser Ansatz bietet eine intuitivere Konfiguration, da Sie die Kapazität Ihres AD Decision Server (ADS) in Form von Transaktionen pro Sekunde (TPS) und erwarteten gleichzeitigen Benutzern angeben können, anstatt Zeitberechnungen vorzunehmen.

So funktioniert TPS-basiertes Traffic Shaping

Anstatt die Dauer des Abruffensters anzugeben, geben Sie die folgenden Parameter an:

Spitzenwert TPS

Die maximale Anzahl von Anfragen pro Sekunde, die Ihr ADS verarbeiten kann. Dieser Parameter hat keinen Standardwert.

Höchstzahl gleichzeitiger Benutzer

Die erwartete Spitzenzahl gleichzeitiger Zuschauer für Ihre Inhalte. Dieser Parameter hat keinen Standardwert.

MediaTailor verteilt Prefetch-Anfragen automatisch über einen bestimmten Zeitraum, sodass das angegebene TPS-Limit eingehalten wird, unabhängig von der Anzahl der gleichzeitigen Sitzungen.

Example Beispiel für eine TPS-basierte Konfiguration

Ihr ADS kann 500 TPS verarbeiten, und Sie erwarten zu Spitzenzeiten 100.000 gleichzeitige Zuschauer. Sie konfigurieren:

- Spitzen-TPS: 500

- Maximale Anzahl gleichzeitiger Benutzer: 100.000

MediaTailor verteilt Prefetch-Anfragen automatisch über einen bestimmten Zeitraum, sodass das angegebene TPS-Limit eingehalten wird, unabhängig von der Anzahl der gleichzeitigen Sitzungen.

Prefetch-Zeitpläne werden gelöscht

Im folgenden Verfahren wird erklärt, wie Sie einen Prefetch-Zeitplan mithilfe der Konsole löschen. MediaTailor Informationen zum programmgesteuerten Löschen von Prefetch-Zeitplänen mithilfe der MediaTailor API finden Sie in der API-Referenz. [DeletePrefetchSchedule](#) AWS Elemental MediaTailor

Note

Das Löschen erfolgt nicht in Echtzeit. Beim MediaTailor Löschen der Prefetch-Zeitpläne kann es zu Verzögerungen kommen. Während dieser Zeit laufen der Prefetch-Abruf und die Nutzung weiterhin im Hintergrund.

Um einen Prefetch-Zeitplan mit der Konsole zu löschen

1. Öffnen Sie die MediaTailor Konsole unter. <https://console.aws.amazon.com/mediatailor/>
2. Wählen Sie im Navigationsbereich Konfigurationen aus. Wählen Sie die Wiedergabekonfiguration aus, die die Prefetch-Zeitpläne enthält, die Sie löschen möchten.
3. Wählen Sie auf der Registerkarte Prefetch-Zeitpläne den Prefetch-Zeitplan aus, den Sie löschen möchten. Wählen Sie dann Löschen aus.

Verwenden von vorkonditionierten Anzeigen mit AWS Elemental MediaTailor

In einem [typischen Arbeitsablauf zum Einfügen von Anzeigen](#) werden Anzeigen MediaTailor dynamisch transkodiert, sodass sie dem Inhaltsstream entsprechen, gespeichert und in den Livestream integriert. Da dieser Vorgang erst stattfindet, nachdem die Anzeige in einer VAST-Antwort vom Ad Decision Server (ADS) MediaTailor empfangen wurde, kommt es zu einer Verzögerung, bis die Anzeige zum Zusammenfügen verfügbar ist. Wenn beim Ad Stitching-Workflow zusätzliche Latenz entsteht (entweder aufgrund eines ADS-Timeouts oder anderer Inhalts- oder

Netzwerkprobleme), MediaTailor kann dies dazu führen, dass die Verfügbarkeit teilweise aufgefüllt wird oder die Werbeunterbrechung ganz verpasst wird.

Um den Zeitaufwand für das Einfügen von Anzeigen in Ihre Inhalte zu reduzieren, können Sie vorkonditionierte Anzeigen verwenden. Eine vorkonditionierte Anzeige ist eine Anzeige, die Sie transkodieren, bevor Sie sie bei der Anzeigeneinfügung verwenden. MediaTailor Anstatt die unkonditionierten Anzeigen URLs für Ihre ADS bereitzustellen, stellen Sie die URLs für die vorkonditionierten Anzeigen bereit. In seiner VAST-Antwort auf die MediaTailor Anfrage enthält das ADS direkte Links zu den vorkonditionierten Anzeigen. Durch das Entfernen des transkodierenden Teils des Anzeigen-Stitchings MediaTailor muss die Anzeige lediglich gespeichert und in den Inhaltsstream eingefügt werden. Beim Anzeigen-Stitching-Verfahren mit vorkonditionierten Anzeigen wird die Zeit zwischen dem Zeitpunkt, an dem die Anzeige durch die VAST-Antwort auf eine Anzeige aufmerksam gemacht MediaTailor wird, und dem Zeitpunkt, an dem die Anzeige in den Inhalt eingebunden wird, verkürzt.

Alternativ können Sie auch das Ad Prefetching verwenden. In diesem Fall konfigurieren Sie, dass der Ad Stitching-Vorgang MediaTailor zu einem festgelegten Zeitpunkt ausgeführt wird, bevor die Werbepause erforderlich ist. Weitere Informationen zum Vorabruf von Anzeigen finden Sie unter.

[Anzeigen vorab abrufen](#)

Anforderungen an vorkonditionierte Anzeigen

Die folgenden Anforderungen müssen bei der Einrichtung eines Workflows zur Anzeigenzusammenstellung mit vorkonditionierten Anzeigen berücksichtigt werden.

MediaFiles-Voraussetzungen

Die VAST-Antwort, an die der Ad-Server sendet, MediaTailor muss enthalten MediaFiles, dass diese Anforderungen erfüllt sind:

Die Anzeige (Creative) muss Varianten enthalten, die den Bitratenvarianten des Inhaltsstreams entsprechen. Es liegt in Ihrer Verantwortung, sicherzustellen, dass die VAST-Antwort die richtigen Anzeigenvarianten verwendet, die mit den Vorlagenmanifesten übereinstimmen.

Die Verwendung von vorkonditionierten Anzeigen kann zwar dazu beitragen, die Anzeigeneinblendung effizienter zu gestalten, ist MediaTailor jedoch nicht in der Lage, den Transcodierungsprozess zu verwalten, um sicherzustellen, dass die Mediendateien für die Anzeigen mit den Spezifikationen der Inhaltsmanifeste kompatibel sind. Wenn die Anzeige nicht mit dem Content-Stream übereinstimmt, MediaTailor könnte die Einblendung verpasst werden, oder die Nichtübereinstimmung könnte zu einem Fehler auf dem Wiedergabegerät führen.

Um ohne MediaTailor Transcodierung in den Inhaltsstream eingebunden zu werden, MediaFile muss eine außerdem die folgenden Anforderungen erfüllen:

- Es muss im öffentlichen Internet zugänglich sein, damit es heruntergeladen MediaTailor werden kann.
- Es muss die Streaming-Bereitstellung verwenden, wie `delivery="streaming"` in der VAST-Antwort angegeben.
- Es muss sich entweder um eine `.m3u8` (für HLS) oder `.mpd` (für DASH) Datei handeln.

Example RIESIGE Antwort

Aus der folgenden VAST-Beispielantwort MediaTailor MediaFile fügt Folgendes ein URLs:

- Für einen HLS-Stream MediaTailor verwendet `https://example-ad-origin.amazonaws.com/ad1/index_low.m3u8`. Dies ist die erste Version MediaFile mit Streaming-Bereitstellung und unterstützter Dateierweiterung (`.m3u8`).
- Für einen DASH-Stream MediaTailor verwendet `https://example-ad-origin.amazonaws.com/ad1/index.mpd`. Dies ist der erste MediaFile mit Streaming-Bereitstellung und einer unterstützten Dateierweiterung (`.mpd`).

```
<?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-f7e566ffc605</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.m3u8]]></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>

```

Und offensichtliche Anforderungen

Um vorkonditionierte Anzeigen verwenden zu können, müssen die Anzeigenlisten für Eltern und Kinder die folgenden Anforderungen erfüllen:

- Bei dem Manifest, auf das im Creative Abschnitt der VAST-Antwort verwiesen wird, muss es sich um das übergeordnete Anzeigenmanifest handeln.
- URLs Bei den Anzeigenmanifesten für das untergeordnete Objekt muss es sich um relative Pfade handeln.
- Die untergeordneten Anzeigenmanifeste müssen sich im selben Verzeichnis wie die übergeordnete multivariante Playlist auf derselben Ebene befinden. Untergeordnete Manifeste dürfen sich nicht in einem Unterverzeichnis oder an einem anderen Ort befinden.

Example unterstützte übergeordnete multivariante Playlist

Die folgende multivariante Playlist für übergeordnete und untergeordnete Playlisten enthält Playlisten URLs für Mediendateien. Die untergeordneten Playlisten befinden sich ebenfalls im selben Verzeichnis wie die übergeordnete multivariante Playlist.

```

#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 Die übergeordnete multivariante Playlist wird nicht unterstützt: Unterverzeichnisse

Die folgende übergeordnete und multivariante Playlist enthält untergeordnete Playlisten, die sich relativ zur übergeordneten multivarianten Playlist in Unterverzeichnissen befinden. Es handelt sich nicht um eine unterstützte Playlist für vorkonditionierte Werbung.

```
#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 Multivariante übergeordnete Playlist wird nicht unterstützt: absolut URLs

Die folgende übergeordnete und multivariante Playlist enthält untergeordnete Playlisten mit absoluten Werten. URLs Es handelt sich nicht um eine unterstützte Playlist für vorkonditionierte Werbung.

```
#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
```

Arbeitsablauf für vorkonditionierte Anzeigen

Im Folgenden finden Sie eine grundlegende Beschreibung der Funktionsweise vorkonditionierter Anzeigen in einem Ad Stitching-Workflow mit MediaTailor. Der erste Teil des Workflows besteht aus Maßnahmen, die Sie ergreifen müssen, um die Verwendung vorkonditionierter Anzeigen einzurichten. Im zweiten Teil wird beschrieben, wie die Anzeigen MediaTailor verarbeitet werden.

Teil 1: Einrichtung von vorkonditionierten Anzeigen

Gehen Sie wie folgt vor, um einen Workflow einzurichten, der vorkonditionierte Anzeigen in verwendet. MediaTailor

1. Verwenden Sie einen Transcoder-Service, z. B. um Ihre Werbemittel in Varianten zu unterteilen AWS Elemental MediaConvert, die die verschiedenen Bitraten, Auflösungen und Codecs Ihrer Vorlagenmanifeste unterstützen.
2. Stellen Sie Ihrem ADS die URLs für die vortranskodierte Mediendateien zur Verfügung, damit sie in VAST-Antworten verwendet werden können.
3. [Erstellen Sie Ihre Wiedergabekonfiguration](#) in. MediaTailor Um vorkonditionierte Werbung zu verwenden, wählen Sie in der Konfiguration für die Einstellung Konditionierung von Streaming-Mediendateien die Option Keine aus.
4. Fahren Sie mit der Einrichtung Ihrer Inhaltsbereitstellung wie gewohnt fort.

Teil 2: MediaTailor Anzeigenverarbeitung

MediaTailor Das Zusammenfügen von Anzeigen wird wie unter beschrieben abgeschlossen.

[Wie funktioniert das Einfügen von MediaTailor Anzeigen](#) Wenn die ADS eine VAST-Antwort MediaTailor erhält, bestimmt sie anhand der folgenden Logik, welche Maßnahmen für die Anzeigen ergriffen werden sollen. Diese Logik wird durch die Einstellung zur Konditionierung von Streaming-Mediendateien in der Wiedergabekonfiguration vorgegeben.

- Wenn die Konditionierung von Streaming-Mediendateien auf Transkodieren eingestellt ist, werden die Mediendateien bei der progressive Auslieferung MediaTailor transkodiert und in das Manifest zusammengefügt. Wenn nicht genügend Mediendateien für Werbeanzeigen mit progressive Versand zur Verfügung stehen, werden diese Dateien MediaTailor transkodiert und bei Lieferung verwendet. streaming
- Wenn die Konditionierung von Streaming-Mediendateien auf „Keine“ gesetzt ist, werden Anzeigen mit MediaTailor streaming Übertragungsmediendateien in das Manifest zusammengefügt, ohne sie zu transkodieren. Wenn nicht genügend Werbeanzeigen mit Mediendateien zur streaming Auslieferung vorhanden sind, werden diese Dateien MediaTailor transkodiert und bei der Auslieferung verwendet. progressive

MediaTailor dynamische Anzeigenvariablen für ADS-Anfragen

AWS Elemental MediaTailor verwendet dynamische Anzeigenvariablen, um Informationen aus Ihrer Anzeigesitzung an den Ad Decision Server (ADS) weiterzuleiten. Diese Informationen helfen ADS dabei, die relevantesten Anzeigen für Ihre Zuschauer auszuwählen.

Dieser Abschnitt bietet einen Überblick über dynamische Anzeigenvariablen und Links zu spezifischen Implementierungsleitfäden. Anweisungen zur step-by-step Konfiguration finden Sie in den einzelnen Themen unten.

Dynamische Variablentypen

MediaTailor unterstützt vier Typen dynamischer Variablen:

- Sitzungsvariablen — Automatisch generierte Werte wie Sitzungs-ID und SCTE-35-Daten. Siehe [MediaTailor Sitzungsvariablen für ADS-Anfragen](#).
- Player-Variablen — Benutzerdefinierte Parameter, die von Ihrem Videoplayer gesendet werden. Siehe [MediaTailor Player-Variablen für ADS-Anfragen](#).
- Domain-Variablen mit Konfigurations-Aliasen — Dynamische URL-Domains für Konfigurationen mit mehreren Ursprüngen.
- Konfigurationsalias — Vordefinierte Zuordnungen für den dynamischen Variablenersatz. Siehe [Konfigurations-Aliase](#).

Häufige Anwendungsfälle

Verwenden Sie dynamische Anzeigenvariablen, um:

- Übergeben Sie demografische Daten und Präferenzen der Zuschauer an Ihre ADS
- Leitet Anfragen je nach geografischem Standort an unterschiedliche Ursprünge weiter
- Ermöglichen Sie mit MediaPackage der Integration eine zeitversetzte Anzeige
- Implementieren Sie A/B Test- und Failover-Szenarien

In den folgenden Abschnitten finden Sie weitere Informationen zur Verwendung dynamischer Anzeigenvariablen mit MediaTailor.

Themen

- [MediaTailor Sitzungsvariablen für ADS-Anfragen](#)

- [MediaTailor Player-Variablen für ADS-Anfragen](#)
- [MediaTailor Domain-Variablen für mehrere Inhaltsquellen](#)
- [MediaTailor Übersicht über Konfigurationsaliasse](#)
- [MediaTailor Parameter an ADS übergeben](#)
- [MediaTailor Parameter-Routing für ADS und Origins](#)
- [MediaTailor und MediaPackage zeitversetzte Anzeigeintegration](#)
- [MediaTailor Verhalten und Persistenz von Parametersitzungen](#)
- [MediaTailor Parameterreferenz und Einschränkungen](#)
- [MediaTailor Leitfaden zur Fehlerbehebung bei Parametern](#)
- [MediaTailor Leitfaden zur Fehlerbehebung bei Konfigurationsaliasen](#)

Informationen zu den Anforderungen an die Parameterformatierung und zur Problembehandlung finden Sie unter [MediaTailor Parameterreferenz und Einschränkungen](#) und [MediaTailor Leitfaden zur Fehlerbehebung bei Parametern](#).

MediaTailor Sitzungsvariablen für ADS-Anfragen

AWS Elemental MediaTailor sendet Sitzungsdaten an den Ad Decision Server (ADS), wenn Sie die Konfiguration so konfigurieren AWS Elemental MediaTailor, dass eine oder mehrere der in diesem Abschnitt aufgeführten Variablen in der ADS-URL der Vorlage angegeben werden. Sie können einzelne Variablen verwenden und mehrere Variablen zu einem einzigen Wert verketteten. MediaTailor generiert einige Werte und bezieht den Rest aus Quellen wie dem Manifest und der Anfrage des Players zur Initialisierung der Sitzung.

In der folgenden Tabelle werden die Sitzungsdatenvariablen beschrieben, die Sie in der URL-Konfiguration Ihrer Vorlage für ADS-Anfragen verwenden können. Die in der Tabelle aufgeführten Abschnittsnummern entsprechen der Version 2019a der Spezifikation -35 der Society of Cable Telecommunications Engineers (SCTE), [Digital Program Insertion Cueing Message](#). Weitere Informationen zu Ad Prefetch finden Sie unter [Anzeigen vorab abrufen](#)

Name	Verfügbar für Ad-Prefetch	Abschnitt mit den SCTE-35-Spezifikationen	Description
[avail.index]	Ja		Eine Zahl, die die Position einer verfügbaren Anzeige in einem Index angibt. MediaTailor Erstellt zu Beginn einer Wiedergabesitzung einen Index aller verfügbaren Werbeanzeigen in einem Manifest und speichert den Index für den Rest der Sitzung. Wenn eine MediaTailor-Anfrage an das ADS gestellt wird, um die Verfügbarkeit aufzufüllen, enthält diese die Indexnummer der Anzeigenverfügbarkeit. Dieser Parameter ermöglicht es dem ADS, die Anzeigenauswahl unter Verwendung von Funktionen wie Wettbewerbsausschluss und Frequenzumgrenzung zu verbessern.
[avail.random]	Ja		Eine Zufallszahl zwischen 0 und 10.000.000.000 als lange Zahl, die für jede Anfrage an das ADS MediaTailor generiert wird. Einige Werbe-Server verwenden diesen Parameter, um Funktionen wie z. B. die Trennung der Werbung von Konkurrenzunternehmen zu ermöglichen.
[scte.archive_allowed_flag]	Ja	10.3.3.1	Ein optionaler boolescher Wert. Wenn dieser Wert 0 ist, gelten für das Segment Aufzeichnungseinschränkungen. Wenn dieser Wert 1 ist, gelten für das Segment keine Aufnahmeeinschränkungen.
[scte.available_num]	Ja	9.7.2.1	Der Wert, der MediaTailor aus dem SCTE-35-Feld analysiert wurde <code>avail_num</code> , als lange Zahl. MediaTailor kann diesen Wert verwenden, um lineare und verfügbare Zahlen zu bezeichnen. Der Wert muss eine Ganzzahl sein.

Name	Verfügbar für Ad-Prefetch	Abschnitt mit den SCTE-35-Spezifikationen	Description
[scte.ava ils_expected]	Ja	9,7.2.1	Ein optionaler langer Wert, der die erwartete Anzahl von Verfügungen innerhalb des aktuellen Ereignisses angibt.
[scte.del ivery_not _restrict ed_flag]	Ja	10.3.3.1	Ein optionaler boolescher Wert. Wenn dieser Wert 0 ist, sind die nächsten fünf Bits reserviert. Wenn dieser Wert 1 ist, nehmen die nächsten fünf Bits die in der SCTE-35-Spezifikation beschriebenen Bedeutungen an.
[scte.dev ice_restr ictions]	Ja	10.3.3.1	Ein optionaler Ganzzahlwert, der drei vordefinierte, unabhängige und nicht hierarchische Gerätegruppen signalisiert. Weitere Informationen zu dieser Variablen finden Sie in der Beschreibung segments_expected in der SCTE-35-Spezifikation.
[scte.eve nt_id]	Ja	9.1 und 9.7.2.1	Der Wert, nach dem MediaTailor aus dem SCTE-35-Feld geparsed wurde <code>splICE_event_id</code> , als lange Zahl. MediaTailor verwendet diesen Wert, um lineare Anzeigenverfügbarkeitsnummern zu kennzeichnen oder um Anzeigenserver-Abfragezeichenfolgen, wie z. B. die Positionen von Ad-Pods, aufzufüllen. Der Wert muss eine Ganzzahl sein.
[scte.no_ regional_ blackout_ flag]	Ja	10.3.3.1	Ein optionaler boolescher Wert. Wenn dieser Wert 0 ist, gelten für das Segment regionale Blackout-Einschränkungen. Wenn dieser Wert 1 ist, gelten die regionalen Blackout-Einschränkungen nicht für das Segment.

Name	Verfügbar für Ad-Prefetch	Abschnitt mit den SCTE-35-Spezifikationen	Description
[scte.segment_num]	Ja	10.3.3.1	Ein optionaler Integer-Wert, der Segmente innerhalb einer Sammlung von Segmenten nummeriert. Weitere Informationen zu dieser Variablen finden Sie in der Beschreibung segment_num in der SCTE-35-Spezifikation.
[scte.segmentation_event_id]	Ja	10.3.3.1	MediaTailor macht diese Variable verfügbar als scte.event_id
[scte.segmentation_type_id]	Ja	10.3.3.1	Ein optionaler 8-Bit-Ganzzahlwert, der den Segmentierungstyp angibt. Weitere Informationen zu dieser Variablen finden Sie in der Beschreibung segmentation_type_id in der SCTE-35-Spezifikation.

Name	Verfügbar für Ad-Prefetch	Abschnitt mit den SCTE-35-Spezifikationen	Description
[scte.segmentation_upid]	segmentation_upid_type : Yes private_data : Yes	segmentation_upid: 10.3.3.1 Verwaltet eine private UPID: 10.3.3.3	Entspricht dem SCTE-35-Element. <code>segmentation_upid</code> Das <code>segmentation_upid</code> Element enthält <code>segmentation_upid_type</code> und <code>segmentation_upid_length</code>. MediaTailor unterstützt die folgenden <code>segmentation_upid</code> Typen: • ADS-Informationen (0x0E) — Werbeinformationen . Weitere Informationen finden Sie in der Beschreibung <code>segmentation_upid</code> in der SCTE-35-Spezifikation. • Managed Private UPID (0x0C) — Die Managed Private UPID (MPU) -Struktur, wie in der SCTE-35-Spezifikation definiert. MediaTailor unterstützt binäre oder DASH-XML-SCTE-Repräsentationen. Sie können diese Struktur in einem Podbuster-Workflow verwenden. Geben Sie dazu einen 32-Bit-Wert (4 Byte) an und <code>format_identifizieren</code> Sie die folgenden Parameter in das <code>private_data</code> Attribut auf: <pre>ABCD{"assetId": " my_program ", "cueData": {"cueType": " theAdType ", "key": " pb", "value": " 123456"}}</pre> MediaTailor analysiert die Werte aus dem vorherigen JSON und übergibt sie an die <code>scte.segmentation_upid.cueData.value</code> dynamischen Variablen <code>scte.segmentation_</code>

Name	Verfügbar für Ad-Prefetch	Abschnitt mit den SCTE-35-Spezifikationen	Description
			<code>upid.assetId</code> <code>scte.segmentation_</code> <code>upid.cueData.key</code> , und. Benutzerdefiniert (0x01) — Eine benutzerdefinierte Struktur. Weitere Informationen finden Sie in der Beschreibung <code>segmentation_upid</code> in der SCTE-35-Spezifikation.
[<code>scte.segmentation_upid.assetId</code>]	Ja		Wird in Verbindung mit der Managed Private UPID (0xC) für Podbuster-Workflows verwendet. <code>segmentation_upid_type</code> MediaTailor leitet diesen Wert aus dem <code>assetId</code> Parameter in der JSON-Struktur der MPU ab. <code>private_data</code> Weitere Informationen finden Sie unter Managed Private UPID JSON structure for a podbuster workflow.
[<code>scte.segmentation_upid.cueData.key</code>]	Ja		Wird in Verbindung mit der Managed Private UPID (0xC) <code>segmentation_upid_type</code> für Podbuster-Workflows verwendet. MediaTailor leitet diesen Wert aus dem <code>cueData.key</code> Parameter in der JSON-Struktur der MPU ab. <code>private_data</code> Weitere Informationen finden Sie unter Managed Private UPID JSON structure for a podbuster workflow.

Name	Verfügbar für Ad-Prefetch	Abschnitt mit den SCTE-35-Spezifikationen	Description
[scte.segmentation_upid.cueData.value]	Ja		Wird in Verbindung mit der Managed Private UPID (0xC) <code>segmentation_upid_type</code> für Podbuster-Workflows verwendet. MediaTailor leitet diesen Wert aus dem <code>cueData.key</code> Parameter in der JSON-Struktur der MPU ab. <code>private_data</code> Weitere Informationen finden Sie unter Managed Private UPID JSON structure for a podbuster workflow. Der Wert kann eine Zeichenfolge sein.
[scte.segmentation_upid.private_data.{index}]	Ja		Wird in Verbindung mit der Managed Private UPID (0xC) <code>segmentation_upid_type</code> für gezielte Werbe-Workflows verwendet. MediaTailor teilt durch Doppelpunkte getrennte UPID-Token für die Segmentierung auf und erstellt indizierte Sitzungsvariablen. Der Index entspricht der Position in der durch Doppelpunkte getrennten Liste, wobei führende Leerzeichen vor dem ersten Doppelpunkt ignoriert werden. Zum Beispiel, wenn, dann: <code>segmentation_upid = ":3213214:2313321/5:3943"</code> <code>[scte.segmentation_upid.private_data.0]</code> = 3213214 <code>[scte.segmentation_upid.private_data.1]</code> = 2313321/5 <code>[scte.segmentation_upid.private_data.2]</code> = 3943 Der Wert kann eine Zeichenfolge sein.

Name	Verfügbar für Ad-Prefetch	Abschnitt mit den SCTE-35-Spezifikationen	Description
[scte.segments_expected]	Ja	10.3.3.1	Ein optionaler Integer-Wert, der die erwartete Anzahl einzelner Segmente innerhalb einer Sammlung von Segmenten angibt. Weitere Informationen zu dieser Variablen finden Sie in der Beschreibung segments_expected in der SCTE-35-Spezifikation.
[scte.sub_segment_num]	Ja	10.3.3.1	Ein optionaler Ganzzahlwert, der ein bestimmtes Untersegment innerhalb einer Sammlung von Untersegmenten identifiziert. Weitere Informationen zu dieser Variablen finden Sie in der Beschreibung sub_segment_num in der SCTE-35-Spezifikation.
[scte.sub_segments_expected]	Ja	10.3.3.1	Ein optionaler Ganzzahlwert, der die erwartete Anzahl einzelner Untersegmente innerhalb einer Sammlung von Untersegmenten angibt. Weitere Informationen zu dieser Variablen finden Sie in der Beschreibung sub_segments_expected in der SCTE-35-Spezifikation.
[scte.unique_program_id]	Ja	9.7.2.1	Der Integer-Wert, der MediaTailor aus dem SCTE-35-Feld analysiert wurde. splice_insert unique_program_id Der ADS stellt anhand der eindeutigen Programm-ID (UPID) zielgerichtete Werbung auf Programmebene für lineare Live-Streams bereit. Wenn der SCTE-35-Befehl nicht auf Spleißeinfügung gesetzt ist, wird dieser Wert auf einen leeren Wert MediaTailor gesetzt. Der Wert muss eine Ganzzahl sein.

Name	Verfügbar für Ad-Prefetch	Abschnitt mit den SCTE-35-Spezifikationen	Description
[session.avail_duration_ms]	Ja		Die Dauer des Zeitfensters für die Anzeigenverfügbarkeit in Millisekunden. Der Standardwert ist 300.000 ms. AWS Elemental MediaTailor erhält den Wert für die Dauer wie folgt aus dem Eingabemanifest: - Für HLS: MediaTailor Ruft die Dauer aus den Werten #EXT-X-CUE-OUT: DURATION oder aus den Werten im Tag ab. #EXT-X-DATERANGE. Wenn das Eingabemanifest eine Dauer von Null, ungültig oder 0 für die Anzeigenverfügbarkeit in diesen Tags hat, wird der Standardwert MediaTailor verwendet. - Für DASH: MediaTailor Ruft den Wert für die Dauer aus der Eventdauer ab, sofern eine angegeben wurde. Andernfalls wird der Standardwert verwendet. - Für VOD: Wenn ein VOD-Stream einen Pre-Roll-Werbeaufruf auslöst und das Manifest keine SCTE-Nachrichten mit einem Dauerwert enthält, wird MediaTailor keine Dauer für [session.avail_duration_ms] eingegeben, einschließlich des Standardwerts für die Dauer.
[session.avail_duration_sec]	Ja		Die auf die nächste Sekunde gerundete Dauer des Anzeigenverfügbarkeitsslots oder der Anzeigenverfügbarkeit in Sekunden. MediaTailor bestimmt diesen Wert auf die gleiche Weise, wie er ihn bestimmt [session.avail_duration_ms].

Name	Verfügbar für Ad-Prefetch	Abschnitt mit den SCTE-35-Spezifikationen	Description
[session.client_ip]	Nein		Die Remote-IP-Adresse, von der die MediaTailor Anfrage kam. Wenn der X-forwarded-for -Header festgelegt ist, dann verwendet MediaTailor diesen Wert für die client_ip .
[session.id]	Nein		Eine eindeutige numerische Kennung für die aktuelle Wiedergabesitzung. Alle Anforderungen eines Players für eine Sitzung haben dieselbe ID, sodass sie für ADS-Felder verwendet werden kann, die zum Korrelieren aller Anforderungen in einer einzelnen Ansicht bestimmt sind.
[session.referer]	Nein		In der Regel die URL der Seite, auf der der Videoplayer gehostet wird. MediaTailor setzt diese Variable auf den Wert des Referer Headers, den der Player in seiner Anfrage verwendet hat MediaTailor. Wenn der Player diesen Header nicht angibt, lässt MediaTailor die Variable [session.referer] leer. Wenn Sie ein Content Delivery Network (CDN) oder einen Proxy vor dem Manifest-Endpunkt verwenden und möchten, dass diese Variable angezeigt wird, stellen Sie hier den richtigen Header des Players als Proxy bereit.
[session.user_agent]	Nein		Der User-Agent -Header, der von der Sitzungsinitialisierungsanfrage des Players MediaTailor empfangen wurde. Wenn vor dem Manifest-Endpunkt ein CDN oder ein Proxy angegeben wird, müssen Sie den korrekten Header vom Player hier als Proxy einfügen.

Name	Verfügbar für Ad-Prefetch	Abschnitt mit den SCTE-35-Spezifikationen	Description
[session.uuid]	Nein		Alternative zu. [session.id] Hierbei handelt es sich um eine eindeutige Kennung für die aktuelle Wiedergabesitzung, wie z. B. im Folgenden: e039fd39-09f0-46b2-aca9-9871cc116cde

Name	Verfügbar für Ad-Prefetch	Abschnitt mit den SCTE-35-Spezifikationen	Description
[avail.source_content_time_epoch_ms]	Nein		Für HLS ist der Wert die PDT des Ursprungssegments, mit dem die Verfügbarkeit gestartet wurde. Für DASH ist der Wert derjenige <code><Period></code>, <code><SupplementalProperty></code> <code>urn:scte:dash:utc-time</code> der den enthält. <code><EventStream></code> - Bei Preroll-HLS- oder DASH-Anzeigen <code>[avail.source_content_time_epoch_ms]</code> ist das PDT des ersten HLS-Segments und das des <code><SupplementalProperty></code> <code>urn:scte:dash:utc-time</code> ersten. <code><Period></code> Bei Livestreams mit kurzen Manifestfenstern ist dieser Wert für jeden Zuschauer unterschiedlich, je nachdem, wann er mit der Wiedergabe des Streams beginnt. Bei einem Livestream mit einem vollen Event-DVR-Fenster ist der Wert für jeden Zuschauer derselbe. - Bei teilweiser Verfügbarkeit aufgrund von Verfügbarkeitsunterbrechung <code>[avail.source_content_time_epoch_ms]</code> ist dies die PDT des Quellinhaltssegments, mit dem die Verfügbarkeit gestartet wurde. Wenn beispielsweise die ersten 20 Sekunden der Verfügbarkeit MediaTailor entfernt werden, <code>[avail.source_content_time_epoch_ms]</code> wird immer noch die ursprüngliche PDT verwendet und nicht um 20 Sekunden verschoben. - Für ADS-Prefetch-Anfragen <code>[avail.source_content_time_epoch_ms]</code> ist dieses Feld leer. Kunden können es jedoch mithilfe

Name	Verfügbar für Ad-Prefetch	Abschnitt mit den SCTE-35-Spezifikationen	Description

dynamischer Variablen für den Prefetch-Zeitplanabruf festlegen.

Example

Wenn der ADS erfordert, dass zusammen mit der eindeutigen Sitzungskennung ein Abfrageparameter mit dem Namen `deviceSession` übergeben wird, könnte die ADS-Vorlagen-URL in AWS Elemental MediaTailor wie folgt aussehen:

```
https://my.ads.server.com/path?deviceSession=[session.id]
```

AWS Elemental MediaTailor generiert automatisch einen eindeutigen Bezeichner für jeden Stream und gibt den Bezeichner anstelle von `ein.session.id` Wenn die ID lautet `1234567`, würde die letzte Anfrage, die MediaTailor an das ADS gestellt wird, in etwa so aussehen:

```
https://my.ads.server.com/path?deviceSession=1234567
```

Wenn für ADS mehrere Abfrageparameter übergeben werden müssen, AWS Elemental MediaTailor könnte die ADS-URL in der Vorlage wie folgt aussehen:

```
https://my.ads.server.com/sample?
e=[scte.avails_expected]&f=[scte.segment_num]&g=[scte.segments_expected]&h=[scte.sub_segment_num]
```

Das folgende Beispiel-XML-Fragment für einen DASH-Marker zeigt, wie man es benutzt:

```
<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>

```

Das folgende Beispiel-XML-Fragment für einen DASH-Marker zeigt, wie es verwendet wird `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: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>

```

Das folgende Beispiel-XML-Fragment für einen DASH-Marker zeigt, wie es verwendet wird `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>QW5vdGhlciB0ZXN0IHN0cmVuZyBmb3IgaWZlZjB2RpbmcgdG8gQmFzZTY0IGVuY29kZWQgYm1uYXJ5J5Lg=
Binary>
      </scte35:Signal>

```

```
</Event>
```

Das folgende Beispiel für ein HLS-Tag zeigt die Verwendung EXT-X-DATERANGE von:

```
#EXT-X-DATERANGE:ID="splice-6FFFFFFF0",START-DATE="2014-03-05T11:
15:00Z",PLANNED-DURATION=59.993,SCTE35-OUT=0xFC002F0000000000FF0
00014056FFFFFFF000E011622DCAFF0000526362000000000000A0008029896F50
0000087000000000
```

Das folgende Beispiel für ein HLS-Tag zeigt die Verwendung von: EXT-X-CUE-OUT

```
#EXT-OATCLS-SCTE35:/DA0AAAAAAAAAAAAABQb+ADAQ6QAeAhxDVUVJQAAA03/PAAEUrEoICAAAAAAg
+2UBNAAANvrtoQ==
#EXT-X-ASSET:CAID=0x00000000020FB6501
#EXT-X-CUE-OUT:201.467
```

Das folgende Beispiel für ein HLS-Tag zeigt die Verwendung von: EXT-X-SPLICEPOINT-SCTE35

```
#EXT-X-SPLICEPOINT-SCTE35:/DA9AAAAAAAAAAP/wBQb+uYbZqwAnAiVDVUVJAAAKqX//
AAEjW4AMEU1EU05CMDAXMTMyMjE5M190NAAAmXz5JA==
```

Das folgende Beispiel zeigt, wie scte35:Binary Decode verwendet wird:

```
{
  "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
  }
}

```

MediaTailor Player-Variablen für ADS-Anfragen

AWS Elemental MediaTailor sendet vom Player empfangene Daten an das ADS, wenn Sie die Angabe von `player_params.<query_parameter_name>` Variablen in der AWS Elemental MediaTailor ADS-URL der Vorlage konfigurieren. Wenn der Player beispielsweise einen `user_id` in seiner Anfrage genannten Abfrageparameter an sendet MediaTailor, um diese Daten in der ADS-Anfrage zu übergeben, fügen Sie diese `[player_params.user_id]` in die ADS-URL-Konfiguration ein.

Auf diese Weise können Sie kontrollieren, welche Abfrageparameter in die ADS-Anforderung eingeschlossen werden. In der Regel fügen Sie einen vom ADS erkannten speziellen Abfrageparameter zur ADS-Anforderungs-URL hinzu und geben Schlüssel-Wert-Paare als Wert des Parameters an.

Die Beispiele im folgenden Verfahren verwenden die folgenden Schlüssel-Wert-Paare:

- param1 mit einem Wert von value1:
- param2 mit einem Wert von value2:

So fügen Sie Abfrageparameter als Schlüssel-Wert-Paare hinzu

1. Konfigurieren Sie unter die URL der ADS-Anforderungsvorlage so, dass sie auf die Parameter verweist. AWS Elemental MediaTailor Die folgende URL veranschaulicht den Einschluss der Beispielparameter:

```
https://my.ads.com/path?param1=[player_params.param1]&param2=[player_params.param2]
```

2. (Optional) URL-codieren Sie für die serverseitige Web-Nachverfolgungsberichte die Schlüssel-Wert-Paare auf dem Player. Beim MediaTailor Empfang der Anfrage zur Sitzungsinitialisierung werden die Werte einmal per URL dekodiert, bevor sie in die ADS-Anforderungs-URL eingefügt werden.

Note

Wenn Ihr ADS einen URL-codierten Wert benötigt, URL-codieren Sie den Wert zweimal auf dem Player. Auf diese Weise MediaTailor führt die Dekodierung von zu einem einmal codierten Wert für das ADS.

Beispiel: Wenn die an den ADS gesendete Darstellung der Werte decodiert `param1=value1:&param2=value2:` ergibt, dann ist die URL-codierte Darstellung `param1=value1%3A&param2=value2%3A`.

3. Übergeben Sie beim Aufruf zur Sitzungsinitialisierung vom Player die Schlüssel-Wert-Paare MediaTailor als Wert eines einzelnen Abfrageparameters an. Die folgenden Beispielaufufe stellen die Beispiel-Schlüssel-Wert-Paare für die server- und clientseitigen Werbenachverfolgungs-Berichte bereit.

- Beispielanforderungen für serverseitige Nachverfolgungsberichte – mit URL-codierten Paaren

HLS:

```
<master>.m3u8?ads.param1=value1%3A&ads.param2=value2%3A
```

DASH:

```
<manifest>.mpd?ads.param1=value1%3A&ads.param2=value2%3A
```

- Beispielanforderung für clientseitige Werbenachverfolgungs-Berichte – ohne URL-Codierung

HLS:

```
POST <master>.m3u8
{
  "adsParams": {
    "param1": "value1:",
    "param2": "value2:"
  }
}
```

DASH:

```
POST <manifest>.mpd
{
  "adsParams": {
    "param1": "value1:",
    "param2": "value2:"
  }
}
```

Für serverseitige Berichte werden die Parameter MediaTailor dekodiert, wenn die Spieleraanfrage empfangen wird. Bei der clientseitigen Berichterstattung werden die in der JSON-Nutzlast empfangenen Parameter nicht geändert. MediaTailor sendet die folgende Anfrage an das ADS:

```
https://my.ads.com/<path>?param1=value1:&param2=value2:
```

Auf diese Weise werden die Schlüssel-Wert-Paare `param1` und `param2` als First-Class-Abfrageparameter in die ADS-Anforderung eingeschlossen.

MediaTailor Domain-Variablen für mehrere Inhaltsquellen

AWS Elemental MediaTailor Mit dynamischen Domänenvariablen können Sie mehrere Domains, wie z. B. den `my-ads-server.com`-Teil der URL `http://my-ads-server.com`, mit den Player-Parametern in Ihrer Konfiguration verwenden. Auf diese Weise können Sie mehr als eine Inhaltsquelle oder einen Ad Decision Server (ADS) in einer einzigen Konfiguration verwenden.

Sie können Domänenvariablen mit jedem Parameter verwenden, der einen URI enthält:

- `AdDecisionServerUrl`
- `AdSegmentUrlPrefix`
- `ContentSegmentUrlPrefix`
- `LivePreroll.AdDecisionServerUrl`
- `VideoContentSourceUrl`

Domänenvariablen werden zusammen mit Konfigurationsaliasnamen verwendet, um dynamische Variablen zu ersetzen. Konfigurationsalias ordnen den Player-Parametern, die für die dynamische Domänenkonfiguration verwendet werden, eine Reihe von Aliasnamen und Werten zu. Anleitungen zur Einrichtung finden Sie unter [Erstellen und Verwenden von Konfigurationsaliasnamen mit MediaTailor](#). Ausführliche Referenzinformationen finden Sie unter [MediaTailor Übersicht über Konfigurationsalias](#).

MediaTailor Übersicht über Konfigurationsalias

AWS Elemental MediaTailor Konfigurationsalias ermöglichen den dynamischen Variablenersatz in URL-Domänen und anderen unterstützten Feldern. Verwenden Sie diese Funktion, um mehrere Domänen zu verwenden und sie URLs während der Sitzungsinitialisierung dynamisch zu konfigurieren.

Anwendungsfälle

Konfigurationsalias ermöglichen ausgeklügelte Architekturen mit mehreren Konfigurationen für die folgenden Szenarien:

- Geografisches Routing: Leitet Anfragen anhand des Standorts des Betrachters mithilfe von regionsspezifischen Aliasnamen an unterschiedliche Herkunfts- oder Werbeserver weiter. [Anleitungen zur Implementierung finden CloudFront Sie unter Origin Failover.](#)
- Inhaltsbasiertes Routing: Leitet verschiedene Inhaltstypen an spezielle Quellen oder Verarbeitungspipelines weiter. Informationen zur Konfiguration des Routing-Verhaltens finden Sie unter. [the section called “Richten Sie das CDN-Routing-Verhalten ein”](#)
- Failover-Szenarien: Implementieren Sie Backup-Origins und Ad-Server mit automatischem Failover mithilfe von Alias-Switching. Eine ausführliche Implementierung finden Sie unter [Implementieren Sie Resilienz für mehrere Regionen MediaTailor mit MQAR](#) und. [the section called “Planen Sie die CDN-Integration”](#)
- A/B-Tests: Testen Sie verschiedene Werbeserver, Ursprünge oder Konfigurationen, indem Sie den Traffic auf der Grundlage der Player-Parameter weiterleiten. Anleitungen zu Belastungstests finden Sie unter [Vorbereiten und Ausführen von Leistungstests für Amazon CloudFront mit echter Benutzerüberwachung.](#)
- Gerätespezifische Optimierung: Optimieren Sie die Bereitstellung von Inhalten und die Anzeigenschaltung für verschiedene Gerätetypen oder Funktionen. Umfassende Anleitungen finden Sie unter. [Richten Sie die Manifestfilterung mit MediaTailor MediaPackage, und CDN ein](#)
- Lastenausgleich: Verteilen Sie die Last mithilfe von dynamischem Routing auf mehrere Origins oder Ad-Server. Anleitungen zur Implementierung finden Sie unter [Implementieren Sie Resilienz für mehrere Regionen MediaTailor mit MQAR](#) und [the section called “Planen Sie die CDN-Integration”](#).

Unterstützte Felder

Sie können dynamische Variablen in den folgenden Konfigurationsfeldern verwenden:

- VideoContentSourceUrl
- AdDecisionServerUrl
- LivePreroll.AdDecisionServerUrl
- AdSegmentUrlPrefix
- ContentSegmentUrlPrefix
- TranscodeProfileName
- SlateAdUrl
- StartUrl
- EndUrl

In den folgenden Abschnitten wird beschrieben, wie Konfigurationsaliasse verwendet werden.

Themen

- [Erstellen und Verwenden von Konfigurationsaliasnamen mit MediaTailor](#)
- [Konfigurations-Alias mit MediaTailor Anwendungsbeispiel](#)

Erstellen und Verwenden von Konfigurationsaliasnamen mit MediaTailor

Bevor Sie mit der Verwendung von Domänenvariablen beginnen, erstellen Sie Konfigurationsaliasse für Ihre Konfiguration. Sie verwenden die Konfigurationsaliasse bei der Initialisierung der Sitzung als Variablen für den Domänenersatz.

Einschränkungen

Beachten Sie die folgenden Einschränkungen bei der Verwendung von Konfigurationsaliasen:

- Alle in der Domain verwendeten dynamischen Variablen müssen als `ConfigurationAliases` dynamische Variable definiert werden.
- Den Player-Parametervariablen muss ein Präfix vorangestellt werden. `player_params`. Beispiel, `player_params.origin_domain`.
- Die Liste der Aliaswerte muss für Domänenvariablen in Critical URLs (`VideoContentSourceUrl`, `AdSegmentUrlPrefix`) vollständig sein.
`ContentSegmentUrlPrefix`
- Wenn eine Anfrage für eine Domainvariable mit dem Status Critical gestellt wird URLs , die entweder die dynamische Variable nicht spezifiziert oder einen ungültigen Alias verwendet, schlägt die Anfrage mit einem 400 HTTP-Statuscode fehl. Bei unkritischen Feldern (`SlateAdUrl`, `TranscodeProfileName`, Bumper URLs) werden zwar Warnungen protokolliert, die Anfrage wird jedoch nicht fehlschlagen.

Ausweichverhalten bei fehlenden Aliasen

Wenn Konfigurationsaliasse nicht gefunden werden oder ungültig sind, MediaTailor implementiert das folgende Fallback-Verhalten:

- Domain-Variablen: Wenn ein Domainvariablen-Alias fehlt oder ungültig ist, schlägt die Anfrage mit dem HTTP 400-Statuscode fehl. Für alle Domänenvariablen müssen gültige Aliase definiert sein.

- **Nicht-Domain-Variablen:** Bei Variablen, die in Teilen verwendet werden, die nicht zur Domäne gehören URLs (wie Pfadelemente oder Abfrageparameter), führen fehlende Aliase dazu, dass leere Zeichenketten ersetzt werden.
- **Konfigurationsvalidierung:** MediaTailor überprüft, ob alle erforderlichen Aliase bei der Erstellung und Aktualisierung der Konfiguration vorhanden sind.

Schritt 1: Konfigurations-Aliase erstellen

Gehen Sie wie folgt vor, um Konfigurationsalias für den Domainersatz mithilfe der MediaTailor Konsole zu erstellen.

Console

So erstellen Sie Konfigurationsalias mithilfe der Konsole

1. Öffnen Sie die MediaTailor Konsole unter <https://console.aws.amazon.com/mediatailor/>
2. Wählen Sie auf der Konfigurationsseite im Abschnitt Konfigurationsalias die Option Player-Parameter hinzufügen aus.
3. Geben Sie unter Spielerparameter den Namen des Spielerparameters ein, den Sie als dynamische Variable verwenden möchten. Beispiel, `player_params.origin_domain`.
4. Geben Sie unter Aliase die Aliase und ihre Werte ein, die Sie für den Player-Parameter verwenden möchten.
5. Wählen Sie OK aus.

AWS Elemental MediaTailor zeigt den neuen Parameter in der Tabelle im Abschnitt Konfigurations-Aliase an.

6. Wiederholen Sie die vorherigen Schritte, um weitere Player-Parameter hinzuzufügen.
7. Wählen Sie Speichern.

API

Um Konfigurationsalias mithilfe der API zu erstellen

Wenn Sie eine MediaTailor Konfiguration erstellen oder aktualisieren, verwenden Sie den `ConfigurationAliases` Parameter mit der folgenden JSON-Struktur:

```
{
```

```
"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"
  }
}
```

Schritt 2: Verwenden Sie Konfigurationsalias bei der Sitzungsinitialisierung

Nachdem Sie die Konfigurationsalias eingerichtet haben, können Sie sie als Ersatzvariablen für Domänen in Ihrer Anfrage zur Sitzungsinitialisierung verwenden. Auf diese Weise können Sie die Domänen für Ihre Sitzung dynamisch konfigurieren.

Example Beispiel für Alias für die Grundkonfiguration

Hier ist ein einfaches Beispiel für eine Konfiguration, die Konfigurationsalias und dynamische Domänenvariablen umfasst:

```
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 Initialisierung der Sitzung mit Aliasen

Bei Verwendung der vorherigen Konfiguration würde eine Anfrage zur Sitzungsinitialisierung unter Verwendung der Spielervariablen und Aliase etwa wie folgt aussehen:

```
POST index.m3u8
{
  "playerParams": {
    "origin_domain": "pdx",
    "region": "pdx",
    "endpoint_id": "pdx",
    "ad_type": "customized"
  }
}
```

MediaTailor ersetzt die Aliaszeichenfolgen durch die zugewiesenen Werte in der Konfiguration der Konfigurationsalias.

Die Anfrage an das ADS wird wie folgt aussehen:

```
https://abc.execute-api.us-west-2.amazonaws.com/ads?sid=[session.id]&ad_type=abc12345
```

Die Anfrage an die Quelle für die Manifeste wird wie folgt aussehen:

```
https://abc.mediapackage.us-west-2.amazonaws.com/out/v1/abcd
```

Konfigurations-Alias mit MediaTailor Anwendungsbeispiel

Die folgenden Beispiele zeigen, wie eine vollständige MediaTailor Konfiguration mit Konfigurationsaliasen, eine Anfrage zur Sitzungsinitialisierung mit Aliasen und der Verarbeitungsablauf für Aliase aussehen.

Example Vollständige Konfiguration mit Aliasen

Das folgende Beispiel zeigt eine vollständige Konfiguration, die Konfigurationsaliasse und dynamische Domänenvariablen umfasst:

```
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 Initialisierung der Sitzung mit Aliasen

Das folgende Beispiel zeigt eine Anfrage zur Sitzungsinitialisierung, in der die Spielervariablen und Aliase angegeben werden:

```
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 Ablauf der Parameterverarbeitung

Im folgenden Beispiel werden die Aliaszeichenfolgen MediaTailor durch die zugewiesenen Werte in den Konfigurationsaliasnamen ersetzt. Die Verarbeitung führt zu den folgenden Anfragen:

- ADS-Anfrage:

```
https://abc.execute-api.us-west-2.amazonaws.com/ads?sid=[session.id]&ad_type=abc12345
```

- VideoContentSource Anfrage:

```
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 Parameter an ADS übergeben

AWS Elemental MediaTailor unterstützt die Einrichtung dynamischer Variablen in MediaTailor Anfragen an das ADS mithilfe der folgenden Schritte.

- Hinweise zur unterstützten Formatierung von Abfrageparametern finden Sie unter [MediaTailor Parameterreferenz und Einschränkungen](#).
- Informationen zu Konfigurationsaliasnamen und Domänenvariablen finden Sie unter [MediaTailor Übersicht über Konfigurationsalias](#).
- Weitere Anpassungen der ADS-Anfrage finden Sie unter [Fortgeschrittene Nutzung](#)

Methoden zur Sitzungsinitialisierung

MediaTailor unterstützt mehrere Methoden zur Sitzungsinitialisierung und Parameterübergabe:

1. POST mit dem Hauptteil der Anfrage:

```
POST <master>.m3u8
{
  "adsParams": {"param1": "value1", "param2": "value2"},
  "playerParams": {"param3": "value3"}
}
```

2. Abfrageparameter in der URL:

```
GET <master>.m3u8?ads.param1=value1&ads.param2=value2&playerParams.param3=value3
```

Important

Sie können Parameter bei der Initialisierung nur einmal angeben. Konfigurationsalias werden vor der Weiterleitung in tatsächliche Werte aufgelöst.

So übergeben Sie Sitzungs- und Player-Informationen an den ADS

1. Ermitteln Sie gemeinsam mit dem ADS, welche Informationen benötigt werden, um auf eine Anzeigenanfrage von AWS Elemental MediaTailor zu antworten.

2. Erstellen Sie eine Konfiguration MediaTailor , die eine ADS-Vorlagen-URL verwendet, die die ADS-Anforderungen erfüllt. Schließen Sie in die URL statische Parameter und Platzhalter für dynamische Parameter ein. Geben Sie Ihre Vorlagen-URL in das Feld Ad decision server (Entscheidungsserver für Werbung) ein.

In der folgenden Beispielvorlagen-URL bietet `correlation` Sitzungsdaten und `deviceType` bietet Player-Daten:

```
https://my.ads.server.com/path?  
correlation=[session.id]&deviceType=[player_params.deviceType]
```

3. Konfigurieren Sie auf dem Player die Sitzungsinitiierungs-Anforderung für AWS Elemental MediaTailor , um Parameter für die Player-Daten bereitzustellen. Sie brauchen die Parameter nur in die Sitzungsinitiierungs-Anforderung einzuschließen und können sie in nachfolgenden Anforderungen für die Sitzung weglassen.

Die Art des Aufrufs, den der Player zur Initialisierung der Sitzung tätigt, bestimmt, ob der Player (Client) oder MediaTailor (Server) Werbeverfolgungsberichte für die Sitzung bereitstellt. Weitere Informationen zu diesen beiden Optionen finden Sie unter [Daten melden und verfolgen](#).

Führen Sie je nachdem, ob Sie an server- oder clientseitigen Werbenachverfolgungs-Berichten interessiert sind, den jeweiligen der folgenden Aufrufe durch. Bei beiden Beispielaufrufen ist `userID` für den ADS und `auth_token` für den Ursprungs-Server vorgesehen:

- (Option) Aufruf für serverseitiges Ad-Tracking-Reporting — Stellen Sie den Parametern, die Sie an das ADS senden MediaTailor möchten, ein Präfix voran. `ads` Lassen Sie das Präfix bei Parametern weg, die MediaTailor an den Ursprungs-Server senden soll:

Die folgenden Beispiele zeigen eingehende Anfragen für HLS und DASH an. AWS Elemental MediaTailor verwendet das `deviceType` in seiner Anfrage an das ADS und das `auth_token` in seiner Anfrage an den Ursprungsserver.

HLS-Beispiel:

```
GET master.m3u8?ads.deviceType=ipad&auth_token=kjhdsaf7gh
```

DASH-Beispiel:

```
GET manifest.mpd?ads.deviceType=ipad&auth_token=kjhdsaf7gh
```

- (Option) Aufruf für clientseitiges Ad-Tracking-Reporting — Geben Sie Parameter für das ADS innerhalb eines Objekts an. `adsParams`

HLS-Beispiel:

```
POST master.m3u8
{
  "adsParams": {
    "deviceType": "ipad"
  }
}
```

DASH-Beispiel:

```
POST manifest.mpd
{
  "adsParams": {
    "deviceType": "ipad"
  }
}
```

Wenn der Player eine Sitzung initiiert, werden die Variablen in der ADS-Anforderungs-URL der Vorlage durch die Sitzungsdaten und die Parameter des Players AWS Elemental MediaTailor ersetzt. Die verbleibenden Parameter werden vom Player an den Ursprungs-Server übergeben.

Example MediaTailor Anfragen mit Anzeigenvariablen

Die folgenden Beispiele zeigen die Aufrufe von AWS Elemental MediaTailor an den ADS und den Ursprungs-Server entsprechend den voranstehenden Beispielen für Sitzungsinitialisierungs-Aufrufe des Players:

- MediaTailor ruft das ADS mit Sitzungsdaten und dem Gerätetyp des Players auf:

```
https://my.ads.server.com/path?correlation=896976764&deviceType=ipad
```

- MediaTailor ruft den Ursprungsserver mit dem Autorisierungstoken des Spielers auf.
- HLS-Beispiel:

```
https://my.origin.server.com/master.m3u8?auth_token=kjhdsaf7gh
```

- DASH-Beispiel:

```
https://my.origin.server.com/manifest.mpd?auth_token=kjhdsaf7gh
```

Fortgeschrittene Nutzung

Sie können die ADS-Anforderung in vielerlei Hinsicht mit Player- und Sitzungsdaten anpassen. Sie müssen nur den ADS-Hostnamen angeben.

Die folgenden Beispiele zeigen einige der Möglichkeiten, wie Sie Ihre Anforderung anpassen können:

- Erstellen Sie durch Verketteten von Player-Parametern und Sitzungsparametern neue Parameter. Beispiel:

```
https://my.ads.com?key1=[player_params.value1][session.id]
```

- Verwenden Sie einen Player-Parameter als Teil eines Pfadelements. Beispiel:

```
https://my.ads.com/[player_params.path]?key=value
```

- Verwenden Sie Player-Parameter, um nicht nur die Werte, sondern sowohl Pfadelemente als auch die Schlüssel selbst zu übergeben. Beispiel:

```
https://my.ads.com/[player_params.path]?[player_params.key1]=[player_params.value1]
```

MediaTailor Parameter-Routing für ADS und Origins

AWS Elemental MediaTailor leitet Abfrageparameter basierend auf ihrem Präfix und Zweck an verschiedene Ziele weiter. Das Verständnis des Parameter-Routing ist wichtig, um ursprungsspezifische Funktionen wie die zeitversetzte Anzeige mit implementieren zu können. MediaPackage

Regeln für das Parameter-Routing

MediaTailor verwendet die folgenden Routing-Regeln für Abfrageparameter:

- Origin-Parameter (kein Präfix): Parameter ohne ein bestimmtes Präfix werden für ursprungsspezifische Funktionen an den Original-Server weitergegeben

- ADS-Parameter (**ads.**Präfix): Parameter mit dem Präfix `ads.` werden an den Ad Decision Server gesendet
- Manifestparameter (**manifest.**Präfix): Parameter mit dem Präfix `manifest.` werden für das CDN-Routing und die Autorisierung verwendet

Example Beispiel für Parameter-Routing

Die folgende Sitzungsinitialisierung demonstriert das Parameter-Routing:

```
POST /v1/session/123456789/originId/index.m3u8
{
  "adsParams": {
    "param1": "value1",
    "param2": "value2"
  },
  "manifestParams": {
    "auth_token": "abc123"
  }
}
```

In diesem Beispiel:

- `param1` und `param2` werden an das ADS gesendet
- `auth_token` wird für CDN-Routing und Autorisierung verwendet
- Parameter ohne Präfixe würden an den Ursprungsserver übergeben

Verhalten der Parameter des Origin-Servers

An Original-Server übergebene Parameter ermöglichen ursprungsspezifische Funktionen wie zeitversetzte Anzeige, Inhaltsfilterung und Authentifizierung.

Allgemeine Anwendungsfälle für Ursprungsparameter

Origin-Parameter werden häufig verwendet für:

- Zeitversetzte Anzeige: `start` und `end` Parameter für MediaPackage zeitversetzte Inhalte
- Inhaltsauthentifizierung: Authentifizierungstoken, die vom Ursprungsserver benötigt werden
- Inhaltsfilterung: Parameter, die steuern, welche Inhaltsvarianten zurückgegeben werden

- Ursprungsspezifische Funktionen: Alle Parameter, die der Ursprungsserver für die Inhaltsverarbeitung verwendet

 Important

Parameter werden bei der Sitzungsinitialisierung verarbeitet und während der gesamten Sitzung beibehalten. Um Parameter wie Zeitverschiebungsfenster zu ändern, müssen Sie eine neue Sitzung mit aktualisierten Werten erstellen.

MediaTailor und MediaPackage zeitversetzte Anzeigeintegration

AWS Elemental MediaTailor kann zeitversetzte Anzeigeparameter an MediaPackage Ursprünge weitergeben, um Startover- und Nachholfunktionen zu ermöglichen. Durch diese Integration können Zuschauer bereits zu einem früheren Zeitpunkt mit dem Ansehen von Live-Inhalten beginnen.

MediaPackage zeitversetzte Anzeigeparameter

MediaPackage unterstützt die folgenden zeitversetzten Anzeigeparameter, die übergeben werden können: MediaTailor

- **start**: Epoche oder ISO 8601-Zeitstempel, der den Beginn des zeitversetzten Manifests definiert
- **end**: Epoche oder ISO 8601-Zeitstempel, der das Ende des zeitversetzten Manifests definiert
- **time_delay**: Verzögert die Verfügbarkeit von Inhalten um bestimmte Sekunden
- **manifest_window_seconds**: Fordert ein Manifest an, das kürzer als das konfigurierte Fenster ist

Example MediaTailor Sitzungsinitialisierung mit MediaPackage zeitversetzten Parametern

Das folgende Beispiel zeigt, wie eine Sitzung mit zeitversetzten Anzeigeparametern initialisiert wird:

```
GET /v1/master/123456789/originId/index.m3u8?  
start=2024-08-26T10:00:00Z&end=2024-08-26T11:00:00Z
```

Oder mit expliziter Sitzungsinitialisierung:

```
POST /v1/session/123456789/originId/index.m3u8
```

```
{
  "adsParams": {
    "param1": "value1"
  }
}
```

Mit zusätzlichen Abfrageparametern:

```
?start=2024-08-26T10:00:00Z&end=2024-08-26T11:00:00Z
```

Verhalten der Parameter während Sitzungen

Zeitversetzte Anzeigeparameter weisen spezifische Verhaltensmerkmale auf:

- Sitzungsinitialisierung: Parameter werden verarbeitet, wenn die Sitzung erstellt wird
- Persistenz der Parameter: Die Parameter bleiben während der gesamten Wiedergabe mit der Sitzung verknüpft
- Nach der Initialisierung unveränderlich: Parameter können während einer aktiven Sitzung nicht geändert werden
- Neue Sitzung erforderlich: Um Zeitverschiebungsfenster zu ändern, erstellen Sie eine neue Sitzung mit aktualisierten Parameterwerten

MediaPackage Anforderungen an das Startfenster

Damit die zeitversetzte Anzeige funktioniert MediaPackage, müssen Sie Folgendes sicherstellen:

1. Konfigurieren Sie ein Startfenster auf Ihrem MediaPackage Endpunkt (bis zu 24 Stunden)
2. Stellen Sie sicher, dass Ihr CDN die erforderlichen Abfrageparameter weiterleitet an MediaPackage
3. Verwenden Sie konsistente Wiedergabefenster für alle Player-Sitzungen, um das CDN-Caching zu verbessern
4. Stellen Sie sicher, dass die Start- und Endzeiten innerhalb des konfigurierten Startfensters liegen

⚠ Important

Verwenden Sie bei zeitversetzter Anzeige einheitliche Wiedergabefenster für alle Spielsitzungen, anstatt für jeden Zuschauer eindeutige Start- oder Endzeiten zu generieren. Dies führt zu einem besseren Caching im CDN und vermeidet mögliche Drosselungen.

Vollständige Informationen zur Konfiguration und zu den Parametern der MediaPackage zeitversetzten Anzeige finden Sie unter [Zeitversetzte Anzeige](#) mit im Benutzerhandbuch. AWS Elemental MediaPackage

MediaTailor Verhalten und Persistenz von Parametersitzungen

AWS Elemental MediaTailor verarbeitet Parameter bei der Sitzungsinitialisierung und verwaltet sie während des gesamten Sitzungslebenszyklus. Das Verständnis des Sitzungsverhaltens ist für die Implementierung dynamischer Parameterszenarien von entscheidender Bedeutung.

Methoden zur Sitzungsinitialisierung

MediaTailor unterstützt mehrere Methoden zur Sitzungsinitialisierung mit Parametern:

1. Implizite Sitzungsinitialisierung: Parameter, die in der ersten Manifestanforderung enthalten sind

```
GET /v1/master/123456789/originId/index.m3u8?  
manifest.auth_token=abc123&start=2024-08-26T10:00:00Z
```

2. Explizite Sitzungsinitialisierung (POST): Im Hauptteil der Anfrage angegebene Parameter

```
POST /v1/session/123456789/originId/index.m3u8  
{  
  "adsParams": {"param1": "value1"},  
  "manifestParams": {"auth_token": "abc123"}  
}
```

3. Explizite Sitzungsinitialisierung (GET): Als Abfrageparameter bereitgestellte Parameter

```
GET /v1/session/123456789/originId/index.m3u8?  
ads.param1=value1&manifestParams.auth_token=abc123
```

Persistenz und Unveränderlichkeit von Parametern

MediaTailor Das Verhalten der Parameter folgt diesen Regeln:

- Einmalige Angabe: Parameter können nur einmal angegeben werden, und zwar bei der Sitzungsinitialisierung
- Sitzungsweite Persistenz: Die Parameter werden während der gesamten Sitzung beibehalten
- Nach der Initialisierung unveränderlich: Parameter können nach der Erstellung der Sitzung nicht geändert werden
- Auflösung von Konfigurations-Aliassen: Aliase werden in tatsächliche Werte aufgelöst, bevor sie an Ziele weitergeleitet werden

Szenarien zur Änderung von Parametern

Um Parameter während der Wiedergabe zu ändern:

- Neue Sitzung erstellen: Initialisieren Sie eine neue Sitzung mit aktualisierten Parameterwerten
- Spielerübergang: Der Spieler wechselt nahtlos zur neuen Sitzung
- Vererbung von Parametern: Übernehmen Sie unveränderte Parameter, um die Konsistenz zu wahren

ExampleÄnderung der Time-Shift-Parameter

Um von einem 1-Stunden-Fenster zu einem 2-Stunden-Fenster zu wechseln:

1. Aktuelle Sitzung: `start=2024-08-26T10:00:00Z&end=2024-08-26T11:00:00Z`
2. Neue Sitzung erstellen: `start=2024-08-26T10:00:00Z&end=2024-08-26T12:00:00Z`
3. Den Spieler auf die neue Sitzungs-URL umstellen

Important

Bei mehreren multivarianten Playlist-Anfragen für eine einzelne Sitzung werden die Parameter nach der ersten Anfrage nicht aktualisiert. Die Parameter bleiben für die Dauer der Sitzung unveränderlich.

MediaTailor Parameterreferenz und Einschränkungen

Bevor Sie dynamische Anzeigenvariablen konfigurieren, sollten Sie sich mit den Anforderungen und Einschränkungen für die Parameterformatierung vertraut machen, die für alle MediaTailor Konfigurationen gelten.

AWS Elemental MediaTailor bietet umfassende Informationen zu Zeichenbeschränkungen und Längenbeschränkungen für Parameter sowie zu unterstützten Formaten sowohl für Manifestabfrageparameter als auch für ADS-Parameter.

Zeichenbeschränkungen für Manifest-Abfrageparameter

Manifest-Abfrageparameter unterstützen bestimmte Zeichen und haben definierte Längenbeschränkungen.

Unterstützte Zeichen (ohne URL-Kodierung)

Sie können die folgenden Zeichen direkt in Manifest-Abfrageparametern verwenden:

- Alphanumerische Zeichen (A-Z, a-z, 0-9)
- Perioden (.)
- Bindestriche (-)
- Unterstriche (_)
- Backslashes (\)

Unterstützte Zeichen mit URL-Kodierung

Die folgenden Sonderzeichen werden unterstützt, wenn sie URL-kodiert sind:

- Zeitraum (.) = %2E
- Gedankenstrich (-) = %2D
- Unterstrich (_) = %5F
- Prozent (%) = %25
- Tilde (~) = %7E
- Schrägstrich (/) = %2F
- Sternchen (*) = %2A
- entspricht (=) = %3D

- Frage (?) = %3F

Unterstützung für URL-Kodierung

MediaTailor unterstützt das Prozentzeichen (%), wenn Sie es in der URL-Kodierung verwenden (z. B. hello%20world = hello world). Sie können beliebige URL-kodierte Zeichen verwenden, sofern es sich um gültige URL-Kodierungen gemäß der HTTP-Spezifikation handelt.

Nicht unterstützte Zeichen

Sie können die folgenden Zeichen nicht in Manifest-Abfrageparametern ohne URL-Kodierung verwenden: : , , , ? , & =% , / (Schrägstrich).

Important

MediaTailor unterstützt keine Doppelzeichen wie %%% oder ==. Aufgrund von Zeichenbeschränkungen können Sie keine vollständigen Abfrageparameterwerte URLs als Manifest verwenden.

Längenbeschränkungen

Die Gesamtlänge aller Manifest-Abfrageparameter (Schlüssel und Werte zusammen) darf 2000 Zeichen nicht überschreiten.

Längenbeschränkungen für ADS-Parameter

Die folgenden Längenbeschränkungen gelten für Parameter, die in Anfragen an das ADS verwendet werden:

- ADS-Parametername: Maximal 10.000 Zeichen
- ADS-Parameterwert: Maximal 25.000 Zeichen
- ADS-URL: Maximal 25.000 Zeichen

MediaTailor Leitfaden zur Fehlerbehebung bei Parametern

AWS Elemental MediaTailor bietet Anleitungen zur Behebung häufiger Probleme im Zusammenhang mit Parametern MediaTailor, einschließlich Zeichenbeschränkungen, Problemen mit der URL-Kodierung und Fehlern bei Konfigurationsalias.

Fehler bei der Zeichenbeschränkung

Parameterwerte, die Zeichen enthalten, die nicht unterstützt werden, können zu Fehlern oder unerwartetem Verhalten führen.

Häufige Symptome

Die folgenden Symptome können auf Probleme mit der Zeichenbeschränkung hinweisen:

- Parameter, die nicht im Manifest vorkommen URLs
- HTTP 400-Fehler bei der Sitzungsinitialisierung
- Verkürzte oder beschädigte Parameterwerte
- ADS-Anfragen schlagen aufgrund eines fehlerhaften Formats fehl URLs

Lösungsschritte

So beheben Sie Fehler bei der Zeichenbeschränkung:

1. Überprüfen Sie die Parameterwerte für nicht unterstützte Zeichen: : ?, , & , = , % /
2. Wenden Sie die richtige URL-Kodierung für Sonderzeichen an (siehe [MediaTailor Parameterreferenz und Einschränkungen](#))
3. Vermeiden Sie Doppelzeichen wie oder %%% ==
4. Ziehen Sie alternative Parameterformate in Betracht, wenn full URLs nicht verwendet werden kann

Example Beispiel für die URL-Codierung

Anstatt zu verwenden:

```
manifest.redirect_url=https://example.com/path?param=value
```

Verwenden Sie das URL-kodierte Format:

```
manifest.redirect_url=https%3A%2F%2Fexample.com%2Fpath%3Fparam%3Dvalue
```

Fehler bei der Längenbegrenzung

Parameter, die Längenbeschränkungen überschreiten, können gekürzt werden oder Fehler verursachen.

Längenbeschränkungen

Es gelten die folgenden Längenbeschränkungen (vollständige [MediaTailor Parameterreferenz und Einschränkungen](#) Informationen finden Sie unter):

- Manifest-Abfrageparameter (insgesamt): 2000 Zeichen
- ADS-Parameternamen: 10.000 Zeichen
- ADS-Parameterwerte: 25.000 Zeichen
- ADS URLs: 25.000 Zeichen

Lösungsstrategien

Um Längenbeschränkungen zu handhaben:

1. Verwenden Sie nach Möglichkeit kürzere Parameternamen und -werte
2. Teilen Sie große Parameterwerte in mehrere kleinere Parameter auf
3. Verwenden Sie Konfigurationsalias, um kurze Aliase längeren Werten zuzuordnen (siehe [MediaTailor Übersicht über Konfigurationsalias](#))
4. Erwägen Sie die Verwendung eines externen Speichers für große Datenmengen mit Parameterreferenzen

Alias-Fehler bei der Konfiguration

Probleme mit Konfigurationsalias können zu HTTP 400-Fehlern oder unerwarteten Parameterwerten führen.

Häufige Fehler bei Konfigurations-Alias

Die folgenden Fehler treten häufig bei Konfigurationsaliasnamen auf:

- HTTP 400-Fehler: Fehlender oder ungültiger Aliaswert
- Domänenvariablen werden nicht korrekt aufgelöst
- Player-Parameter werden nicht durch Aliaswerte ersetzt

Checkliste für die Lösung

So beheben Sie Fehler bei Konfigurations-Alias:

1. Stellen Sie sicher, dass alle Domänenvariablen definiert sind als `ConfigurationAliases`
2. Stellen Sie sicher, dass Player-Parametervariablen ein `player_params.` Präfix
3. Vergewissern Sie sich, dass die Liste der Aliaswerte für Domänenvariablen im kritischen Bereich URLs (`VideoContentSourceUrl`, `AdSegmentUrlPrefix`) vollständig ist
`ContentSegmentUrlPrefix`
4. Vergewissern Sie sich, dass Anfragen zur Sitzungsinitialisierung gültige Aliaswerte angeben
5. Überprüfen Sie die JSON-Struktur des Parameters `ConfigurationAliases`

Eine ausführliche Anleitung zur Fehlerbehebung finden Sie unter [MediaTailor Leitfaden zur Fehlerbehebung bei Konfigurationsaliasen](#).

Example Validierung von Konfigurations-Alias

Stellen Sie sicher, dass Ihre Konfiguration alle erforderlichen Aliase enthält:

```
"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
  }
}
```

Probleme mit dem Ablauf der Parameterverarbeitung

Wenn Sie den Ablauf der Parameterverarbeitung verstehen, können Sie Probleme bei der Parameterweiterleitung und -transformation beheben.

Reihenfolge der Parameterverarbeitung

MediaTailor verarbeitet Parameter in der folgenden Reihenfolge:

1. Überprüfung der Parameter für die Initialisierung der Sitzung
2. Auflösung des Konfigurationsalias (falls zutreffend)
3. Parameterfilterung (ADS im Vergleich zu Origin und Manifest)
4. URL-Kodierung und -Formatierung
5. Parameteranwendung auf URLs

Debuggen des Parameterflusses

So debuggen Sie Probleme mit der Parameterverarbeitung:

1. Stellen Sie sicher, dass die Parameter bei der Sitzungsinitialisierung korrekt angegeben wurden
2. Stellen Sie sicher, dass die Konfigurationsalias zu den erwarteten Werten aufgelöst werden
3. Vergewissern Sie sich, dass die Parameter korrekt angezeigt werden URLs (Manifest, ADS, Origin)
4. Stellen Sie sicher, dass die URL-Codierung korrekt angewendet wurde

Example Beispiel für einen Parameterfluss

Initialisierung der Sitzung:

```
POST master.m3u8
{
  "playerParams": {"origin_domain": "pdx"},
  "manifestParams": {"test": "123"}
}
```

Nach der Auflösung und Verarbeitung des Alias:

- Herkunft der Anfrage: `https://abc.mediapackage.us-west-2.amazonaws.com/out/v1/abcd`
- Manifest-URL: `/v1/master/.../index.m3u8?aws.sessionId=session&test=123`

Sicherheitsüberlegungen und bewährte Methoden

MediaTailor implementiert Sicherheitsmaßnahmen für die Parameterbehandlung, um allgemeine Sicherheitsprobleme zu vermeiden.

Sicherheitsmaßnahmen

MediaTailor implementiert die folgenden Sicherheitsmaßnahmen:

1. Beschränkungen der Eingabegröße, um eine Überlastung der Datenbank zu verhindern
2. Richtige Kodierung und Bereinigung von Benutzereingaben

3. URL-Kodierung der Eingabe, um eine Beschädigung der Antwort zu verhindern

Best Practices

Folgen Sie diesen bewährten Methoden für die sichere Handhabung von Parametern:

- Überprüfen Sie die Parameterwerte auf der Clientseite, bevor Sie sie senden
- Verwenden Sie Konfigurationsalias, um mögliche Parameterwerte einzuschränken
- Vermeiden Sie es, vertrauliche Informationen in Parameter aufzunehmen
- Überwachen Sie die Verwendung von Parametern auf ungewöhnliche Muster
- Halten Sie die Parameterwerte innerhalb der empfohlenen Längengrenzen

MediaTailor Leitfaden zur Fehlerbehebung bei Konfigurationsaliasen

AWS Elemental MediaTailor bietet systematische Anleitungen zur Fehlerbehebung bei häufigen Problemen mit Konfigurationsaliasen und Fehlerszenarien.

Fehler bei der Validierung von Konfigurations-Alias

Wenn Konfigurationsalias fehlen oder ungültig sind, werden spezifische Fehlerantworten MediaTailor zurückgegeben, um das Problem zu identifizieren.

Häufige Fehlerszenarien

In der folgenden Tabelle werden häufig auftretende Fehler bei Konfigurationsalias und ihre Lösungsschritte beschrieben:

Fehler	Ursache	Auflösung
HTTP 400: Ungültiger Player-Parameter-Alias	Der Wert des Spielerparameters wurde nicht gefunden in ConfigurationAliases	Stellen Sie sicher, dass der Player-Parameterwert als Schlüssel in der entsprechenden ConfigurationAliases Zuordnung existiert
HTTP 400: Der erforderliche Konfigurationsalias fehlt	Die Domänenvariable wurde ohne entsprechende Aliaszuordnung hinzugefügt	Fügen Sie den fehlenden Player-Parameter ConfigurationAliases mit allen erforderlichen Aliaszuordnungen hinzu

Fehler	Ursache	Auflösung
	enden ConfigurationAlias es Eintrag verwendet	
HTTP 400: Die Konfigurationsüberprüfung ist fehlgeschlagen	ConfigurationAliases Die Struktur ist fehlerhaft oder unvollständig	Überprüfen Sie die JSON-Struktur und stellen Sie sicher, dass alle Domänenvariablen über entsprechende Aliase verfügen
Ersetzung einer leeren Zeichenfolge in URLs	Alias für Nicht-Domain-Variablen wurde nicht gefunden	Fügen Sie eine fehlende Aliaszuordnung hinzu oder geben Sie den Standardwert in an ConfigurationAliases

Checkliste für die Validierung

Verwenden Sie die folgende Checkliste, um die Einrichtung Ihrer Konfigurationsalias zu überprüfen:

1. Abdeckung von Domänenvariablen: Stellen Sie sicher, dass alle Variablen, die in Domänenteilen von verwendet werden, entsprechende Einträge URLs haben ConfigurationAliases
2. Vollständigkeit der Alias: Stellen Sie sicher, dass alle möglichen Player-Parameterwerte als Schlüssel in den Aliaszuordnungen enthalten sind
3. JSON-Struktur: Stellen Sie sicher, dass das ConfigurationAliases JSON richtig formatiert und verschachtelt ist
4. Benennung der Parameter: Stellen Sie sicher, dass alle Player-Parameter das Präfix verwenden `player_params`.
5. Einheitlichkeit der Werte: Stellen Sie sicher URLs, dass Aliaswerte für den vorgesehenen Verwendungszweck (Profilnamen usw.) gültig sind

Aliasauflösung der Konfiguration beim Debuggen

Folgen Sie diesem systematischen Ansatz, um Probleme mit der Auflösung von Konfigurationsalias zu debuggen.

Step-by-step Debugging-Methodik

Gehen Sie wie folgt vor, um Probleme mit Konfigurationsalias zu identifizieren und zu lösen:

Verfahren zum Debuggen von Konfigurationsalias

1. Überprüfen Sie die Konfigurationsstruktur: Stellen Sie sicher, dass Ihre Wiedergabekonfiguration korrekt formatierte Dateien enthält ConfigurationAliases

```
{
  "ConfigurationAliases": {
    "player_params.example_param": {
      "alias1": "value1",
      "alias2": "value2"
    }
  }
}
```

2. Überprüfen Sie das Format der Player-Parameter: Stellen Sie sicher, dass die Sitzungsinitialisierung korrekt formatierte Player-Parameter enthält

```
{
  "playerParams": {
    "example_param": "alias1"
  }
}
```

3. Alias-Zuordnung validieren: Vergewissern Sie sich, dass der Player-Parameterwert („alias1“) als Schlüssel in der Zuordnung existiert ConfigurationAliases
4. Testen Sie mit einer einfachen Konfiguration: Beginnen Sie mit einer Minimalkonfiguration, um das Problem zu isolieren
5. Überwachen Sie die Fehlerantworten: Überprüfen Sie die MediaTailor Fehlerantworten auf bestimmte Bestätigungsmeldungen
6. Überprüfen Sie, ob das Problem gelöst wurde URLs: Stellen Sie sicher, dass die endgültigen Lösungen gültig und zugänglich URLs sind

Bewährte Methoden für Konfigurationsalias

Folgen Sie diesen bewährten Methoden, um eine zuverlässige Implementierung von Konfigurationsalias sicherzustellen.

Sicherheitsüberlegungen

Implementieren Sie die folgenden Sicherheitsmaßnahmen, wenn Sie Konfigurationsalias verwenden:

- **Eingabevalidierung:** Überprüfen Sie alle Player-Parameterwerte, bevor Sie sie bei der Alias-Auflösung verwenden
- **Bereinigung von Aliaswerten:** Stellen Sie sicher, dass Aliaswerte nur die erwarteten Zeichen und Formate enthalten
- **Domain-Einschränkungen:** Beschränken Sie Domain-Aliase auf vertrauenswürdige, kontrollierte Domains
- **Zugriffskontrolle:** Beschränken Sie Konfigurationsänderungen nur auf autorisiertes Personal

Leistungsoptimierung

Optimieren Sie die Leistung von Konfigurationsalias mit diesen Empfehlungen:

- **Minimierung der Aliasanzahl:** Verwenden Sie nur die benötigten Aliase, um den Verarbeitungsaufwand zu reduzieren
- **Effiziente Benennung:** Verwenden Sie klare, konsistente Benennungskonventionen für Aliase und Parameter
- **Standardwerte:** Stellen Sie sinnvolle Standardalias für allgemeine Anwendungsfälle bereit
- **Konfigurations-Caching:** Nutzen Sie MediaTailor das Konfigurations-Caching, um die Leistung zu verbessern

Wartung und Überwachung

Sorgen Sie mit diesen Methoden für zuverlässige Alias-Alias-Konfigurationen:

- **Regelmäßige Überprüfung:** Stellen Sie regelmäßig sicher, dass alle Aliaszuordnungen aktuell und funktionsfähig sind
- **Fehlerüberwachung:** Achten Sie auf HTTP 400-Fehler, die auf fehlende oder ungültige Aliase zurückzuführen sind
- **Dokumentation:** Sorgen Sie für eine klare Dokumentation aller Aliaszuordnungen und ihrer Zwecke
- **Testverfahren:** Implementieren Sie umfassende Tests für alle Alias-Kombinationen

MediaTailor Manifest-Abfrageparameter

AWS Elemental MediaTailor verarbeitet Abfrageparameter für verschiedene Zwecke: Manifest-Abfrageparameter für CDN-Routing und Autorisierung sowie andere Abfrageparameter, die für ursprungsspezifische Funktionen verwendet werden können.

AWS Elemental MediaTailor bewahrt Abfrageparameter aus der Sitzungsinitialisierung und hängt sie an personalisierte Manifeste und andere Ressourcen an. URLs Verwenden Sie diese Funktion, wenn Sie ein Content Delivery Network (CDN) zwischen MediaTailor und dem Client Player haben.

Verwenden Sie Manifest-Abfrageparameter, wenn Ihr CDN die Abfrageparameter für Folgendes benötigt:

- Dynamisches Routing zu verschiedenen Endpunkten MediaTailor
- Token-Autorisierung

Verhalten auf der Clientseite im Vergleich zum CDN

MediaTailor hängt Abfrageparameter für clientseitige Berichtsendpunkte an, fügt sie jedoch nicht für CDN-Segmente an. Die aktualisierte Funktionalität bietet eine umfassendere Unterstützung für Abfrageparameter für verschiedene MediaTailor Ressourcen und erhöht so die Flexibilität bei Anwendungsfällen für CDN-Routing und Autorisierung.

MediaTailor hängt Abfrageparameter für clientseitige Berichtsendpunkte an, aber die Abfrageparameter für die CloudFront (oder andere CDN-) Segmente werden nicht angehängt.

Um die Beibehaltung von Parametern zu verwenden, wenden Sie sich an den [AWS-Support](#), um zu beantragen, dass die Weitergabe von Manifestabfrageparametern aktiviert wird.

Das Verhalten variiert zwischen HLS und DASH sowie zwischen expliziter und impliziter Sitzungsinitialisierung. In den folgenden Themen wird beschrieben, wie Anfragen zur Sitzungsinitialisierung so konfiguriert werden, dass Parameter MediaTailor an das Manifest übergeben werden:

MediaTailor Behandlung von Abfrageparametern für Ursprünge

AWS Elemental MediaTailor behandelt Abfrageparameter je nach Verwendungszweck unterschiedlich. Manifest-Abfrageparameter (mit dem Präfix `manifest.`) werden für das CDN-Routing und die Autorisierung verwendet, während andere Abfrageparameter für ursprungsspezifische Funktionen verwendet werden können.

Zeitversetzte Anzeige mit MediaPackage

Für die zeitversetzte Anzeigefunktion mit MediaPackage können Sie end Parameter in Ihre Anfragen start einbeziehen. Diese Parameter definieren spezifische Inhaltsfenster für die Anzeige von Startover und Catch-up.

Example zeitversetzte Betrachtungsanfrage

Nehmen Sie Start- und Endparameter in Ihre Manifestanforderung auf, damit die Anzeige zeitversetzt angezeigt wird:

```
GET /v1/master/111122223333/originId/index.m3u8?  
start=2024-08-26T10:00:00Z&end=2024-08-26T11:00:00Z
```

Verhalten der Parameter während Sitzungen

Abfrageparameter werden bei der Sitzungsinitialisierung verarbeitet. Für zeitversetzte Ansichten oder andere ursprungsspezifische Funktionen:

- Sitzungsinitialisierung: Fügen Sie der ersten Manifestanforderung die erforderlichen Parameter hinzu
- Persistenz der Parameter: Die Parameter sind der Sitzung zugeordnet und werden während der gesamten Wiedergabe beibehalten
- Parameter ändern: Um Zeitverschiebungsfenster oder andere Parameter zu ändern, erstellen Sie eine neue Sitzung mit aktualisierten Werten

Important

Die spezifische Behandlung von Abfrageparametern hängt von Ihrer ursprünglichen Konfiguration und den Parametern ab, die Ihr Inhaltsursprung unterstützt. Stellen Sie für die MediaPackage Integration sicher, dass Ihr CDN so konfiguriert ist, dass es die erforderlichen Abfrageparameter weiterleitet, wie unter beschrieben [Konfigurieren Sie wichtige Abfrageparameter](#).

MediaTailor Zeichenbeschränkungen für Parameter und URL-Kodierung

AWS Elemental MediaTailor unterstützt bestimmte Zeichen in Manifest-Abfrageparametern. Sie können die URL-Kodierung für Sonderzeichen verwenden.

Unterstützte Zeichen mit URL-Kodierung

Die folgenden Sonderzeichen werden bei der URL-Kodierung unterstützt:

- Zeitraum (.) = %2E
- Gedankenstrich (-) = %2D
- Unterstrich (_) = %5F
- Prozent (%) = %25
- Tilde (~) = %7E
- Schrägstrich (/) = %2F
- Sternchen (*) = %2A
- entspricht (=) = %3D
- Frage (?) = %3F

Unterstützung für URL-Kodierung

MediaTailor unterstützt das Prozentzeichen (%), wenn Sie es in der URL-Kodierung verwenden (z. B. hello%20world = hello world). Sie können beliebige URL-kodierte Zeichen verwenden, sofern es sich um gültige URL-Kodierungen gemäß der HTTP-Spezifikation handelt.

Important

MediaTailor unterstützt keine Doppelzeichen wie%%% oder ==.

Sicherheitsüberlegungen

MediaTailor implementiert die folgenden Sicherheitsmaßnahmen für die Parameterbehandlung:

1. Beschränkungen der Eingabegröße, um eine Überlastung der Datenbank zu verhindern
2. Richtige Kodierung und Bereinigung von Benutzereingaben
3. URL-Kodierung der Eingabe, um eine Beschädigung der Antwort zu verhindern

Themen

- [MediaTailor Implizite HLS-Sitzungsinitialisierung](#)

- [MediaTailor Implizite Initialisierung der DASH-Sitzung](#)
- [MediaTailor Explizite Initialisierung von HLS- und DASH-Sitzungen](#)
- [MediaTailor protokollspezifisches Verhalten der Parameter](#)
- [MediaTailor CDN-Integration und Parameter-Routing](#)

MediaTailor Implizite HLS-Sitzungsinitialisierung

AWS Elemental MediaTailor schließt Abfrageparameter in Links zu MediaTailor Ressourcen ein, wenn Ihre Anfrage Abfrageparameter mit dem Schlüssel enthält. `manifest.*` Das folgende Beispiel zeigt dieses Anforderungsformat:

```
GET /v1/master/111122223333/originId/index.m3u8?manifest.test=123&other=456
```

Links enthalten das `manifest.*` Präfix nicht.

Anwendung von Parametern in HLS

MediaTailor Wendet bei impliziten HLS-Sitzungen Parameter auf die folgenden Stellen in der Manifesthierarchie an:

- Multivariante Playlist URLs
- Medien-Playlist URLs
- Inhaltssegment URLs
- Anzeigensegment URLs
- HLS-Initialisierung URLs

Example multivariante Playlist

Das folgende Beispiel zeigt, wie MediaTailor die Abfrageparameter in die URL für die multivariante Playlist aufgenommen werden:

```
#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 Medien-Playlist

Das folgende Beispiel zeigt, wie MediaTailor die Abfrageparameter in die URLs Inhaltssegmente aufgenommen werden:

```
#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 Implizite Initialisierung der DASH-Sitzung

AWS Elemental MediaTailor erstellt eine Sitzung für den Client und leitet sie mit Abfrageparametern um, wenn der Client eine Manifestanforderung ohne Sitzung stellt. Das folgende Beispiel zeigt dieses Anforderungsformat:

```
GET /v1/dash/111122223333/originId/index.mpd?manifest.test=123&other=456
```

MediaTailor erstellt eine Sitzung für den Client und leitet sie mit den Abfrageparametern weiter:

```
/v1/dash/111122223333/originId/index.mpd?sessionId=session&test=123
```

Anwendung von Parametern in DASH

Die Antwort auf das DASH-Manifest umfasst die Abfrageparameter an verschiedenen Stellen, einschließlich Inhaltssegmenten, Anzeigensegmenten und Initialisierung URLs. MediaTailor wendet Parameter auf Folgendes an:

- Elemente des DASH-Manifests:
- SegmentTemplate Initialisierungsattribute

- SegmentTemplate Medienattribute
- Inhaltssegment URLs
- Anzeigensegment URLs

Wenn der Client die Anfrage stellt, MediaTailor antwortet er mit einem DASH-Manifest, das dem folgenden Beispiel ähnelt. Die erste Periode ist eine Inhaltsperiode, weshalb der Manifest-Abfrageparameter dort MediaTailor nicht eingefügt wird. In der zweiten Periode, bei der es sich um eine Werbeperiode handelt, wird der Manifest-Abfrageparameter in das `initialization` Attribut und `media` das Attribut des `SegmentTemplate` Elements MediaTailor eingefügt. Das `Location` Element hat auch die Manifest-Abfrageparameter.

```
<?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 Explizite Initialisierung von HLS- und DASH-Sitzungen

AWS Elemental MediaTailor schließt die manifestParams AS-Abfrageparameter in die multivariante Playlist und das Tracking URLs in die Antwort ein, wenn der Client eine explizite Anfrage zur Sitzungsinitialisierung stellt.

Methoden zur Sitzungsinitialisierung

Für die explizite Sitzungsinitialisierung können Sie entweder POST mit Anforderungstext oder GET mit Abfrageparametern verwenden:

1. POST mit Anforderungstext:

```

POST /v1/session/111122223333/originId/index.m3u8
{
    "adsParams": {"param1": "value1", "param2": "value2", "param3": "value3"},
    "manifestParams": {"test": "123"}
}

```

2. GET mit Abfrageparametern:

```
GET /v1/session/111122223333/originId/index.m3u8?  
ads.param1=value1&ads.param2=value2&manifestParams.test=123
```

Example Anfrage zur Initialisierung der Sitzung

```
POST /v1/session/111122223333/originId/index.m3u8  
{  
  "adsParams": {  
    "param1": "value1",  
    "param2": "value2",  
    "param3": "value3"  
  },  
  "manifestParams": {  
    "test": "123"  
  },  
  "reportingMode": "client"  
}
```

Example Manifest und Antwort zur Nachverfolgung

```
{  
  "manifestUrl": "/v1/master/111122223333/originId/index.m3u8?  
aws.sessionId=session&test=123",  
  "trackingUrl": "/v1/tracking/111122223333/originId/session?test=123"  
}
```

Manifestantworten für die Sitzung haben einen spezifischen manifestParams Charakter, der den zuvor beschriebenen impliziten Workflows zur Sitzungsinitialisierung MediaTailor URLs ähnelt. Der Hauptunterschied besteht darin, dass Manifestparameter für die explizite Sitzungsinitialisierung nicht mit beginnen. manifest.

Manifest-Abfrageparameter sind unveränderlich und können nur bei der Sitzungsinitialisierung festgelegt werden. Wenn ein Client mehrere multivariante Playlist-Anfragen für eine einzelne Sitzung stellt, werden die Manifest-Abfrageparameter nach der ersten Anfrage MediaTailor nicht aktualisiert.

Ablauf der Parameterverarbeitung

Sie können Parameter bei der Initialisierung nur einmal angeben. Konfigurationsaliase werden vor der Weiterleitung in tatsächliche Werte aufgelöst. Beispiel: `player_params.ad_type=customized` wird `ad_type=abc12345` basierend auf der Konfiguration zu aufgelöst. `ConfigurationAliases`

MediaTailor protokollspezifisches Verhalten der Parameter

AWS Elemental MediaTailor behandelt Manifest-Abfrageparameter für HLS- und DASH-Protokolle unterschiedlich. Jeder Protokolltyp hat spezifische Anwendungsorte und Verarbeitungsmethoden.

Vergleich der Parameterhandhabung zwischen HLS und DASH

In der folgenden Tabelle wird verglichen, wie MediaTailor mit Manifest-Abfrageparametern in den Protokollen HLS und DASH umgegangen wird:

Aspekt	HLS-Verhalten	DASH-Verhalten
Anwendung von Parametern	Wird direkt auf Manifest URLs und Segment angewendet URLs	Wird auf Standortelemente, SegmentTemplate Attribute und Segmente angewendet URLs
Manifeste Hierarchie	Multivariante Playlist → Medien-Playlisten → Segmente	MPD → Perioden → Repräsentationen AdaptationSets
Initialisierung URLs	Wird auf HLS URLs Init angewendet, falls vorhanden	Wird auf Initialisierungsattribute SegmentTemplate angewendet
Behandlung von Sitzungen	Parameter bleiben bei Playlist-Updates erhalten	Im MPD Location-Element enthaltene Parameter sorgen für Sitzungskontinuität
Handhabung von Anzeigensegmenten	Wird auf das Anzeigensegment URLs in Medien-Playlisten angewendet	Wird auf SegmentTemplate Medienattribute für den Zeitraum der Anzeige angewendet

Speicherorte von Parameteranwendungen

MediaTailor wendet Manifest-Abfrageparameter auf die folgenden Speicherorte an:

Anwendung von HLS-Parametern

MediaTailor Wendet bei HLS-Streams Manifest-Abfrageparameter an auf:

- Multivariante Playlist URLs: Parameter werden an Verweise auf Medien-Playlisten angehängt
- Medien-Playlist URLs: Parameter sind in Segmenten URLs innerhalb von Medien-Playlisten enthalten
- Inhaltssegment URLs: Alle Inhaltssegmente enthalten die Manifest-Abfrageparameter
- Anzeigensegment URLs: Anzeigensegmente erhalten Parameter für das CDN-Routing und die Autorisierung
- HLS-Initialisierung URLs: Init-Segmente enthalten Parameter, wenn sie im Stream vorhanden sind
- Slate-Segment URLs: Der Slate-Inhalt enthält Parameter für ein konsistentes CDN-Verhalten

Example Anwendungsbeispiel für HLS-Parameter

Angesichts der Sitzungsinitialisierung:

```
GET /v1/master/123456789/originId/index.m3u8?  
manifest.auth_token=abc123&manifest.region=us-west
```

Die multivariante Playlist enthält Parameter in Verweisen auf Medien-Playlisten:

```
#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
```

Die Medien-Playlist enthält Parameter in Segmenten: URLs

```
#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
```

Anwendung der DASH-Parameter

MediaTailor Wendet für DASH-Streams Manifest-Abfrageparameter an auf:

- MPD-Standortelemente: Das Location-Element enthält Parameter für Manifest-Aktualisierungsanforderungen
- SegmentTemplate Initialisierungsattribute: Das Init-Segment enthält Parameter URLs
- SegmentTemplate Medienattribute: URL-Vorlagen für Mediensegmente enthalten Parameter
- Inhaltssegment URLs: Alle anhand von Vorlagen generierten Inhaltssegmente enthalten Parameter
- Anzeigensegment URLs: Die Segmente für den Anzeigenzeitraum enthalten Parameter für die CDN-Integration
- Serverseitige Berichtsweiterleitungen: 302-Weiterleitungen zu Anzeigensegmenten behalten die Parameter bei

Example Anwendungsbeispiel für DASH-Parameter

Angesichts der Initialisierung der Sitzung:

```
GET /v1/dash/123456789/originId/index.mpd?  
manifest.auth_token=abc123&manifest.region=us-west
```

Das DASH-Manifest enthält Parameter an mehreren Stellen:

```
<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-Integration und Parameter-Routing

AWS Elemental MediaTailor Manifest-Abfrageparameter ermöglichen anspruchsvolle CDN-Integrationsszenarien. Sie können sie für dynamisches Routing, Autorisierung und Lastenausgleich verwenden.

Anwendungsfälle für CDN-Routing

Zu den häufigsten CDN-Integrationsszenarien, die von Manifest-Abfrageparametern profitieren, gehören:

- **Geografisches Routing:** Leitet Anfragen auf der Grundlage des Standorts des Betrachters an regionsspezifische MediaTailor Endpunkte weiter
- **Tokenbasierte Autorisierung:** Übergeben Sie Autorisierungstoken über das CDN an, um sicheren Zugriff auf Inhalte zu erhalten MediaTailor
- **Lastenausgleich:** Verteilen Sie den Datenverkehr mithilfe der CDN-Routing-Logik auf mehrere MediaTailor Endpunkte
- **A/B-Tests:** Leiten Sie verschiedene Benutzersegmente zu Testzwecken an unterschiedliche Konfigurationen weiter MediaTailor
- **Gerätespezifische Optimierung:** Leitet Anfragen basierend auf Gerätetyp oder Funktionen weiter

Beibehaltung der Parameter auf allen CDN-Ebenen

MediaTailor stellt sicher, dass die Manifest-Abfrageparameter über mehrere CDN-Ebenen und Anforderungstypen hinweg beibehalten werden:

1. **Erste Anfrage:** Die Parameter werden aus der Anfrage zur Initialisierung der Sitzung extrahiert
2. **Manifestgenerierung:** Parameter werden auf alle relevanten Elemente URLs im Manifest angewendet
3. **Segmentanfragen:** Parameter sind in allen Segmenten enthalten, um ein URLs konsistentes CDN-Verhalten zu gewährleisten
4. **Anzeigeneinfügung:** Die Parameter werden beim Einfügen von Anzeigen und beim Ersetzen von Segmenten beibehalten

Example Ablauf der CDN-Autorisierung

Das folgende Beispiel zeigt einen vollständigen CDN-Autorisierungsablauf unter Verwendung von Manifest-Abfrageparametern:

1. Manifest für Client-Anfragen mit Autorisierungstoken:

```
GET https://cdn.example.com/mediatailor/v1/master/123456789/originId/index.m3u8?
manifest.auth_token=jwt_token_here&manifest.user_id=12345
```

2. CDN leitet die Anfrage MediaTailor mit folgenden Parametern weiter:

```
GET https://mediatailor.amazonaws.com/v1/master/123456789/originId/index.m3u8?
manifest.auth_token=jwt_token_here&manifest.user_id=12345
```

3. MediaTailor generiert ein Manifest mit Parametern, die auf alle angewendet werden: URLs

```
#EXTM3U
#EXT-X-STREAM-INF:BANDWIDTH=2665212
../.../manifest/123456789/originId/session/0.m3u8?
auth_token=jwt_token_here&user_id=12345
```

4. Nachfolgende Segmentanfragen enthalten Parameter für die CDN-Autorisierung:

```
GET https://cdn.example.com/mediatailor/segment/123456789/originId/session/0/1?
auth_token=jwt_token_here&user_id=12345
```

Daten melden und verfolgen

MediaTailor bietet zwei Optionen für die Nachverfolgung und Berichterstattung darüber, wie viel von einer Anzeige sich ein Zuschauer angesehen hat. Bei der serverseitigen Anzeigenberichterstattung wird die Anzeige MediaTailor nachverfolgt und Beacons (Tracking-Signale) direkt an den Anzeigenserver gesendet. Alternativ verfolgt beim clientseitigen Tracking-Ansatz der Client-Player (das Gerät des Nutzers) die Anzeige und sendet die Beacons an den Anzeigenserver. Die Art der Anzeigenberichterstattung, die in einer Playback-Sitzung verwendet wird, hängt von der spezifischen Anfrage ab, mit der der Player die Sitzung initiiert. MediaTailor

Hinweise zur Weitergabe von Sitzungs- und Spielerdaten mithilfe dynamischer Variablen an den Anzeigenserver finden Sie unter [MediaTailor dynamische Anzeigenvariablen für ADS-Anfragen](#).

Einzelheiten zu den Initialisierungsparametern von Sitzungen finden Sie unter [MediaTailor Manifest-Abfrageparameter](#).

Themen

- [MediaTailor Serverseitiges Ad-Tracking und Reporting](#)
- [Kundenseitiges Anzeigen-Tracking](#)

MediaTailor Serverseitiges Ad-Tracking und Reporting

AWS Elemental MediaTailor verwendet standardmäßig serverseitige Berichte für eine umfassende Nachverfolgung und Messung von Anzeigen. Wenn der Player bei der serverseitigen Berichterstellung eine Werbe-URL vom Manifest anfordert, meldet der Service die Werbekonsumierung direkt der Werbeverfolgungs-URL. Nachdem der Player eine Wiedergabesitzung mit initialisiert hat MediaTailor, sind keine weiteren Eingaben von Ihnen oder dem Player erforderlich, um serverseitige Berichte zu erstellen. MediaTailor Sendet bei der Wiedergabe jeder Anzeige Beacons an den Anzeigenserver, um zu melden, wie viel von der Anzeige angesehen wurde. MediaTailor sendet Beacons für den Beginn der Anzeige und für den Verlauf der Anzeige in Quartilen: erstes Quartil, Mittelpunkt, drittes Quartil und Abschluss der Anzeige.

So führen Sie die serverseitige Werbe-Berichterstellung durch

- Initialisieren Sie vom Player aus eine neue MediaTailor Wiedergabe-Sitzung mit einer Anfrage in einem der folgenden Formate gemäß Ihrem Protokoll:
 - Beispiel: HLS-Format

```
GET <mediatailorURL>/v1/master/<hashed-account-id>/<origin-id>/<asset-id>?  
ads.<key-value-pairs-for-ads>&<key-value-pairs-for-origin-server>
```

- Beispiel: DASH-Format

```
GET <mediatailorURL>/v1/dash/<hashed-account-id>/<origin-id>/<asset-id>?ads.<key-  
value-pairs-for-ads>&<key-value-pairs-for-origin-server>
```

Die Schlüssel-Wert-Paare sind die dynamischen Targeting-Parameter für die Werbenachverfolgung. Weitere Informationen zum Hinzufügen von Parametern zur Anforderung finden Sie unter [the section called “Dynamische Anzeigenvariablen”](#).

AWS Elemental MediaTailor beantwortet die Anfrage mit der Manifest-URL. Das Manifest enthält URLs für die Medienmanifeste. Die Medien-Manifeste enthalten eingebettete Links für Werbesegment-Anforderungen.

 Note

Wenn in MediaTailor einer Tracking-URL ein doppelter Schrägstrich (//) vorkommt, werden die Schrägstriche zu einem (/) zusammengefasst.

Wenn der Player die Wiedergabe über eine Anzeigensegment-URL (/v1/segmentPfad) anfordert, AWS Elemental MediaTailor sendet er das entsprechende Beacon über das Anzeigen-Tracking an den Anzeigenserver. URLs Gleichzeitig gibt der Service eine Umleitung zum tatsächlichen *.ts-Werbesegment aus. Das Anzeigensegment befindet sich entweder in der CloudFront Amazon-Distribution, in der transkodierte Anzeigen MediaTailor gespeichert werden, oder im Content Delivery Network (CDN), in dem Sie die Anzeige zwischengespeichert haben.

In den folgenden Abschnitten finden Sie weitere Informationen zur Arbeit mit serverseitigem Ad-Tracking von MediaTailor

Themen

- [Serverseitiges Tracking mit serverseitiger Anzeigeneinfügung \(SGAI\)](#)
- [Glossar für serverseitige Tracking-Beacons](#)
- [Serverseitiges Tracking-Timing und Caching-Verhalten](#)
- [MediaTailor Funktionen und Funktionen zur serverseitigen Nachverfolgung](#)

Serverseitiges Tracking mit serverseitiger Anzeigeneinfügung (SGAI)

Wenn Sie die serverseitige Anzeigeneinfügung (SGAI) verwenden, verwendet das serverseitige Tracking einen Beacons-Mechanismus ohne Sitzung, der sich von dem oben beschriebenen Ansatz im gestickten Modus unterscheidet. Anstatt Anzeigensegmente in das Inhaltsmanifest MediaTailor einzufügen (wo /v1/segment Anfragen nachverfolgt werden), gibt SGAi Anzeigenverweise als separate Playlisten in einer Asset-Listen-Antwort mit in die Anzeige eingebetteten Beacon-Metadaten zurück. URIs

So funktioniert serverseitiges Beaconsing ohne Sitzung

In den folgenden Schritten wird beschrieben, wie serverseitiges Beaconsing für SGAI-Sitzungen funktioniert:

1. **Sitzungsinitialisierung:** Der Player fordert die multivariante HLS-Wiedergabeliste mit an. `aws.insertionMode=GUIDED` Serverseitige Berichterstattung ist die Standardeinstellung (kein `aws.reportingMode` Parameter erforderlich). Im Gegensatz zum Modus „Zusammenfügen“ enthält die Antwort auf die Sitzungsinitialisierung kein. `trackingUrl`
2. **Zwischenspeicherbares Manifest:** MediaTailor Gibt ein zwischenspeicherbares Manifest zurück, das `EXT-X-DATERANGE` Tags `CLASS="com.apple.hls.interstitial"` und `X-ASSET-LIST` Attribute enthält, die auf den Endpunkt der MediaTailor interstitiellen Asset-Liste verweisen.
3. **Asset-Liste mit Beacon-Metadaten:** Wenn der Player auf eine Werbeunterbrechung stößt, ruft er die Asset-Liste ab. MediaTailor gibt eine JSON-Antwort zurück, in der jede Anzeigen-URI verschlüsselte Beacon-Metadaten enthält:

```
{
  "ASSETS": [
    {
      "DURATION": 30.0,
      "URI": "https://cdn.example.com/ad/master.m3u8?
awsBeaconData=<encrypted>&awsBeaconDomain=<MediaTailor-
endpoint>&awsConfigurationName=<config-name>"
    }
  ]
}
```

Wenn die serverseitige Berichterstattung aktiv ist, enthält die Antwort keinen TRACKING Abschnitt. Die Anzeige enthält URIs alle Beacon-Daten.

4. **HLS-Variablenersetzung:** Der Player ruft die multivariante Anzeigen-Playlist ab. Das Anzeigenmanifest verwendet `#EXT-X-DEFINE:QUERYPARAM` Direktiven, um die Beacon-Parameter aus der URI-Abfragezeichenfolge per HLS-Variablenersetzung an das Segment zu übergeben: URLs

```
#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
```

Der Player löst die `{awsConfigurationName}` Variablen `{awsBeaconData}``{awsBeaconDomain}`, und anhand der Werte aus der URI-Abfragezeichenfolge für das Anzeigenmanifest auf und fordert dann jedes Anzeigensegment an. MediaTailor

5. Auslösung eines Beacons bei Segmentanforderung: Wenn der Player jedes Anzeigensegment anfordert, wird die Anfrage weitergeleitet. MediaTailor Der Dienst entschlüsselt die Beacon-Daten, bestimmt die Position des Segments innerhalb der Anzeige (Impression, erstes Quartil, Mittelpunkt, drittes Quartil oder vollständig) und sendet das entsprechende VAST-Tracking-Beacon an den Anzeigenserver. MediaTailor leitet den Player dann zum eigentlichen Anzeigeninhaltssegment weiter.

Spieleranforderungen für serverseitiges SGAI-Beaconing

Um serverseitiges Beaconing mit SGAI verwenden zu können, muss Ihr Player die folgenden Anforderungen erfüllen:

- HLS Version 11 oder höher
- Support für EXT-X-DATERANGE mit CLASS Attribut für HLS Interstitials
- Support für #EXT-X-DEFINE: QUERYPARAM Variablensubstitution (RFC 8216bis). Der Spieler muss die Werte der Abfrageparameter prozentual dekodieren, bevor er sie in ein Segment einfügt. URLs

Note

Serverseitiges SGAI-Beaconing wird derzeit nur für HLS unterstützt. DASH wird für serverseitiges SGAI-Beaconing noch nicht unterstützt.

Vergleich mit serverseitigem Tracking im Stitched-Modus

In der folgenden Tabelle wird zusammengefasst, wie sich serverseitiges Tracking zwischen gestickter und serverseitiger Anzeigeneinfügung unterscheidet:

Aspekt	Gestickt (SSAI)	Servergeführt (SGAI)
Offensichtliche Cache-Fähigkeit	Pro Sitzung, nicht zwischenspeicherbar	Cachefähig, von allen Zuschauern gemeinsam genutzt
Routing von Anzeigensegmenten	Durch die /v1/segment/ Verwendung der Sitzungs-ID	Durch die /v1/segment/ Verwendung eines verschlüsselten Beacon-Datenblobs
Sitzungsstatus für Beacons	Pro Sitzung gespeichert in MediaTailor	Sitzungslos — der gesamte Status wird im verschlüsselten Parameter übernommen <code>awsBeaconData</code>
Tracking-URL beim Start der Sitzung	Wurde in der Antwort auf die Initialisierung der Sitzung zurückgegeben	Nicht angegeben — Beacon-Daten sind in die Anzeige URIs in jeder Antwort auf die Asset-Liste eingebettet
DASH-Unterstützung	Unterstützt	Noch nicht unterstützt

Note

Für Live-SGAI-Sitzungen können Sie das manifestbasierte Vorabrufen von Anzeigen mithilfe von `aws.guidedPrefetchMode=MANIFEST` aktivieren. Dies ist unabhängig von der zeitplanbasierten Prefetch-API, die bei zusammengestellten Sitzungen (SSAI) verwendet wird. Details hierzu finden Sie unter [Geführter Prefetch mit offensichtlichem Herzschlag](#).

Glossar für serverseitige Tracking-Beacons

MediaTailor Beim serverseitigen Tracking wird ein standardisierter Satz von Beacons verwendet, um den Fortschritt der Anzeige an Anzeigenserver und Bestätigungsdienste zu melden. Diese Beacons entsprechen den Standards des Interactive Advertising Bureau (IAB) für die Messung von Videoanzeigen und bieten genaue Berichte über Anzeigenimpressionen und Abschlussquoten.

Typen serverseitiger Tracking-Beacons

Typ des Beacons	Wann gefeuert	Zweck	Einzelheiten zum Zeitplan
Eindruck	Wenn der Spieler das erste Anzeigensegment anfordert	Weist darauf hin, dass der Anzeigeninhalt geladen wurde und demnächst dem Zuschauer angezeigt wird	Wird bei der ersten /v1/segment Anforderung einer Anzeige ausgelöst. Entspricht den IAB-Richtlinien, nach denen Anzeigeninhalte geladen werden müssen, bevor eine Impression gezählt wird. Die vollständige Workflow zur serverseitigen Nachverfolgung Reihenfolge finden Sie unter.
Starten	Wenn der Player mit dem Rendern des Anzeigeninhalts beginnt	Bestätigt, dass die Anzeigenwiedergabe tatsächlich gestartet wurde	Wird normalerweise gleichzeitig mit dem Impression-Beacon bei der ersten Segmentanforderung ausgelöst, stellt jedoch den tatsächlichen Beginn des Anzeigenrenderings dar. Diese Unterscheidung ist wichtig für Verifizierungsdienste, die sowohl Impressions- als auch Startereignisse getrennt verfolgen.
Erstes Quartil	Wenn der Spieler 25% der Anzeigendauer erreicht	Measures setzte die Anzahl der Werbeanzeigen während des ersten Quartals der Anzeige fort	Wird ausgelöst, wenn der Player das Segment anfordert, das den 25% -Punkt der Anzeigendauer enthält. Bei einer 20-Sekunden-Anzeige mit 2-Sekunden-Segmenten würde dies in der Regel bei der Anfrage für das dritte

Typ des Beacons	Wann gefeuert	Zweck	Einzelheiten zum Zeitplan
			Segment ausgelöst (etwa 4-6 Sekunden nach Beginn der Anzeige).
Mittelpunkt	Wenn der Spieler 50% der Anzeigendauer erreicht	Measures hat die Anzeige während der Hälfte der Werbeanzeige fortgesetzt	Wird ausgelöst, wenn der Player das Segment anfordert , das den 50% -Punkt der Anzeigendauer enthält. Bei einer 20-Sekunden-Anzeige mit 2-Sekunden-Segmenten würde dies in der Regel bei der Anfrage für das 5. Segment ausgelöst (etwa 8 bis 10 Sekunden nach Beginn der Anzeige).
Drittes Quartil	Wenn der Spieler 75% der Anzeigendauer erreicht	Measures hat die Anzeige während drei Vierteln der Werbeanzeige fortgesetzt	Wird ausgelöst, wenn der Player das Segment anfordert , das den 75% -Punkt der Anzeigendauer enthält. Bei einer 20-Sekunden-Anzeige mit 2-Sekunden-Segmenten würde dies in der Regel bei der Anfrage für das 8. Segment ausgelöst (etwa 14-16 Sekunden nach Beginn der Anzeige).

Typ des Beacons	Wann gefeuert	Zweck	Einzelheiten zum Zeitplan
Vollständig	Wenn der Spieler das Ende der Anzeige erreicht	Bestätigt, dass dem Zuschauer die gesamte Anzeige zugestellt wurde	Wird ausgelöst, wenn der Player das letzte Segment der Anzeige anfordert. Dies deutet darauf hin, dass der Zuschauer möglicherweise den gesamten Anzeigeninhalt gesehen hat. Bei einer 20-Sekunden-Anzeige mit 2-Sekunden-Segmenten würde dies in der Regel bei der Anfrage für das 10. Segment ausgelöst (etwa 18 bis 20 Sekunden nach Beginn der Anzeige).

Note

Der genaue Zeitpunkt der Auslösung des Beacons hängt von der Dauer des Segments und der Länge der Anzeige ab. MediaTailor berechnet auf der Grundlage der spezifischen Anzeigendauer und Segmentstruktur die entsprechende Segmentanforderung, die jeder Quartilposition entspricht.

Serverseitiges Tracking-Timing und Caching-Verhalten

Bei serverseitigen Berichten werden Tracking-Ereignisse auf der Grundlage von tatsächlichen Segmentanfragen des Players MediaTailor ausgelöst, nicht auf manifesten Parsing- oder Pre-Loading-Aktivitäten. Dieser Ansatz gewährleistet eine genaue Zählung der Impressionen, die den Industriestandards für die Messung von Videoanzeigen entspricht.

Wichtige Timing-Prinzipien

MediaTailor Das serverseitige Tracking folgt diesen grundlegenden Timing-Prinzipien:

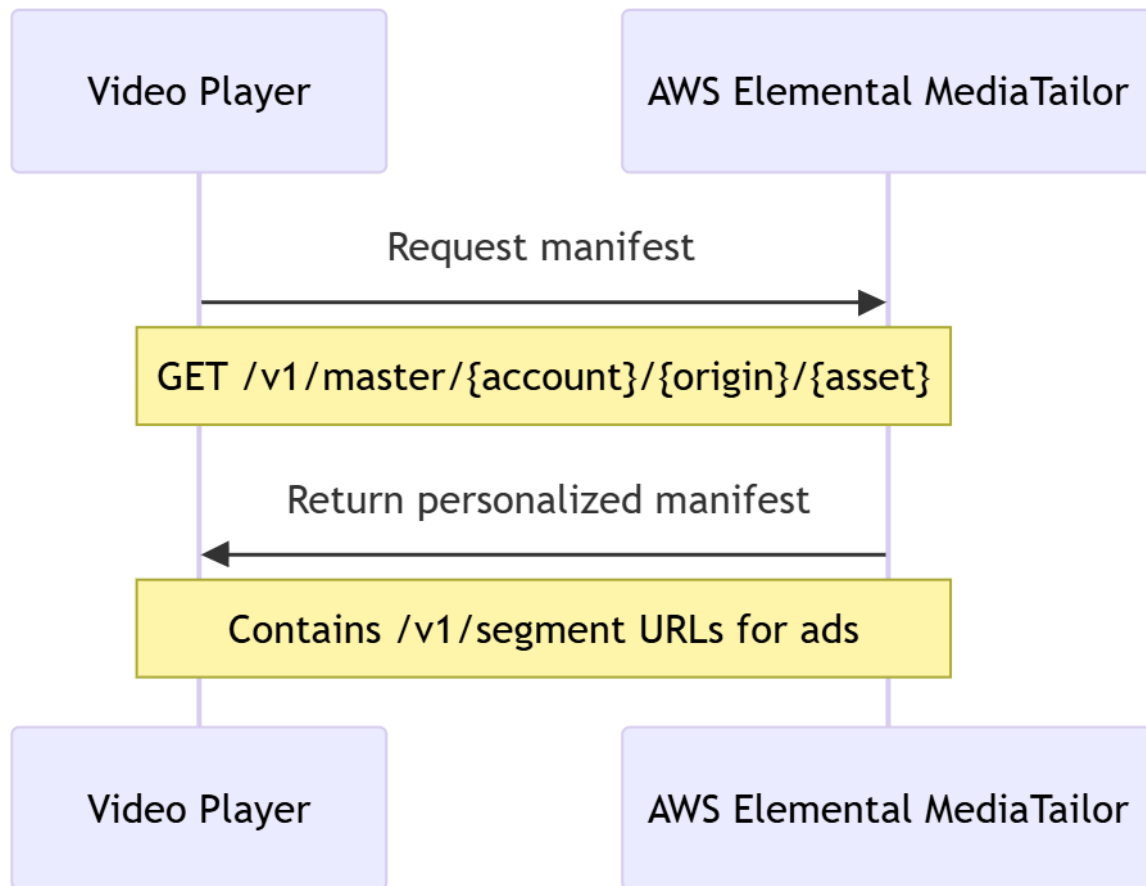
- Tracking-Ereignisse werden bei tatsächlichen Segmentanfragen ausgelöst. Beacons werden nur gesendet, wenn der Player HTTP-Anfragen an sie stellt /v1/segment URLs, nicht beim Analysieren oder Zwischenspeichern von Manifesten.
- Durch das Zwischenspeichern von Spielern und das Vorabladen von Manifesten werden KEINE Ereignisse ausgelöst. Spieler können Manifestinformationen analysieren, zwischenspeichern oder vorab laden, ohne Tracking-Ereignisse zu generieren.
- Das Vorabrufen von Segmenten löst Ereignisse aus — Wenn Spieler vor der Wiedergabe tatsächliche Anzeigensegmente vorab abrufen, entspricht dies dem branchenüblichen Verhalten, bei dem Segmentanfragen gültige Impressionen darstellen.
- Jede /v1/segment-Anfrage löst den entsprechenden Beacon aus. Das spezifische Tracking-Ereignis (Impression, Quartil, Abschluss) wird durch die Anzeigenposition und das angeforderte Segment bestimmt.
- Das Timing entspricht den IAB-Standards — Der Ansatz entspricht den Richtlinien des Interactive Advertising Bureau für die Messung von Videoanzeigen und die Zählung von Impressionen.

Workflow zur serverseitigen Nachverfolgung

Die folgenden Diagramme veranschaulichen den vollständigen serverseitigen Tracking-Workflow und zeigen, wann Tracking-Events im Zusammenhang mit Spieleranfragen ausgelöst werden:

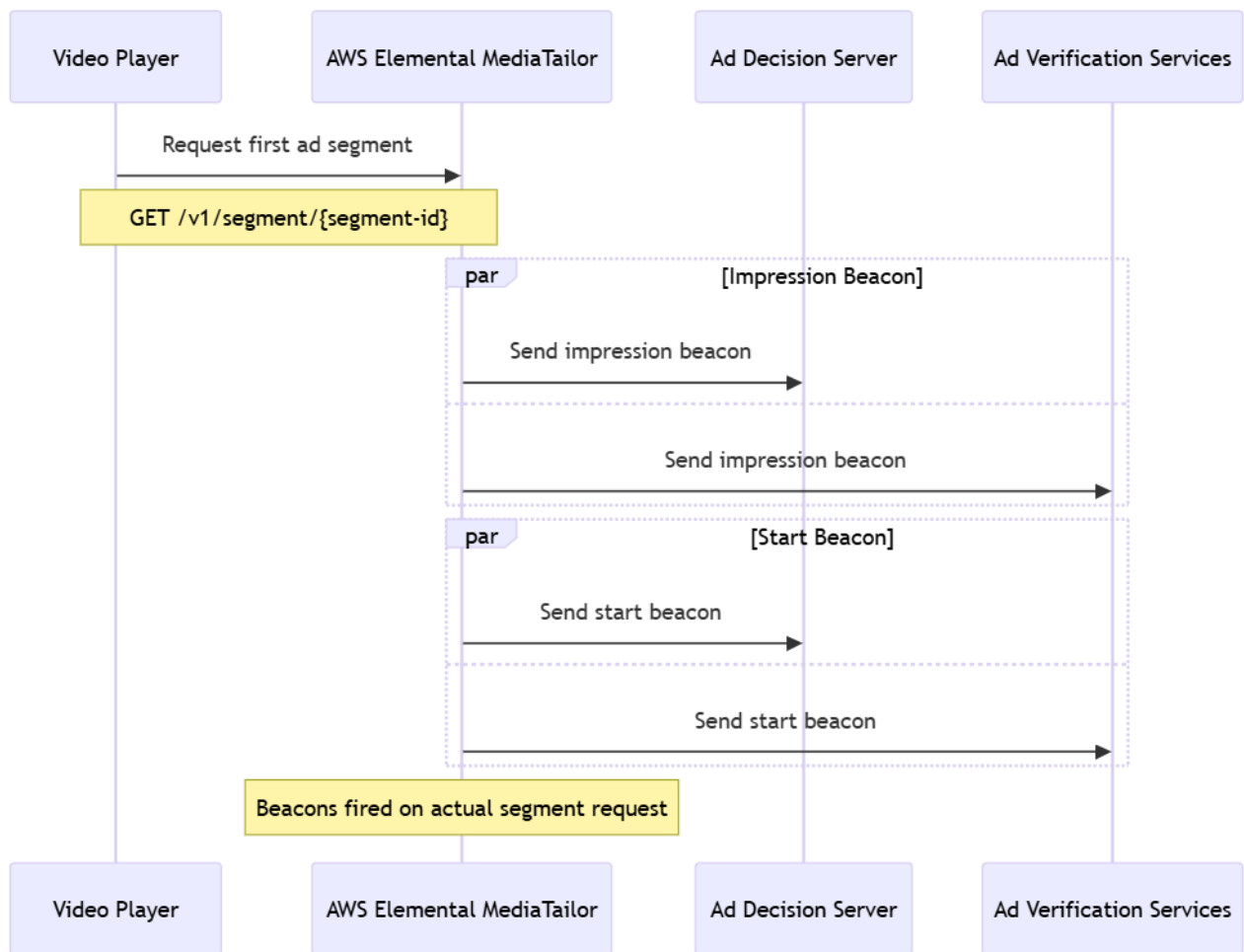
Phase 1: Initialisierung der Sitzung

Der Player fordert ein Manifest von an MediaTailor, das ein personalisiertes Manifest mit einem Anzeigensegment URLs zurückgibt:



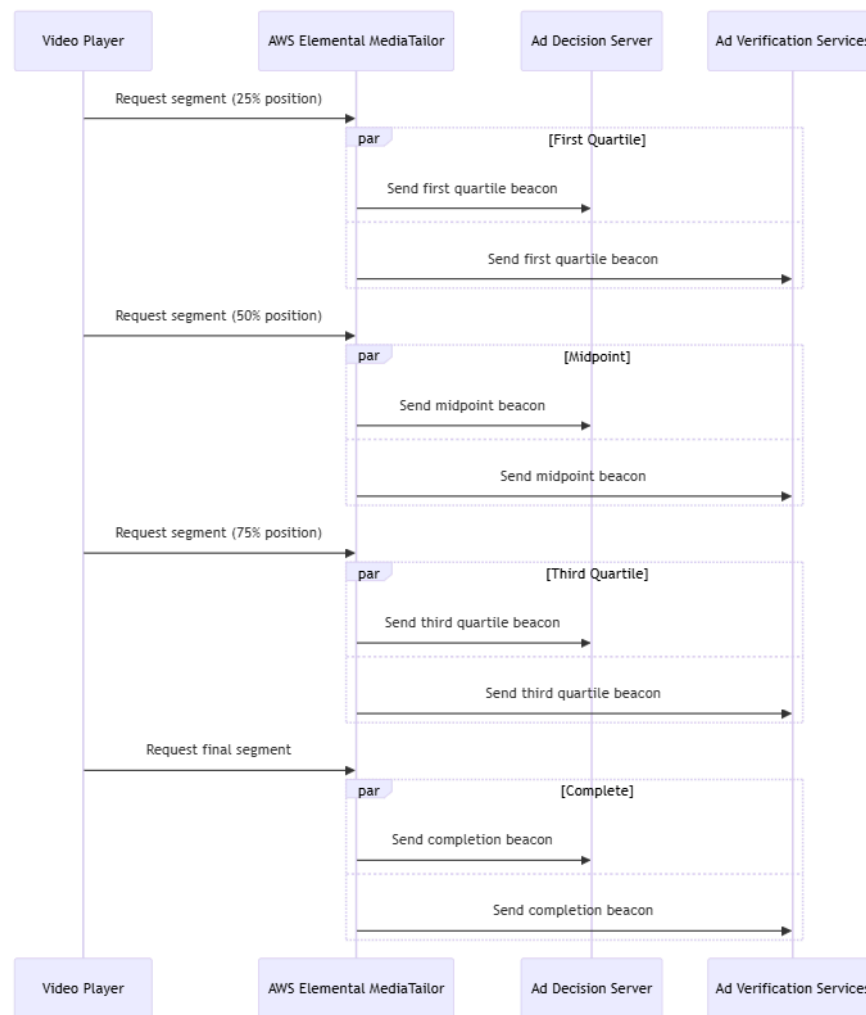
Phase 2: Anzeigenanfrage und Nachverfolgung von Impressions

Wenn der Player das erste Anzeigensegment anfordert, sendet er MediaTailor Impressions- und Start-Beacons sowohl an den Ad Decision Server als auch an die Anzeigenverifizierungsdienste:



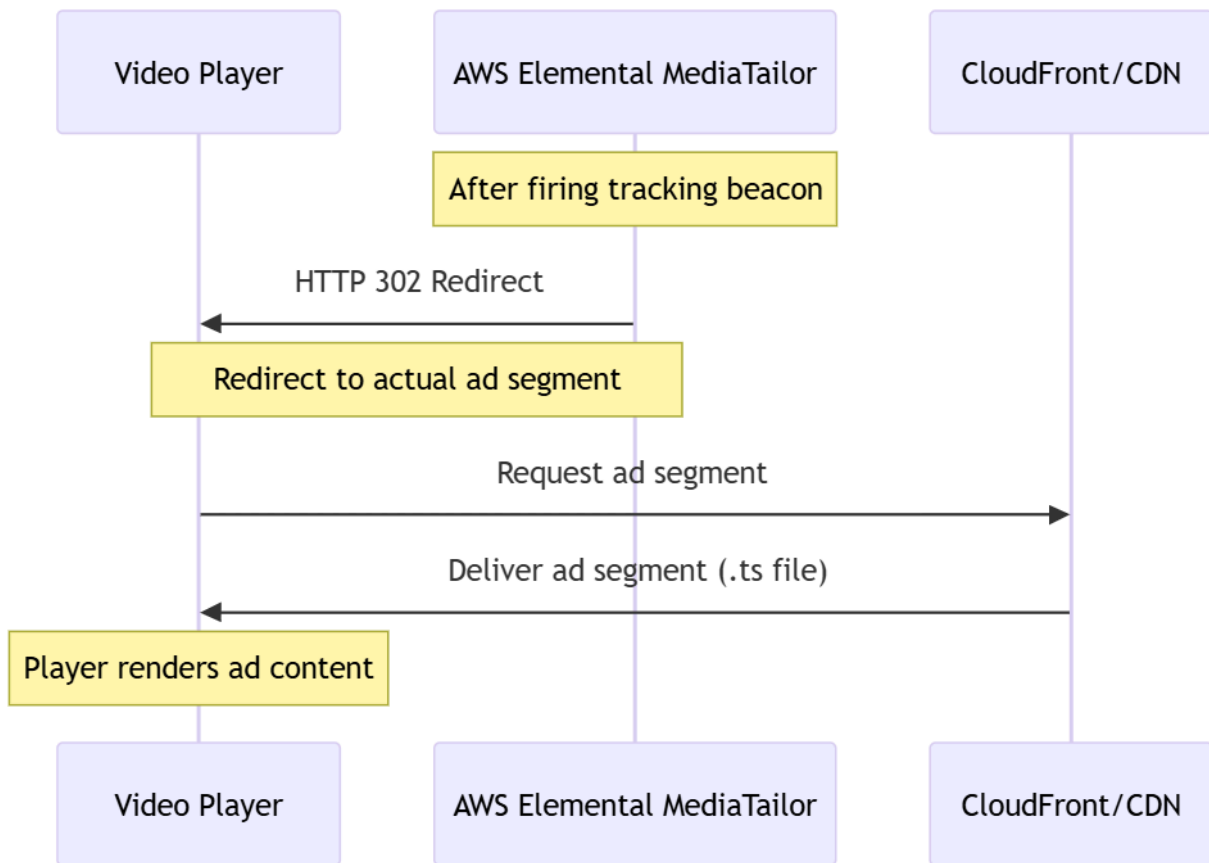
Phase 3: Verfolgung von Quartilen

MediaTailor löst Quartil-Beacons (erstes Quartil, Mittelpunkt, drittes Quartil, Abschluss) auf der Grundlage nachfolgender Segmentanfragen aus:



Phase 4: Bereitstellung in Segmenten

Nachdem Sie die Tracking-Beacons ausgelöst haben, werden Sie von Amazon CloudFront oder Ihrem CDN zum eigentlichen Anzeigensegment MediaTailor weitergeleitet:



Der serverseitige Tracking-Workflow umfasst die folgenden wichtigen Timing-Verhaltensweisen:

1. Sitzungsinitialisierung — Der Player fordert ein Manifest von an. MediaTailor MediaTailor gibt ein personalisiertes Manifest zurück, das ein Anzeigensegment URLs mit dem `/v1/segment` Pfad enthält.
2. Analysieren und Zwischenspeichern von Manifesten — Der Spieler analysiert das Manifest und kann Segmentinformationen vorab laden oder zwischenspeichern. In dieser Phase werden keine Tracking-Ereignisse ausgelöst, unabhängig vom Verhalten des Spielers beim Zwischenspeichern.
3. Anforderung von Anzeigensegmenten und Nachverfolgung von Impressionen — Wenn der Player tatsächlich das erste Anzeigensegment anfordert (normalerweise für die Wiedergabe), wird das Impression-Beacon MediaTailor ausgelöst und das Ereignis wird sowohl auf dem Ad Decision Server als auch auf den Anzeigenverifizierungsdiensten verfolgt. Dies geschieht bei der eigentlichen HTTP-Anfrage an die `/v1/segment` URL, nicht bei der Analyse des Manifests.

4. Quartil-Tracking auf der Grundlage von Segmentanfragen — MediaTailor sendet Quartil-Beacons (erstes Quartil, Mittelpunkt, drittes Quartil, Abschluss) sowohl an den Ad Decision Server als auch an die Anzeigenverifizierungsdienste auf der Grundlage von nachfolgenden Segmentanfragen, die den berechneten Quartilpositionen innerhalb der Anzeigendauer entsprechen.
5. Segmentzustellung — Führt nach dem MediaTailor Auslösen des entsprechenden Tracking-Beacons eine HTTP-Weiterleitung zum eigentlichen Anzeigensegment aus (entweder von Amazon CloudFront oder Ihrem CDN).

Überlegungen zum Zwischenspeichern von Spielern und zum Vorladen

MediaTailor Das serverseitige Tracking ist so konzipiert, dass es mit verschiedenen Strategien zum Zwischenspeichern und Vorladen von Spielern kompatibel ist und gleichzeitig eine genaue Messung der Impressionen gewährleistet:

- Vorladen von Manifesten — Spieler, die Manifestinformationen vorab laden oder zwischenspeichern, lösen keine Tracking-Ereignisse aus. Tracking-Ereignisse werden nur ausgelöst, wenn tatsächlich Segmentanfragen gestellt werden.
- Vorabruf von Segmenten — Wenn ein Player Anzeigensegmente vor der Wiedergabe vorab abrufen, werden Tracking-Events ausgelöst, wenn diese Segmente angefordert werden, möglicherweise vor der tatsächlichen Wiedergabezeit. Dieses Verhalten entspricht den Branchenstandards, die Segmentanfragen als gültige Impressionen betrachten.
- Pufferung durch Spieler — Das Standardverhalten von Playern (das Abrufen von Segmenten kurz vor der Wiedergabe) löst Tracking-Ereignisse zu den entsprechenden Zeiten aus, die auf dem Muster der Segmentanfragen basieren.

Behebung von Tracking-Diskrepanzen

Wenn Sie Diskrepanzen zwischen MediaTailor serverseitigem Tracking und Messwerten von Drittanbietern feststellen, sollten Sie die folgenden Faktoren berücksichtigen:

- Unterschiede im Spielerverhalten — Verschiedene Spieler haben möglicherweise unterschiedliche Prefetching- und Pufferstrategien, die sich darauf auswirken, wann Segmentanfragen gestellt werden.
- Netzwerkbedingungen — Schlechte Netzwerkbedingungen können dazu führen, dass Spieler Segmente mehrmals oder in anderen Intervallen als erwartet anfordern.

- **CDN-Konfiguration** — Falsches CDN-Caching von /v1/segment Anfragen kann dazu führen, dass Tracking-Ereignisse verpasst oder doppelt durchgeführt werden.
- **Sitzungsverwaltung** — Stellen Sie sicher, dass für jede Wiedergabesitzung eine eindeutige Sitzungs-ID verwendet wird, um Konflikte bei Tracking-Ereignissen zu vermeiden.

Eine ausführliche Anleitung zur Problembehebung finden Sie unter [Behebung häufiger Probleme](#).

MediaTailor Funktionen und Funktionen zur serverseitigen Nachverfolgung

AWS Elemental MediaTailor wendet diese integrierten serverseitigen Tracking-Funktionen automatisch an, um die Genauigkeit und Zuverlässigkeit der Anzeigenmessung zu optimieren. Das System verhindert doppelte Beacons, verwaltet den Datenverkehr in Zeiten mit hohem Datenvolumen, sorgt für eine korrekte Reihenfolge der Ereignisse und bietet eine umfassende Leistungsüberwachung, ohne dass Sie eine Konfiguration vornehmen müssen. Sie müssen lediglich sicherstellen, dass Ihr Ad Decision Server (ADS) die Tracking-Beacons in der VAST-Antwort bereitstellt.

Note

Diese Funktionen sind für Neukunden ab dem 30. September 2025 verfügbar. Bestandskunden werden im Rahmen der laufenden Serviceverbesserungen bis 2025 Zugriff darauf haben. Wenn Sie sofort auf diese Funktionen zugreifen möchten, wenden Sie sich an den [AWS-Support](#).

Note

Diese Funktionen gelten sowohl für zusammengefügte (SSAI) als auch für servergeführte (SGAI) Methoden zum Einfügen von Anzeigen. Die Beacon-Typen und das Timing sind in beiden Modi identisch. Sie unterscheiden sich darin, wie Beacons MediaTailor ausgelöst werden. Weitere Informationen [Serverseitiges Tracking mit serverseitiger Anzeigeneinfügung \(SGAI\)](#) zum serverseitigen SGAI-Beaconing finden Sie unter.

Beacon-Deduplizierung

MediaTailor verhindert das doppelte Auslösen von Beacons bei identischen Werbeereignissen. Das serverseitige Tracking-System sendet jede Impression, jedes Quartil und jedes Abschluss-

Beacon nur einmal pro Sitzung, in der sich die Anzeige angesehen hat. Wenn Videoplayer aufgrund von Netzwerkbedingungen, Bitratenänderungen oder Pufferstrategien mehrmals dasselbe Anzeigensegment anfordern, MediaTailor verfolgt es ausgelöste Beacons und blockiert redundante Übertragungen.

Durch die Deduplizierung werden häufig auftretende Szenarien, die zu einer erhöhten Anzahl von Beacons führen, automatisch behoben:

- **Adaptives Bitraten-Streaming** — Wenn Spieler unterschiedliche Qualitätsvarianten desselben Anzeigensegments herunterladen
- **Wiederholungsszenarien im Netzwerk** — Wenn Spieler Segmente aufgrund von Netzwerkproblemen oder Timeouts erneut anfordern
- **Pufferstrategien für Spieler** — Wenn Spieler Segmente zu Pufferzwecken vorab abrufen oder erneut abrufen

Das System ist so konzipiert, dass Impression Beacons nur einmal ausgelöst werden, selbst wenn Spieler zwischen verschiedenen Qualitätsstufen wechseln.

Adaptive Drosselung und wiederholte Beacon-Versuche

MediaTailor verwaltet automatisch die Beacon-Datenverkehrsraten auf der Grundlage von Serverantwortindikatoren. Das System überwacht HTTP-Antwortmuster, Verbindungs-Timeouts und Fehlercodes, um Überlastungen zu erkennen, und passt dann die Datenverkehrsraten entsprechend an. Wenn das System Indikatoren für die Serverauslastung identifiziert, reduziert es die Datenverkehrsraten für die betroffene Domäne und erhöht die Rate automatisch, wenn Server eine verbesserte Kapazität nachweisen.

Das System überwacht den Serverstatus anhand dieser Indikatoren:

- **Timeouts bei HTTP-Verbindungen** — Wenn Messplattformen nicht innerhalb des erwarteten Zeitrahmens reagieren
- **Fehlerantwortcodes** — 503-, 504- und 507-Antworten, die auf eine Serverüberlastung hinweisen. Ihr Anzeigenserver muss diese Fehlercodes ebenfalls unterstützen, um eine vollständige Kompatibilität zu gewährleisten.
- **Reaktionsmuster** — Leistungsänderungen der Messplattform, die auf Kapazitätsprobleme hindeuten

Beim Wiederholungsversuch wird automatisch eine Zustellung für bis zu 1 Stunde versucht, wobei zwischen den Versuchen eine Verzögerung von mindestens 30 Sekunden besteht. Dieses Wiederholungsverhalten kann nicht konfiguriert werden.

Verwaltung des Beacon-Datenverkehrs pro Sekunde

Sie können TPS-Grenzwerte festlegen, um die Übertragungsraten von Beacons zu steuern. Dies ist die einzige konfigurierbare Einstellung für serverseitige Tracking-Funktionen. Limits auf Kontoebene begrenzen die Gesamtzahl der Ad-Tracking-Anfragen, die an alle Messpartner gesendet werden. MediaTailor erzwingt ein TPS-Limit von mindestens 10.000, um ausreichend Kapazität für den Betrieb auf Unternehmensebene bereitzustellen.

Reichen Sie ein AWS Supportticket ein, um TPS-Limits mit den folgenden Informationen festzulegen:

- AWS-Konto-ID — Ihre spezifische Konto-ID
- Zielregion — Die AWS-Region, in der das TPS-Limit angewendet werden soll
- Gewünschter TPS-Schwellenwert — Ihr erforderliches Limit für Transaktionen pro Sekunde (mindestens 10.000)

Standardmäßig gibt es kein TPS-Limit. Sie können ein TPS-Limit anfordern, wenn Ihr Ad Decision Server (ADS) dies erfordert, aber das Limit muss höher als 10.000 TPS sein. MediaTailor wird das angegebene Limit nicht überschreiten, garantiert aber keinen gleichbleibenden Durchsatz bis zu diesem Limit. Ihr Anzeigenentscheidungsserver teilt Ihnen mit, welche TPS-Grenzwerte er unterstützen kann.

Beaconing in der richtigen Reihenfolge

MediaTailor sorgt automatisch für die sequentielle Bereitstellung von Ad-Tracking-Ereignissen. Das System behält die Reihenfolge der Beacons bei, auch wenn Netzwerkprobleme, Wiederholungsversuche oder Datenverkehrsmanagement auftreten. Dadurch wird sichergestellt, dass die Messpartner die Ereignisse in der richtigen Reihenfolge erhalten, sodass genaue Analysen möglich sind.

Das System folgt der branchenüblichen Beacon-Reihenfolge:

1. Ereignisse starten — Wird ausgelöst, wenn die Anzeigenwiedergabe beginnt
2. Ereignisse im ersten Quartil — werden ausgelöst, wenn die Anzeige zu 25% abgeschlossen ist
3. Ereignisse in der Mitte — Die Anzeige wird bei 50% abgeschlossener Anzeige ausgelöst

4. Ereignisse im dritten Quartil — Die Anzeige wurde zu 75% ausgelöst
5. Fertigstellungsereignisse — werden ausgelöst, wenn die Werbung beendet ist

Diese Funktionen arbeiten automatisch zusammen:

- Die Beacons werden während der Drosselung aktiviert, um die richtige Reihenfolge aufrechtzuerhalten
- Jede Domäne des Messpartners verfügt über separate Ereigniswarteschlangen, um Störungen bei Ratenanpassungen zu vermeiden
- Bei der Deduplizierung werden der Ereignistyp und die zeitliche Position unter Beibehaltung der chronologischen Reihenfolge verfolgt

Kundenseitiges Anzeigen-Tracking

Mithilfe der AWS Elemental MediaTailor clientseitigen Tracking-API können Sie Player-Steuerelemente während Werbepausen in Streaming-Workflows integrieren. Beim clientseitigen Tracking sendet der Player oder Client Tracking-Ereignisse wie Impressionen und Quartil-Ad-Beaconing an den Ad Decision Server (ADS) und andere Einheiten zur Anzeigenverifizierung. Bei diesen Ereignissen wird sowohl der Gesamtstatus der Werbeunterbrechung als auch die Anzahl der einzelnen Werbezeiträume innerhalb der einzelnen Werbeunterbrechungen erfasst. Hier finden Sie weitere Informationen zu Impression and Quartile (ADS) und anderen Einheiten zur Anzeigenverifizierung. Weitere Informationen zu Impression und Quartil-Ad-Beaconing finden Sie unter. [Clientseitiges Beaconing](#) Weitere Informationen zu ADS und anderen Entitäten zur Anzeigenverifizierung finden Sie unter. [Kundenseitige Ad-Tracking-Integrationen](#)

Informationen zur Weitergabe von Spielerparametern und Sitzungsdaten an das ADS für die clientseitige Nachverfolgung finden Sie unter und. [MediaTailor Player-Variablen für ADS-Anfragen](#)
[MediaTailor Sitzungsvariablen für ADS-Anfragen](#)

Das clientseitige Tracking ermöglicht Funktionen wie die folgenden:

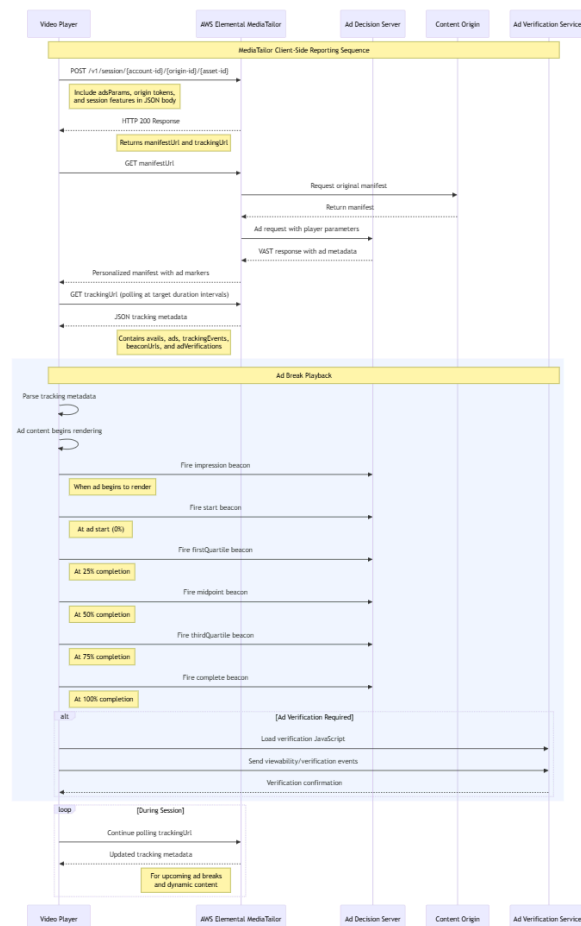
- Countdown-Timer für Werbeunterbrechungen — Weitere Informationen finden Sie unter. [Countdown-Timer hinzufügen](#)
- Durchklicken von Werbeanzeigen — Weitere Informationen finden Sie unter. [Und Click-through](#)
- Anzeige von Begleitanzeigen — Weitere Informationen finden Sie unter. [Begleitende Anzeigen](#)
- Überspringbare Anzeigen — Weitere Informationen finden Sie unter. [Überspringbare Anzeigen](#)

- Anzeige von VAST-Symbolen zur Einhaltung der Datenschutzbestimmungen — Weitere Informationen finden Sie unter [Symbole für Google Warum diese Anzeige \(WTA\)](#)
- Steuerung des Spieler-Scrubbings bei Werbeanzeigen — Weitere Informationen finden Sie unter [Schrubben](#).

Mithilfe der MediaTailor clientseitigen Tracking-API können Sie Metadaten an das Wiedergabegerät senden, die zusätzlich zum clientseitigen Tracking weitere Funktionen ermöglichen:

Workflow für die clientseitige Berichterstattung

Das folgende Diagramm zeigt den gesamten Workflow für die clientseitige Berichterstattung von der Sitzungsinitialisierung über die Anzeigenwiedergabe bis hin zum Beaconing:



Der clientseitige Berichtsworkflow umfasst die folgenden Schritte:

1. Sitzungsinitialisierung — Der Videoplayer sendet eine POST-Anfrage mit JSON-Metadaten `adParams`, einschließlich Ursprungs-Tokens und MediaTailor Sitzungsfunktionen, an den Sitzungsendpunkt. MediaTailor reagiert mit `manifestUrl` und `trackingUrl` für die Sitzung.

2. Manifestanfrage und Anzeigenentscheidung — Der Spieler fordert das personalisierte Manifest von an MediaTailor. MediaTailor fordert das ursprüngliche Inhaltsmanifest vom Absender an, sendet mithilfe von Player-Parametern eine Anzeigenanfrage an den Ad Decision Server (ADS), erhält eine VAST-Antwort mit Anzeigenmetadaten und übermittelt dem Player ein personalisiertes Manifest mit Werbemarkierungen.
3. Abruf von Tracking-Daten — Der Player fragt die Tracking-URL in regelmäßigen Abständen ab (entspricht der Zieldauer für HLS oder dem Mindestaktualisierungszeitraum für DASH). MediaTailor gibt JSON-Tracking-Metadaten zurück, die Verfügbarkeiten, Anzeigen, Tracking-Ereignisse, Beacon URLs - und Anzeigenverifizierungsdaten enthalten.
4. Anzeigenwiedergabe und Beacons — Während Werbeunterbrechungen analysiert der Player die Tracking-Metadaten, feuert Impression-Beacons ab, wenn die Werbung gerendert wird, löst zum richtigen Zeitpunkt Quartil-Beacons (Start, FirstQuartile, Midpoint, ThirdQuartile, complete) aus, lädt und führt bei Bedarf eine Anzeigenüberprüfung durch und sendet Ereignisse an Bestätigungsdienste von Drittanbietern. JavaScript viewability/verification
5. Kontinuierliche Abfrage — Der Player fragt die Tracking-URL während der gesamten Sitzung weiter ab, um aktualisierte Metadaten für bevorstehende Werbeunterbrechungen und dynamische Inhalte zu erhalten.

Dieser Workflow ermöglicht erweiterte Funktionen wie Countdown-Timer für Werbeanzeigen, Click-Through-Funktionen, Begleitanzeigen, überspringbare Anzeigen und die Anzeige von VAST-Symbolen zur Einhaltung der Datenschutzbestimmungen.

Themen

- [Aktivierung von clientseitigem Tracking](#)
- [Serverparameter hinzufügen](#)
- [Abfrageparameter für Origin-Interaktionen](#)
- [Für die Sitzung konfigurierte Funktionen](#)
- [Bewährte Methoden für die clientseitige Nachverfolgung](#)
- [Schema und Eigenschaften für das clientseitige Ad-Tracking](#)
- [Zeitlicher Ablauf der Aktivitäten bei der Anzeigenverfolgung](#)
- [Player-Bedienelemente und Funktionen für das clientseitige Ad-Tracking](#)
- [Clientseitiges Beacons](#)
- [Hybridmodus mit serverseitigen Ad Beacons](#)

- [Kundenseitige Ad-Tracking-Integrationen](#)
- [Durch Werbe-Beacons blättern mit GetTracking](#)

Aktivierung von clientseitigem Tracking

Sie aktivieren das clientseitige Tracking für jede Sitzung. Der Player sendet ein HTTP POST an den Endpunkt des MediaTailor Sitzungsinitialisierungspräfixes der Konfiguration. Optional kann der Player zusätzliche Metadaten senden, die dann verwendet werden können, wenn er Werbeaufrufe tätigt, den Ursprung für ein Manifest aufruft und MediaTailor Funktionen auf Sitzungsebene aufruft oder deaktiviert. MediaTailor

Das folgende Beispiel zeigt die Struktur der JSON-Metadaten:

```
{
  "adsParams": {
    "param1": "value1",
    "param2": "value2",
    "origin_access_token": "abc123",
    "overlayAvails": "on"
  },
  "origin_access_token": "abc123",
  "overlayAvails": "on"
}
```

'adsParams' is case sensitive
key is not case sensitive
Values can contain spaces. For example, 'value 2' is an allowed value.
this is an example of a query parameter designated for the origin
'overlayAvails' is case sensitive. This is an example of a feature that is enabled at the session level.

Verwenden Sie die MediaTailor Konsole oder API, um die URL der ADS-Anforderungsvorlage so zu konfigurieren, dass sie auf diese Parameter verweist. Im folgenden Beispiel `player_params.param1` sind die Player-Parameter für `param1` und `player_params.param2` die Player-Parameter für `param2`.

```
https://my.ads.com/path?param1=[player_params.param1]&param2=[player_params.param2]
```

Serverparameter hinzufügen

Auf der obersten Ebene der JSON-Struktur befindet sich ein JSON-Objekt. `adsParams` In diesem Objekt befinden sich key/value Paare, die in allen Sitzungsanfragen lesen und an den Ad-Server senden MediaTailor können. MediaTailor unterstützt die folgenden Anzeigenserver:

- Google Ad Manager

- SpringServe
- FreeWheel
- Öffentlich

Abfrageparameter für Origin-Interaktionen

Alle reservierten key/value Paare innerhalb der obersten Ebene der JSON-Struktur, wie, und adsParams availSuppressionoverlayAvails, werden der ursprünglichen Anforderungs-URL nicht in Form von Abfrageparametern hinzugefügt. Jede Sitzungsmanifestanforderung, die an MediaTailor den Ursprung gerichtet wird, enthält diese Abfrageparameter. Der Ursprung ignoriert überflüssige Abfrageparameter. MediaTailor Kann die key/value Paare beispielsweise verwenden, um Zugriffstoken an den Ursprung zu senden.

Für die Sitzung konfigurierte Funktionen

Verwenden Sie die JSON-Struktur für die Sitzungsinitialisierung, um Funktionen wie, und zu aktivieren, zu deaktivieren oder zu überschreiben MediaTailor. overlayAvails availSuppression adSignaling Alle während der Sitzungsinitialisierung übergebenen Funktionskonfigurationen überschreiben die Einstellung auf der Konfigurationsebene. MediaTailor

Note

Die MediaTailor bei der Sitzungsinitialisierung übermittelten Metadaten sind unveränderlich, und zusätzliche Metadaten können für die Dauer der Sitzung nicht hinzugefügt werden. Verwenden Sie SCTE-35-Markierungen, um Daten zu übertragen, die sich während der Sitzung ändern. Weitere Informationen finden Sie unter [MediaTailor Sitzungsvariablen für ADS-Anfragen](#).

Example: Durchführung von kundenseitigem Ad-Tracking für HLS

```
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: Durchführung von kundenseitigem Ad-Tracking für DASH

```
POST mediatailorURL/v1/session/hashed-account-id/origin-id/asset-id.mpd
```

```
{
  "adsParams": {
    "deviceType": "androidmobile",
    "uid": "xjhddli-9189901-uic"
  }
}
```

Parameter für den Berichtsmodus

Sie können den Berichtsmodus bei der Initialisierung einer Sitzung angeben, indem Sie den `reportingMode` Parameter in den Anfragetext aufnehmen. Dieser Parameter steuert, ob für die MediaTailor Sitzung clientseitiges oder serverseitiges Ad-Tracking durchgeführt wird.

- `client`- Der Player führt Ad-Tracking durch und sendet Beacons an den Anzeigenserver. Dies ist der Standardmodus, wenn kein Modus angegeben `reportingMode` ist.
- `server`- MediaTailor führt serverseitiges Ad-Tracking durch und sendet Beacons direkt an den Anzeigenserver.

Example Sitzungsinitialisierung mit serverseitigem Berichtsmodus

```
POST mediatailorURL/v1/session/hashed-account-id/origin-id/asset-id.m3u8
```

```
{
  "adsParams": {
    "deviceType": "ipad",
    "uid": "abdgfdyei-2283004-ueu"
  },
  "reportingMode": "server"
}
```

Example Sitzungsinitialisierung mit clientseitigem Berichtsmodus (explizit)

```
POST mediatailorURL/v1/session/hashed-account-id/origin-id/asset-id.mpd
```

```
{
  "adsParams": {
    "deviceType": "androidmobile",
    "uid": "xjhddli-9189901-uic"
  },
  "reportingMode": "client"
}
```

Note

Der `reportingMode` Parameter wird bei der Sitzungsinitialisierung festgelegt und kann während der Sitzung nicht geändert werden. Wenn kein Wert angegeben `reportingMode` ist, wird MediaTailor standardmäßig die clientseitige Berichterstattung verwendet, um die Abwärtskompatibilität zu gewährleisten.

Eine erfolgreiche Antwort ist ein HTTP 200 mit einem Antworttext. Der Hauptteil enthält ein JSON-Objekt mit einem `manifestUrl` und einem `trackingUrl` Schlüssel. Bei den Werten handelt es sich um relative Werte URLs , die der Player sowohl für die Wiedergabe als auch für die Nachverfolgung von Werbeereignissen verwenden kann.

```
{
  "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"
}
```

Weitere Informationen zum clientseitigen Tracking-Schema finden Sie unter. [Schema und Eigenschaften für das clientseitige Ad-Tracking](#)

Bewährte Methoden für die clientseitige Nachverfolgung

In diesem Abschnitt werden die bewährten Methoden für das clientseitige Tracking sowohl für Live- als auch MediaTailor für VOD-Workflows beschrieben.

Live-Workflows

Fragen Sie den Tracking-Endpunkt in einem Intervall ab, das jeder Zieldauer für HLS oder dem minimalen Aktualisierungszeitraum für DASH entspricht, um immer über die aktuellsten Ad-Tracking-

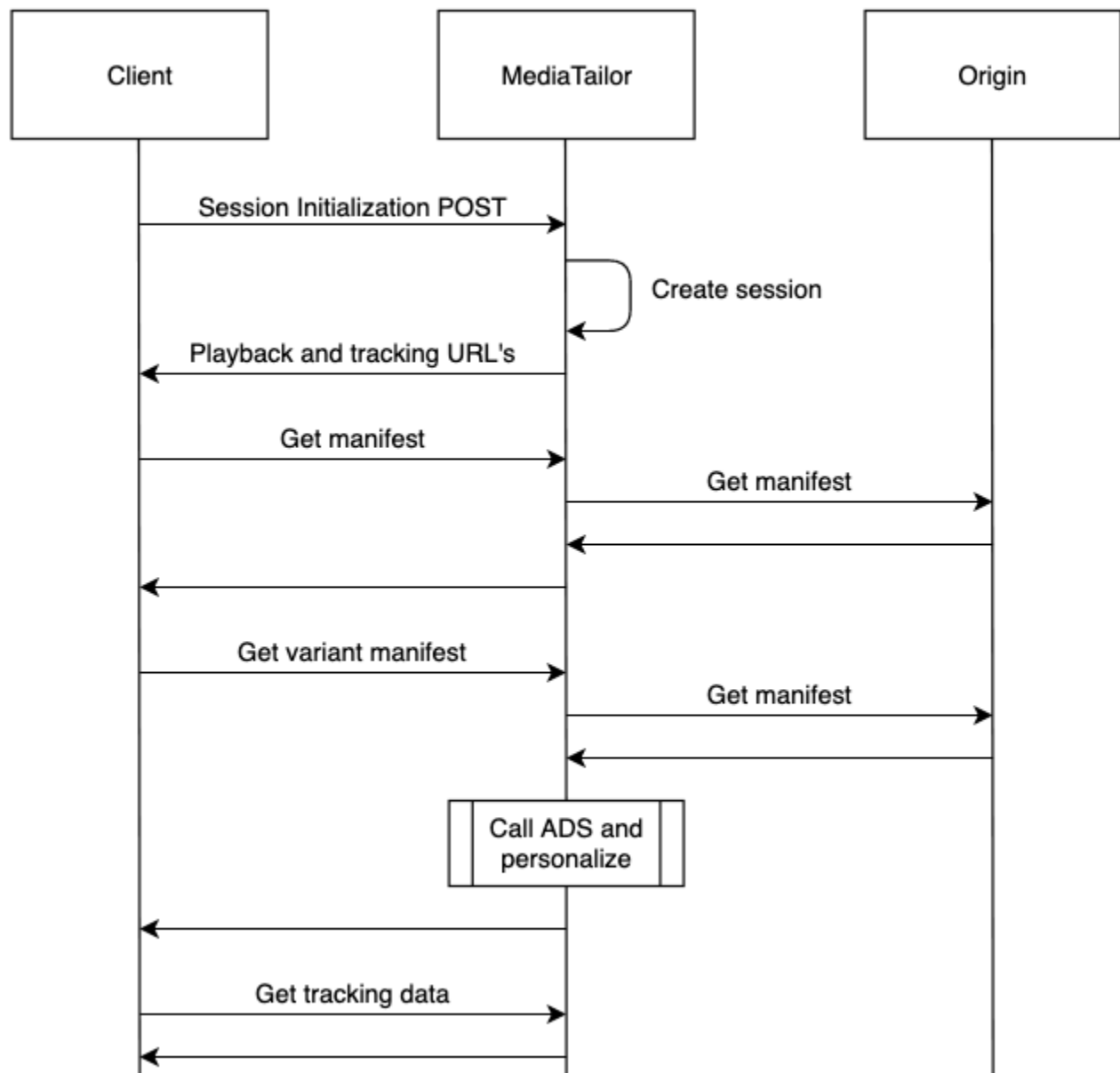
Metadaten zu verfügen. Die Anpassung dieses Intervalls ist besonders wichtig in Workflows, bei denen die Kreativen möglicherweise über eine interaktive Komponente oder eine Overlay-Komponente verfügen.

 Note

Einige Player unterstützen Event-Listener, die als Alternative zu Umfragen verwendet werden könnten. Beispielsweise müsste die Funktion zur Dekoration von MediaTailor Werbe-IDs für jede Sitzung aktiviert werden. Weitere Informationen finden Sie unter [Werbe-ID-Dekoration](#). Wenn Sie diese Funktion verwenden, wird für jede verfügbare Anzeige ein Datumsbereich (HLS) oder ein Eventelement (DASH) -Identifizier verwendet. Spieler können diese Manifest-Tags als Aufforderung verwenden, um den MediaTailor Tracking-Endpunkt für die Sitzung aufzurufen.

VOD-Workflows

Nach einer erfolgreichen Sitzungsinitialisierung und nach Erhalt des ersten MediaTailor Manifests, das Medien enthält, müssen Sie den Tracking-Endpunkt nur einmal aufrufen.



Servergesteuerte Anzeigeneinfügung

Servergesteuerte Ad-Insertion-Sitzungen (SGAI) verwenden die API nicht. `GetTracking` MediaTailor stellt stattdessen bei der Verwendung `aws.reportingMode=CLIENT` von Tracking-Informationen im TRACKING Abschnitt jeder Antwort auf die Asset-Liste bereit, wenn Spieler Werbeeinhalte anfordern. Die Antwort auf die Initialisierung der Sitzung enthält kein `trackingUrl`.

Die Antwort auf die Assetliste für clientseitig verfolgte SGAI-Sitzungen hat die folgende Struktur:

```
{
```

```
"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...
}
}
```

Bei der Implementierung von clientseitigem Tracking für SGAI-Methoden:

- Analysieren Sie den TRACKING Abschnitt anhand der Antworten auf die Asset-Liste, anstatt ihn aufzurufen `GetTracking`
- Verwenden Sie das in der Asset-Liste URLs bereitgestellte Tracking für die Berichterstattung über Anzeigenereignisse
- Feuern Sie Tracking-Beacons ab, die auf tatsächlichen Ereignissen bei der Anzeigenwiedergabe im Player basieren
- Verwalte das Tracking für jede Werbeunterbrechung unabhängig, wenn die Asset-Listen abgerufen werden

Important

Der TRACKING Abschnitt ist nur dann in der Asset-Liste enthalten, wenn er festgelegt `aws.reportingMode=CLIENT` ist. Wenn serverseitiges Reporting verwendet wird (Standard für SGAI), wird der TRACKING Abschnitt MediaTailor weggelassen und stattdessen Beacon-Daten in die Anzeige eingebettet. URIs Details hierzu finden Sie unter [Serverseitiges Tracking mit serverseitiger Anzeigeneinfügung \(SGAI\)](#).

Schema und Eigenschaften für das clientseitige Ad-Tracking

Mit der MediaTailor clientseitigen Ad-Tracking-Funktion können Sie detaillierte clientseitige Ad-Tracking-Daten in Ihre Player-Umgebung integrieren. In den folgenden Abschnitten werden das allgemeine Ad-Tracking-Schema sowie die spezifischen Eigenschaften und Werte behandelt, aus denen sich das Schema zusammensetzt.

Schema

Die folgende JSON-Struktur zeigt das MediaTailor clientseitige Ad-Tracking-Schema. Diese Darstellung veranschaulicht die Verschachtelungsstruktur des Schemas, um Ihnen das Verständnis der Beziehungen zwischen den verschiedenen Teilen zu erleichtern.

Ausführliche Informationen zu den einzelnen Eigenschaften finden Sie unter [the section called “Eigenschaften”](#).

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

Eigenschaften

In der folgenden Tabelle sind die Eigenschaften in der clientseitigen Tracking-API, ihre Definitionen, Wertetypen und Beispiele aufgeführt.

Eigenschaft	Definition	Werttyp	Beispiel
adID	- HLS — die Sequenznummer, die dem Anfang der Anzeige zugeordnet ist - DASH — die Perioden-ID der Anzeige Pfad: /avails/ads/adId VAST-Zuordnung: Keine	Zeichenfolge	10
adBreakTrackingEvents	Ein Array, das VMAP-Tracking-Ereignisse aus der VAST-Antwort enthält. Weitere Informationen finden Sie in Abschnitt 2.3.3 der VMAP 1.0-Spezifikation. Pfad: /avails/ads/adBreakTrackingEvents	Array	[]

Eigenschaft	Definition	Werttyp	Beispiel
<code>adMarkerDuration</code>	Die anhand der Anzeigenmarkierung im Manifest beobachtete Nutzungsdauer.	Zeichenfolge	30
<code>adParameters</code>	Eine Reihe von Anzeigenparametern aus dem VAST VPAID, die an den MediaTailor Player weitergegeben wird. Pfad: <code>/avails/ads/adParameters</code> VAST-Zuordnung: <code>VAST/Ad/Inline/Creatives/Creative/Linear/AdParameters</code>	Zeichenfolge	
<code>adProgramDateTime</code>	- HLS — das Datum im Format ISO/IEC 8601:2004, das die erste Mediensequenz der Anzeige darstellt. - DASH — [Nicht spezifiziert]	Zeichenfolge	
<code>ads</code>	Ein Array, das die Werbeobjekte enthält, aus denen sich der Avail zusammensetzt. Die Anzeigen werden in der Reihenfolge aufgeführt, in der sie im Manifest erscheinen. Pfad: <code>/avails/ads</code>	Array	[]

Eigenschaft	Definition	Werttyp	Beispiel
adSystem	Der Name des Systems, das die Anzeige bereitstellt.  Important Stellen Sie sicher, dass Sie einen Wert angeben. Wenn Sie keinen Wert angeben, können Probleme auftreten.	Zeichenfolge	myADS
adTitle	Der Titel der Anzeige.	Zeichenfolge	ad1
adVerifications	Enthält die Ressourcen und Metadaten, die für die Ausführung von Messcode von Drittanbietern erforderlich sind, um die kreative Wiedergabe zu überprüfen. Weitere Informationen zu dieser Eigenschaft finden Sie in Abschnitt 3.16 der VAST 4.2-Spezifikation. MediaTailor unterstützt adVerifications als VAST 3 Erweiterungsknoten. Pfad: /avails/ads/adVerifications VAST-Zuordnung: VAST/Ad/InLine/AdVerifications	Array	[]

Eigenschaft	Definition	Werttyp	Beispiel
<code>altText</code>	Der alternative Text für ein Bild einer Begleitanzeige. Dieser Text ermöglicht es Spielern mit beschreibender Audiounterstützung für sehbehinderte Personen, eine Beschreibung des Bildes vorzulesen. Pfad: <code>/avails/ads/companionAds/altText</code>	Zeichenfolge	<pre>video sequence advertising sneakers</pre>
<code>apiFramework</code>	Stellen Sie auf ein VPAID, um dem Player mitzuteilen, dass es sich bei dieser Anzeige um eine VPAID-Anzeige handelt. Kann an mehreren Stellen im Schema erscheinen.	Zeichenfolge	VPAID
<code>availID</code>	- HLS — die Sequenznummer, die mit dem Start der verfügbaren Anzeige verknüpft ist. - DASH — die Perioden-ID der Werbeanzeige, bei der es sich in der Regel um die Perioden-ID des Inhalts handelt, der durch eine Anzeige ersetzt werden soll. Pfad: <code>/avails/availID</code>	Zeichenfolge	34 - PT34S_1

Eigenschaft	Definition	Werttyp	Beispiel
<code>avails</code>	Ein Array, das Ad-Break-Objekte oder Avails enthält, die im aktiven Manifestfenster angezeigt werden. Die Avails werden in der Reihenfolge aufgeführt, in der sie im Manifest erscheinen. Pfad: <code>/avails</code>	Array	<code>[]</code>
<code>adType</code>	Der Typ der Anzeige. Pfad: <code>/avails/adType</code> und <code>/avails/ads/adType</code>	Zeichenfolge	
<code>dateTime</code>	Programmdatum und Uhrzeit im Format ISO 8601 Sekunden für den Start der Anzeige Avail oder Anzeige. Pfad: und <code>/avails/dateTime</code> / <code>avails/ads/dateTime</code>	Zeichenfolge	
<code>duration</code>	Länge im ISO-8601-Sekunden-Format. Die Antwort umfasst die Dauer für die gesamte Anzeigenvorfugbarkeit sowie für jede Anzeige und jedes Beacon, obwohl die Dauer der Beacons immer Null ist. Pfad: und <code>/avails/duration</code> / <code>avails/ads/duration</code>	Zeichenfolge	<code>15.015</code>
<code>durationInSeconds</code>	Länge im Sekundenformat. Pfad: <code>/avails/durationInSeconds</code> und <code>/avails/ads/durationInSeconds</code>	Zahl	

Eigenschaft	Definition	Werttyp	Beispiel
extensions	Benutzerdefinierte Erweiterungen von VAST, die Werbeserver verwenden. Weitere Informationen zu Erweiterungen finden Sie in Abschnitt 3.18 der VAST 4.2-Spezifikation. Pfad: /avails/ads/extensions VAST-Zuordnung: VAST/Ad/Inline/Extensions	Array	[]
icons	Symbolelemente für die Anzeige. Pfad: /avails/ads/icons VAST-Zuordnung: VAST/Ad/Inline/Creatives/Creative/Linear/Icons	Array	
mediaFiles	Video und andere Inhalte, die der Spieler für die Anzeige benötigt, stehen zur Verfügung. Pfad: /avails/ads/mediaFiles	Objekt	
nonLinearAvails	Eine Reihe nichtlinearer und verfügbarer Objekte. Pfad: /nonLinearAvails	Array	

Eigenschaft	Definition	Werttyp	Beispiel
<code>executableResource</code>	Ausführbare Ressourcen zur Überprüfung. Pfad: <code>/avails/ads/adVerifications/executableResource</code> VAST-Zuordnung: <code>VAST/Ad/InLine/AdVerifications/Verification/ExecutableResource</code>	Array	
<code>javascriptResource</code>	JavaScript Ressourcen zur Überprüfung. Pfad: <code>/avails/ads/adVerifications/javascriptResource</code> VAST-Zuordnung: <code>VAST/Ad/InLine/AdVerifications/Verification/JavaScriptResource</code>	Array	
<code>trackingEvents</code>	Erfassung von Ereignissen zur Verifizierung oder als Begleitwerbung Pfad: <code>/avails/ads/adVerifications/trackingEvents</code> oder <code>/avails/ads/companionAds/trackingEvents</code>	Array	

Eigenschaft	Definition	Werttyp	Beispiel
vendor	Anbieter der Verifizierung. Pfad: /avails/ads/adVerifications/vendor VAST-Zuordnung: VAST/Ad/InLine/AdVerifications/Verification/@vendor	Zeichenfolge	
uri	URI, die entweder auf ein ausführbares Asset, ein Video-Asset oder einen Tracking-Endpunkt verweist. Pfad: Verschiedene Speicherorte im Schema VAST-Zuordnung: Verschiedene CDATA-Elemente in VAST	Zeichenfolge	https://tracking.example.com/impression
verificationParameters	Überprüfungsparameter. Pfad: /avails/ads/adVerifications/verificationParameters VAST-Zuordnung: VAST/Ad/InLine/AdVerifications/Verification/VerificationParameters	Zeichenfolge	
attributes	Begleitende Anzeigenattribute wie Abmessungen und Rendermodus. Pfad: /avails/ads/companionAds/attributes	Objekt	

Eigenschaft	Definition	Werttyp	Beispiel
<code>companionClickThrough</code>	Eine URL zur Seite des Werbetreibenden, die der Media Player öffnet, wenn der Zuschauer auf die Begleitanzeige klickt. Pfad: /avails/ads/companionAds/companionClickThrough VAST-Zuordnung: VAST/Ad/Inline/Creatives/Creative/CompanionAds/Companion/CompanionClickThrough	Zeichenfolge	<code>https://aws.amazon.com/</code>
<code>companionClickTracking</code>	Die Tracking-URL für die companion ClickThrough Immobilie. Pfad: /avails/ads/companionAds/companionClickTracking VAST-Zuordnung: VAST/Ad/Inline/Creatives/Creative/CompanionAds/Companion/CompanionClickTracking	Zeichenfolge	<code>https://myads.com/beaconing/event=clicktracking</code>

Eigenschaft	Definition	Werttyp	Beispiel
htmlResource	Der CDATA-kodierte HTML-Code, der direkt in die HTML-Seite des Streaming-Anbieters eingefügt wird. Pfad: /avails/ads/companionAds/htmlResource VAST-Zuordnung: VAST/Ad/Inline/Creatives/Creative/CompanionAds/Companion/HTMLResource	Zeichenfolge	<pre><![CDATA[<!doctype html><htm l><head>< meta name= \"viewport \" content= \"width=1, initial-s cale=1.0, minimum-s cale=1.0, ...]]></pre>
iFrameResource	Die URL zu einer HTML-Ressourcendatei, die der Streaming-Anbieter in einen Iframe lädt. Pfad: /avails/ads/companionAds/iFrameResource VAST-Zuordnung: VAST/Ad/Inline/Creatives/Creative/CompanionAds/Companion/iFrameResource	Zeichenfolge	
sequence	Der Sequenzwert, der für das Creative in der VAST-Antwort angegeben wurde. Pfad: /avails/ads/companionAds/sequence	Zeichenfolge	1

Eigenschaft	Definition	Werttyp	Beispiel
<code>startTime</code>	Die Zeitposition im ISO-8601-Sekundenformat. Bei HLS bezieht sich dies auf den Beginn der Wiedergabesitzung. Für DASH ist dies relativ zur AST (Availability Start Time) des Manifests. Die Antwort enthält Startzeiten für den gesamten Ad-Avail und für jede Werbung und jedes Beacon. Pfad: <code>/avails/startTime</code> und <code>/avails/ads/startTime</code>	Zeichenfolge	PT18.581355S
<code>startTimeInSeconds</code>	Die Zeitposition im Sekundenformat. Bei HLS bezieht sich dies auf den Beginn der Wiedergabesitzung. Für DASH ist dies relativ zur AST (Availability Start Time) des Manifests. Die Antwort enthält Startzeiten für den gesamten Ad-Avail und für jede Werbung und jedes Beacon. Pfad: <code>/avails/startTimeInSeconds</code> und <code>/avails/ads/startTimeInSeconds</code>	Zahl	18.581
<code>eventId</code>	- HLS — die dem Beacon zugeordnete Sequenznummer. - DASH — der <code>ptsTime</code> Zeitpunkt des Starts der Anzeige.	Zeichenfolge	

Eigenschaft	Definition	Werttyp	Beispiel
event	Der Name des Tracking-Ereignisses. Pfad: /avails/ads/adVerifications/trackingEvents/event oder /avails/ads/companionAds/trackingEvents/event	Zeichenfolge	impression , start, firstQuartile , midpoint, thirdQuartile , complete
beaconUrls	Die URL, an die der Ad Beacon MediaTailor gesendet wird. Pfad: /avails/ads/mediaFiles/trackingEvents/beaconUrls	Array	
bitrate	Die Bitrate des Video-Assets. Diese Eigenschaft ist in der Regel nicht für ein ausführbares Asset enthalten.	Zeichenfolge	2048
companion Ads	Eine oder mehrere Spezifikationen für begleitende Anzeigeninhalte, von denen jede eine zu verwendende Ressourcendatei spezifiziert. Begleitanzeigen begleiten die Werbeanzeige und bieten Inhalte, wie z. B. einen Rahmen um die Anzeige oder ein Banner, der in der Nähe des Videos eingeblendet wird. Pfad: /avails/ads/companionAds	Array	[]
creativeId	Der Id Attributwert des Creative Tags für die Anzeige.	Zeichenfolge	creative-1

Eigenschaft	Definition	Werttyp	Beispiel
<code>creativeSequence</code>	Die Reihenfolge, in der eine Anzeige abgespielt werden soll, entsprechend dem Ad@id Wert in der VAST-Antwort.	Zeichenfolge	1
<code>dashAvailabilityStartTime</code>	Bei live/dynamic DASH ist es das MPD@availabilityStartTime Ursprungs-Manifest.	Zeichenfolge	2022-10-05T19:38:39.263Z
<code>delivery</code>	Gibt an, ob ein progressive streaming OR-Protokoll verwendet wird.	Zeichenfolge	progressive
<code>eventType</code>	Der Typ des Beacons. Pfad: /avails/ads/mediaFiles/trackingEvents/eventType	Zeichenfolge	impression
<code>height</code>	Die Höhe des Video-Assets in Pixeln.	Zeichenfolge	360
<code>hlsAnchorMediaSequenceNumber</code>	Die Mediensequenznummer der first/oldest Mediensequenz, die im HLS-Ursprungsmanifest zu sehen ist.	Zeichenfolge	77
<code>maintainAspectRatio</code>	Gibt an, ob das Seitenverhältnis des Videos bei der Skalierung beibehalten werden soll.	Boolesch	true
<code>mediaFileList</code>	Gibt das Video und andere Elemente an, die der Player für die Anzeigenbereitstellung benötigt. Pfad: /avails/ads/mediaFiles/mediaFilesList	Array	[]

Eigenschaft	Definition	Werttyp	Beispiel
<code>mediaFileUri</code>	URI, die entweder auf ein ausführbares Asset oder ein Video-Asset verweist.	Zeichenfolge	<code>https://myad.com/ad/ad134/vpaid.js</code>
<code>mediaType</code>	Der MIME-Typ des kreativen Elements oder des Begleitelements.	Zeichenfolge	<code>video/mp4</code>
<code>meta</code>	Zusätzliche Metadaten für die Anzeige.	Objekt	
<code>mezzanine</code>	Die URL des MP4 Mezzanine-Assets, angegeben, falls die VPAID-Anzeige eines enthält. Pfad: <code>/avails/ads/mediaFiles/mezzanine</code>	Zeichenfolge	<code>https://gcdn.2mdn.net/video/playback/id/itag/c/k2/file/file.mp4</code>
<code>nextToken</code>	Der Wert des Tokens, das auf die nächste Ergebnisseite verweist, sofern ein solcher Wert vorhanden ist.	Zeichenfolge	<code>UFQz0S44NzNTXzIwMjMtMDctMzFUMTY6NTA6MDYuMzUwNjI2ODQ1Wl8x</code>
<code>nonLinearAds</code>	Nichtlineare Anzeigen, die neben dem Videoinhalt erscheinen.	Array	<code>[]</code>
<code>nonLinearAdsList</code>	Liste der nichtlinearen Anzeigen.	Array	<code>[]</code>
<code>scalable</code>	Gibt an, ob das Video auf andere Dimensionen skaliert werden soll.	Boolesch	<code>true</code>

Eigenschaft	Definition	Werttyp	Beispiel
skipOffset	Der Zeitwert, der angibt, wann der Player dem Benutzer Steuerelemente zum Überspringen zur Verfügung stellt.	Zeichenfolge	00:00:05
staticResource	Die URL zu einer statischen kreativen Datei, die für die Anzeigenkomponente verwendet wird. Pfad: /avails/ads/companionAds/staticResource	Zeichenfolge	https://very-interactive-ads.com/campaign1/file.json?c=1019113602
vastAdId	Der Id Attributwert des Ad Tags.	Zeichenfolge	ad1
width	Die Breite des Video-Assets in Pixeln.	Zeichenfolge	640
xPosition	Die horizontale Position eines Symbols im Videoplayer. Dies kann ein bestimmter Pixelwert oder eine Position wie „links“ oder „rechts“ sein. Pfad: /avails/ads/icons/attributes/xPosition	Zeichenfolge	left oder 10
yPosition	Die vertikale Position eines Symbols im Videoplayer. Dies kann ein bestimmter Pixelwert oder eine Position wie „oben“ oder „unten“ sein. Pfad: /avails/ads/icons/attributes/yPosition	Zeichenfolge	top oder 10

Eigenschaft	Definition	Werttyp	Beispiel
<code>iconClicks</code>	Enthält Informationen zum Durchklicken und zur Nachverfolgung eines Symbols. Pfad: <code>/avails/ads/icons/iconClicks</code>	Objekt	
<code>iconClickThrough</code>	Eine URL zur Seite des Werbetreibenden, die der Media Player öffnet, wenn der Zuschauer auf das Symbol klickt. Pfad: <code>/avails/ads/icons/iconClicks/iconClickThrough</code>	Zeichenfolge	<code>https://advertiser.com/landing-page</code>
<code>iconClickTracking</code>	Die Tracking-URL für die <code>iconClickThrough</code> Immobilie. Pfad: <code>/avails/ads/icons/iconClicks/iconClickTracking</code>	Objekt	
<code>iconClickFallbackImages</code>	Eine Reihe von Ersatzbildern, die angezeigt werden sollen, falls das Symbol nicht angezeigt werden kann. Pfad: <code>/avails/ads/icons/iconClicks/iconClickFallbackImages</code>	Array	
<code>iconViewTracking</code>	Die URL für die Nachverfolgung, wenn ein Symbol angezeigt wird. Pfad: <code>/avails/ads/icons/iconViewTracking</code>	Zeichenfolge	<code>https://tracking.example.com/icon-view</code>

Eigenschaft	Definition	Werttyp	Beispiel
offset	Der Zeitversatz, zu dem ein Symbol während der Anzeigenwiedergabe erscheinen soll. Pfad: /avails/ads/icons/attributes/offset	Zeichenfolge	00:00:05
program	Das Programm oder die Initiative, die mit dem Symbol verknüpft ist, z. B. "AdChoices". Pfad: /avails/ads/icons/attributes/program	Zeichenfolge	AdChoices
pxratio	Das Pixelverhältnis für das Symbol oder die Begleitanzeige, das für Displays mit hohem DPI-Wert verwendet wird. Pfad: oder /avails/ads/icons/attributes/pxratio / avails/ads/companionAds/attributes/pxratio	Zeichenfolge	1 oder 2
type	Der Typ der Ressource oder Erweiterung. Pfad: /avails/ads/extensions/type oder /avails/ads/adVerifications/executableResource/type	Zeichenfolge	text/java script
content	Der Inhalt einer Erweiterung. Pfad: /avails/ads/extensions/content	Zeichenfolge	

Eigenschaft	Definition	Werttyp	Beispiel
language	Die Programmiersprache einer ausführbaren Ressource. Pfad: /avails/ads/adVerifications/executableResource/language	Zeichenfolge	javascript
browserOptional	Gibt an, ob Browserunterstützung für die JavaScript Ressource erforderlich ist. Pfad: /avails/ads/adVerifications/javascriptResource/browserOptional	Zeichenfolge	true oder false
id	Ein Bezeichner für verschiedene Elemente im Schema. Pfad: /avails/ads/companionAds/attributes/id oder /avails/ads/icons/iconClicks/iconClickTracking/id	Zeichenfolge	companion-1
assetHeight	Die Höhe der Begleitperson und des Assets. Pfad: /avails/ads/companionAds/attributes/assetHeight	Zeichenfolge	250
assetWidth	Die Breite des Begleit-Assets. Pfad: /avails/ads/companionAds/attributes/assetWidth	Zeichenfolge	300

Eigenschaft	Definition	Werttyp	Beispiel
<code>expandedHeight</code>	Die Höhe der Begleitanzeige im erweiterten Zustand. Pfad: <code>/avails/ads/companionAds/attributes/expandedHeight</code>	Zeichenfolge	<code>600</code>
<code>expandedWidth</code>	Die Breite der Begleitanzeige, wenn sie erweitert wird. Pfad: <code>/avails/ads/companionAds/attributes/expandedWidth</code>	Zeichenfolge	<code>600</code>
<code>renderingMode</code>	Der Rendermodus für die Begleitanzeige. Pfad: <code>/avails/ads/companionAds/attributes/renderingMode</code>	Zeichenfolge	<code>default</code> oder <code>transparent</code>
<code>adSlotId</code>	Die ID der Werbefläche, in der die Begleitanzeige angezeigt werden soll. Pfad: <code>/avails/ads/companionAds/attributes/adSlotId</code>	Zeichenfolge	<code>banner-1</code>
<code>creativeType</code>	Der MIME-Typ des kreativen Assets. Pfad: <code>/avails/ads/icons/staticResource/creativeType</code>	Zeichenfolge	<code>image/png</code>

Zeitlicher Ablauf der Aktivitäten bei der Anzeigenverfolgung

Bei der clientseitigen Berichterstattung muss der Spieler Tracking-Ereignisse (Beacons) mit einer gewissen Genauigkeit aussenden. Mithilfe des MediaTailor clientseitigen Tracking-Schemas können Sie sicherstellen, dass für alle Verfügbarkeits-, Anzeige-, Begleitungs-, Overlay- und Tracking-Ereignisse Informationen zum Zeitpunkt und zur Dauer vorliegen, und zwar in unterschiedlicher Form.

Verwenden Sie die folgenden MediaTailor Schlüssel/Wert-Paare für den Player, um Werbeereignisse, wie z. B. das Verfolgen von Ereignissen, genau mit der Wiedergabeposition in Einklang zu bringen:

- [startTime](#)
- [startTimeInSeconds](#)
- [adProgramDateTime](#)
- [adID/eventId](#)

HLS und DASH implementieren den Wert von und unterschiedlich: `startTime`
`startTimeInSeconds`

- HLS — Die `startTime` Werte beziehen sich auf den Beginn der Wiedergabe-Sitzung. Der Beginn der Wiedergabe-Sitzung ist als Zeit Null definiert. Bei der Anzeige `startTime` handelt es sich um die Summe der kumulierten Werte aller EXT-INF Segmentdauern, die bis zur Veröffentlichung geführt haben. Die Mediensequenznummer des Segments, auf das sich die Anzeige oder das Tracking-Ereignis bezieht, entspricht auch dem `adId` oder `eventId` in der clientseitigen Tracking-Antwort.
- DASH:
 - Live/dynamische Manifeste — Die `startTime` Werte beziehen sich auf die Werte des DASH-Manifests `MPD@availabilityStartTime`. Der `MPD@availabilityStartTime` ist ein Timing-Anker für alle MediaTailor Sessions, die den Stream nutzen.
 - VOD/statische Manifeste — Die `startTime` Werte beziehen sich auf den Beginn der Wiedergabe-Sitzung. Der Beginn der Wiedergabesitzung ist als Zeitpunkt Null definiert. Jede Anzeige im Avail ist in einem eigenen Period Element enthalten. Das Period Element hat ein `@start` Attribut mit einem Wert, der den `startTime` Werten in der clientseitigen Tracking-Payload entspricht. Das entspricht `PeriodId` auch dem `adId` oder `eventId` in der clientseitigen Tracking-Antwort.

Example HLS:

Im folgenden Beispiel wurde die MediaTailor Sitzung gestartet, und das folgende Manifest ist das erste, das dem Client zugestellt wird:

```
#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
```

In der JSON-Nutzlast für die clientseitige Nachverfolgung gelten die folgenden Werte:

- `startTime`: "PT18.581355S"
- `startTimeInSeconds`: 18.581
- `availProgramDateTime`: "2023-05-03T21:24:41.453Z"

- adId: 4603269

Example DASH:

Im folgenden Beispiel erhält die MediaTailor Sitzung eine Midroll im Manifest. Beachten Sie, dass der @start Attributwert der zweiten Periode, bei der es sich um den Anzeigenzeitraum handelt, einen Wert hat, der MPD@availabilityStartTime relativ zum Wert ist. Dieser Wert wird für alle Sitzungen in die clientseitigen startTime Tracking-Antwortfelder MediaTailor geschrieben.

```
<?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/
  ittf/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>
```

In der JSON-Nutzlast für clientseitiges Tracking gelten die folgenden Werte:

- startTime: "PT5042H27M59.931S"

- `startTimeInSeconds`: 18152879.931
- `availProgramDateTime`: *null*
- `adId`: 1683151599194_1_1

Player-Bedienelemente und Funktionen für das clientseitige Ad-Tracking

MediaTailor Die clientseitigen Tracking-Metadaten unterstützen verschiedene Player-Steuerelemente und Funktionen. In der folgenden Liste werden beliebte Player-Steuerelemente beschrieben.

Themen

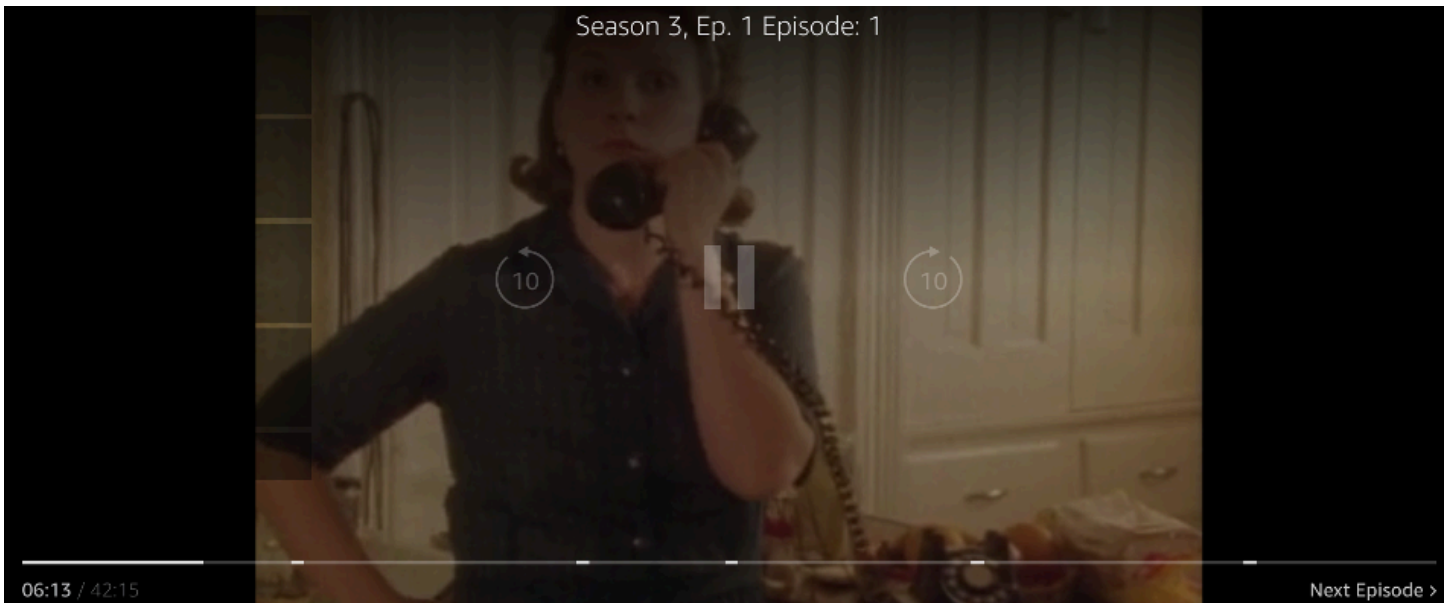
- [Schrubben](#)
- [Countdown-Timer hinzufügen](#)
- [Überspringbare Anzeigen](#)
- [Und Click-through](#)
- [Begleitende Anzeigen](#)
- [Interaktive Anzeigen \(SIMID\)](#)
- [Interaktive Werbung \(VPAID\)](#)
- [Symbole für Google Warum diese Anzeige \(WTA\)](#)

Schrubben

Um das Wiedergabeerlebnis zu verbessern, kann der Player Anzeigenpositionen in der Wiedergabe-Timeline anzeigen. MediaTailor stellt diese Anzeigenpositionen in Form von `startTimeInSeconds` Werten in der clientseitigen Tracking-Antwort zur Verfügung.

Note

Einige Streaming-Anbieter verhindern, dass Sie über eine Anzeigenposition hinausgehen.

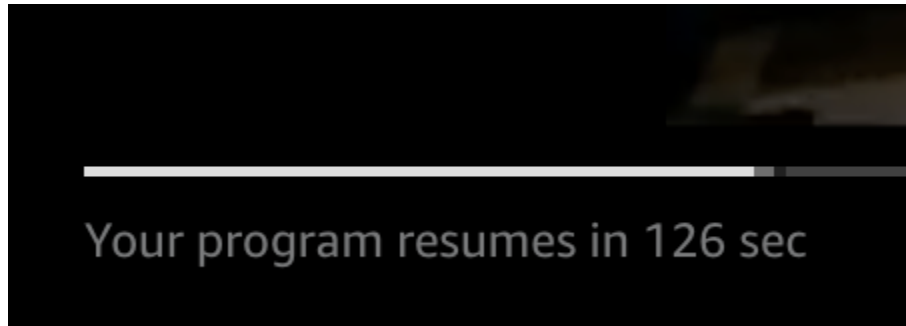


Die folgende JSON-Antwort auf die clientseitige Tracking-Nutzlast zeigt die Startzeit der Verfügbarkeit (Werbeunterbrechung) innerhalb des JSON-Stammobjekts des Avails-Arrays. Der Player verwendet diese Daten, um den Ort der Werbeunterbrechung in der Player-Zeitleiste anzuzeigen, und zwar nach 28 Sekunden.

```
{
  "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": "UFQxMk0zNC44NjhTXzIwMjMtMDctMjFUMjA6MjMMDcuNzc1NzE2MzAyWl8x",
  "nonLinearAvails": []
}
```

Countdown-Timer hinzufügen

Mithilfe eines Countdown-Timers für Werbeanzeigen können MediaTailor Sie dafür sorgen, dass Ihr Publikum während der Werbepause bei der Stange bleibt. Anhand des Timers kann die Zielgruppe nachvollziehen, wann die Werbepause endet und ihr Programm wieder aufgenommen wird.



Die Elemente in den clientseitigen Tracking-Metadaten, die für den Countdown-Timer der Anzeige eine Rolle spielen, sind `startTime`, und `startTimeInSeconds` `duration` `durationInSeconds`. Der Player verwendet diese Metadaten zusammen mit der verstrichenen Sitzungszeit, die er separat verfolgt, um zu bestimmen, wann der Timer angezeigt werden soll und ab welchem Wert er herunterzählen soll.

Die folgende JSON-Antwort auf die clientseitige Tracking-Payload zeigt die Informationen, die für die Anzeige eines Countdown-Timers für Anzeigen erforderlich sind.

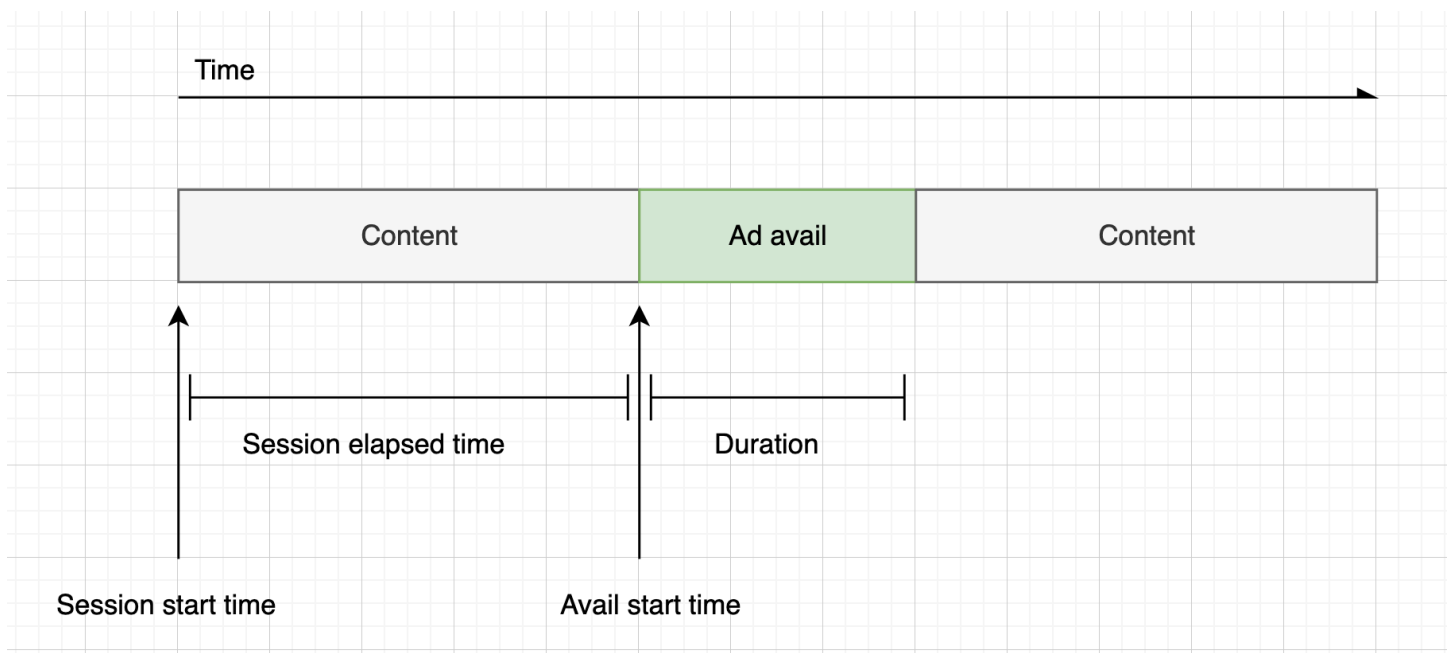
```
{
  "avails": [
    {
      "adBreakTrackingEvents": [],
      "adMarkerDuration": null,
      "ads": [...],
      "availId": "7",
      "availProgramDateTime": null,
      "duration": "PT30S",
      "durationInSeconds": 30,
      "meta": null,
      "nonLinearAdsList": [],
      "startTime": "PT28S",
      "startTimeInSeconds": 28
    }
  ],
  "dashAvailabilityStartTime": null,
  "hlsAnchorMediaSequenceNumber": null,
  "nextToken": "UFQxMk0zNC44NjhTXzIwMjMtMDctMjFUMjA6MjM6MDcuNzc1NzE2MzAyWl8x",
}
```

```
"nonLinearAvails": []
}
```

Wenn die verstrichene Zeit der Sitzung die Startzeit des Avails erreicht, zeigt der Player einen Countdown-Timer mit einem Wert an, der der Dauer der Verfügbarkeit entspricht. Der Wert des Countdown-Timers nimmt ab, wenn die verstrichene Zeit die Startzeit des Avails überschreitet.

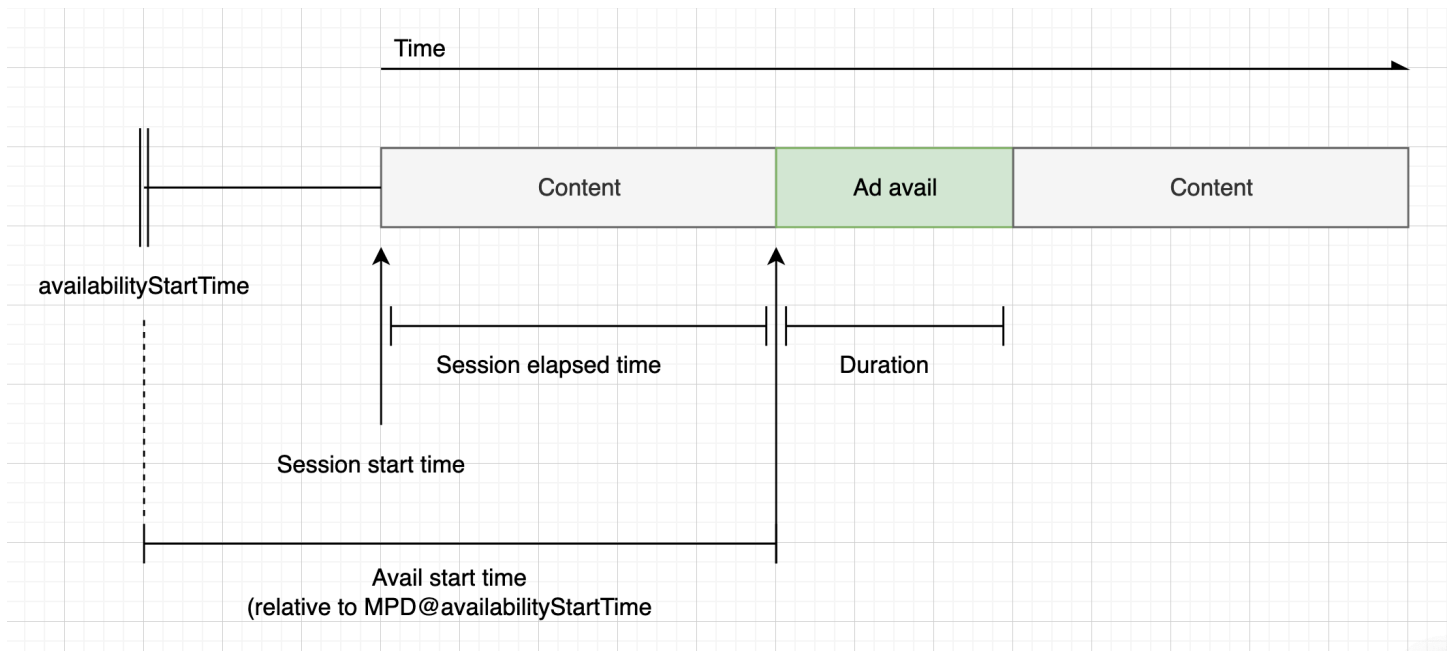
Example Formel: Countdown-Timer für HLS (Live und VOD) und DASH (VOD)

- $\text{session\_start\_time} = \text{die Summe aller EXT-INF Dauerwerte} - \text{der Dauerwert der drei neuesten Mediensequenzen EXT-INF}$
- $\text{Timer-Wert} = \text{duration} - (\text{session\_elapsed\_time} - \text{startTime})$



Example Formel: Countdown-Timer für DASH (live)

- $\text{session\_start\_time} = (\text{neuestes Segment startTime} + \text{duration}) / \text{timescale} - \text{MPD@suggestedPresentationDelay}$
- $\text{Timer-Wert} = \text{duration} - (\text{session\_elapsed\_time} - \text{startTime})$



Überspringbare Anzeigen

Überspringbare Anzeigen sind Werbespots, die es dem Zuschauer ermöglichen, einen Teil der Anzeige zu überspringen, um das Programm wieder anzusehen. In VAST identifiziert das `Linear@skipoffset` Attribut eine Anzeige, die übersprungen werden kann.

Die folgende VAST-Antwort zeigt, wie eine überspringbare Anzeige verwendet wird:

```
<?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>

```

Die folgende JSON-Antwort für die clientseitige Tracking-Nutzlast zeigt die Anzeigenmetadaten innerhalb des Arrays. ads Das Array enthält den skipOffset Wert, der aus der MediaTailor VAST-Antwort abgerufen wurde.

```

{
  "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": []
}

```

Und Click-through

Mithilfe von Click-Through URIs können Werbetreibende messen, wie erfolgreich eine Anzeige die Aufmerksamkeit der Zuschauer auf sich zieht. Nachdem ein Zuschauer auf den aktiven Videoframe einer laufenden Anzeige geklickt hat, öffnet ein Webbrowser den URI für die Startseite oder Kampagnen-Landingpage des Werbetreibenden. Der Player-Entwickler bestimmt das Klickverhalten, z. B. das Überlagern einer Schaltfläche oder eines Labels auf dem Werbevideo mit einer Meldung, auf die Sie klicken müssen, um mehr zu erfahren. Player-Entwickler pausieren das Video der Anzeige häufig, nachdem Zuschauer auf den aktiven Videoframe geklickt haben.



Click here for deals on Amazon.com

MediaTailor kann jedes lineare Video-Klick-Through-Ereignis, das in der VAST-Antwort URLs zurückgegeben wurde, analysieren und verfügbar machen. Die folgende VAST-Antwort zeigt ein Beispiel für einen Klick auf eine Anzeige.

```
<?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>

```

Die folgende JSON-Antwort für die clientseitige Tracking-Payload zeigt, wie Click-Through und Click-Tracking MediaTailor innerhalb des Arrays dargestellt werden. URLs `trackingEvents` Der `clickThrough` Ereignistyp steht für die Click-Through-Anzeige, und der Ereignistyp steht für die Click-Tracking-URL. `clickTracking`

```

{
  "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": "UFQz0S4zMz1TXzIwMjMtMDctMzFUMTY6NTQ6MDQuODA1Mzk2NTI5Wl8x",
"nonLinearAvails": []
}

```

Begleitende Anzeigen

Eine Begleitanzeige erscheint zusammen mit einem linearen Werbemittel. Verwenden Sie Begleitanzeigen, um die Effektivität eines Werbespots zu erhöhen, indem Sie Produkt-, Logo- und Markeninformationen anzeigen. Die Display-Anzeige kann QR-Codes (Quick Response) und anklickbare Bereiche enthalten, um das Engagement der Zuschauer zu fördern.

MediaTailor unterstützt Begleitanzeigen in der VAST-Antwort. Es kann Metadaten von `StaticResource` oder `FrameResource`, bzw. `HTMLResource` Knoten übertragen.

Die folgende VAST-Antwort zeigt ein Beispiel für Position und Format der linearen Anzeige und der Begleitanzeige.

```

<?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>

```

Die Daten werden in der clientseitigen Tracking-Antwort in der `/avail/x/ads/y/companionAds` Liste angezeigt. Jedes lineare Werbemittel kann bis zu 6 Begleitanzeigen enthalten. Wie im Beispiel unten gezeigt, werden die Begleitanzeigen in einer Liste angezeigt

 Note

Als bewährte Methode sollten Anwendungsentwickler eine Logik implementieren, mit der die Begleitanzeige am Ende der Werbeaktion explizit entfernt oder entfernt wird.

```
{
  "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",
            }
          ],
          "sequence": "1",
          "staticResource": "https://emt.com/companion/9973499273",
        }
      ],
      "sequence": "1",
      "staticResource": "https://emt.com/companion/9973499273",
    }
  ],
  "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": "UFQxMFNfMjAyMy0wNy0wNlQyMToxMDowOC42NzQ4NDA1NjJaXzE%3D",
"nonLinearAvails": []
}

```


Interaktive Anzeigen (SIMID)

SecureInteractive Media Interface Definition (SIMID) ist ein Standard für interaktive Werbung, der im VAST 4.x-Standard des Interactive Advertising Bureau (IAB) eingeführt wurde. SIMID entkoppelt das Laden interaktiver Elemente vom primären linearen Werbematerial auf dem Player und bezieht sich in der VAST-Antwort auf beide. MediaTailor fügt das primäre Creative hinzu, um das Wiedergabeerlebnis aufrechtzuerhalten, und platziert Metadaten für die interaktiven Komponenten in der clientseitigen Tracking-Antwort.

In der folgenden VAST-4-Beispielantwort befindet sich die SIMID-Nutzlast innerhalb des Knotens `InteractiveCreativeFile`

```
<?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>

```

In der folgenden VAST-3-Antwort befindet sich die SIMID-Nutzlast innerhalb des Knotens. Extensions

```

<?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>

```

In der folgenden clientseitigen Tracking-Antwort werden die SIMID-Daten in der Liste angezeigt. /
 avails/x/ads/y/extensions

```

{
  "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": ""
}
```

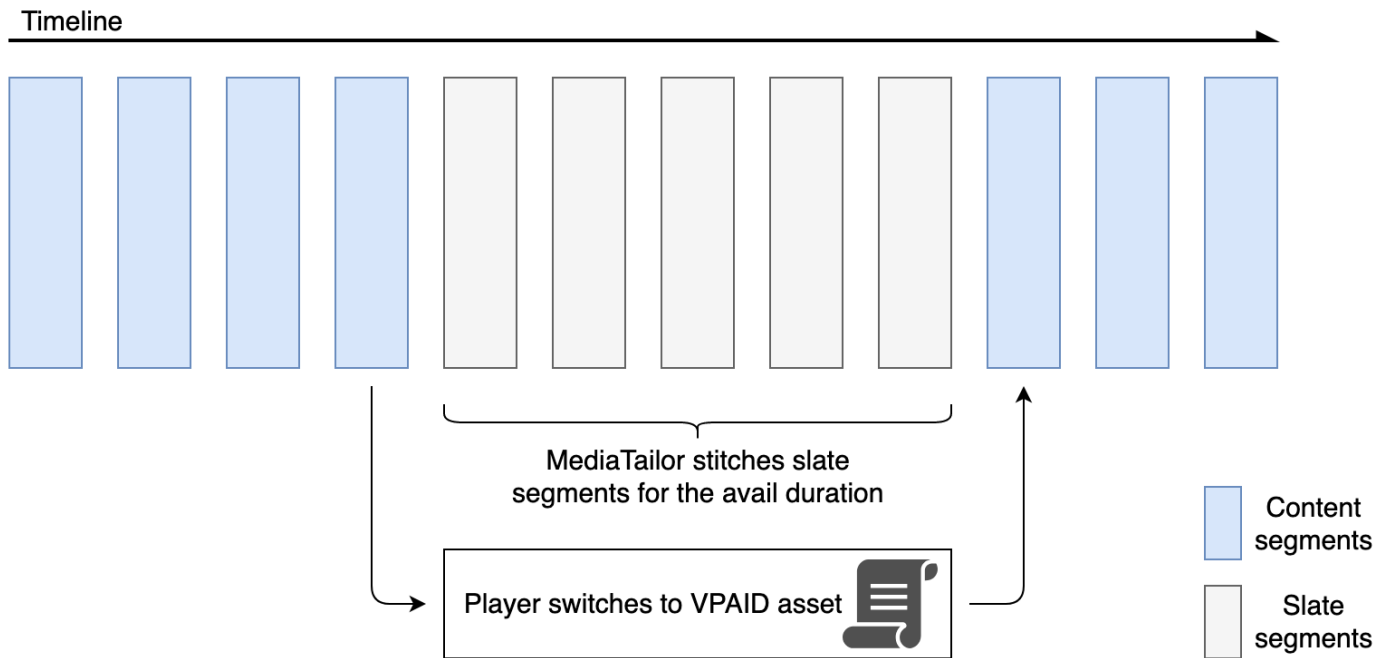
```
    "availId": "2698188",
    "availProgramDateTime": "2023-07-31T16:53:40.577Z",
    "duration": "PT14.982S",
    "durationInSeconds": 14.982,
    "meta": null,
    "nonLinearAdsList": [],
    "startTime": "PT39.339S",
    "startTimeInSeconds": 39.339
  }
],
"dashAvailabilityStartTime": null,
"hlsAnchorMediaSequenceNumber": null,
"nextToken": "UFQz0S4zMz1TXzIwMjMtMDctMzFUMTY6NTQ6MDQu0DA1Mzk2NTI5Wl8x",
"nonLinearAvails": []
}
```

Interaktive Werbung (VPAID)

Die Video Player Ad Interface Definition (VPAID) spezifiziert das Protokoll zwischen der Anzeige und dem Videoplayer, das Anzeigeninteraktivität und andere Funktionen ermöglicht. MediaTailor unterstützt bei Live-Streams das VPAID-Format, indem Slate-Segmente für die Dauer des Angebots zusammengefügt und Metadaten für die VPAID-Kreativen in die clientseitige Tracking-Antwort eingefügt werden, die der Videoplayer verarbeitet. Der Player lädt die VPAID-Dateien herunter, spielt das lineare Creative ab und führt die Skripte des Kunden aus. Der Spieler sollte niemals die Slate-Segmente spielen.

Note

VPAID ist ab VAST 4.1 veraltet.



Das folgende Beispiel zeigt den VPAID-Inhalt in der VAST-Antwort.

```
<?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>

```

Das folgende Beispiel zeigt die Tracking-Informationen.

```

{
  "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": []
}{
    "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": "UFQ4TTQyLjI4OVNfMjAyMy0wOC0xNFQxOT01MDo0MS4zOTc5MjAzODVaXzE%3D",
"nonLinearAvails": []
}

```

Symbole für Google Warum diese Anzeige (WTA)

AdChoices ist ein Industriestandard, der Zuschauern Informationen über die Anzeigen bietet, die sie sehen, einschließlich der Art und Weise, wie diese Anzeigen auf sie ausgerichtet wurden.



Die MediaTailor clientseitige Tracking-API unterstützt Symbolmetadaten, die im VAST-Erweiterungsknoten der VAST-Antwort gespeichert sind. Weitere Informationen zu WTA in der VAST-Antwort finden Sie in [dieser VAST-XML-Beispielantwort](#).

Note

MediaTailor unterstützt derzeit nur VAST Version 3.

```

<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>

```

Das folgende Beispiel zeigt die clientseitige Tracking-Antwort in der /avails/x/ads/y/extensions Liste.

```

{
  "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": "UFQxMFNfMjAyMy0wNy0wNlQyMDo0MT0xNy45NDE4MDM0NDhaXzE%3D",
"nonLinearAvails": []
}

```

Clientseitiges Beaconsing

Mit dem clientseitigen `startTimeInSeconds` Tracking-Element können Sie MediaTailor das Timing von Beacons unterstützen.

Die folgende JSON-Antwort zeigt die wichtigsten Beacon-Typen: Impressionen, Start, Quartile und Abschluss.

Note

In den Richtlinien des Interactive Advertising Bureau (IAB) zur Messung von Videoimpressionen heißt es, dass für eine Impression der Anzeigeninhalt clientseitig geladen werden muss und mindestens die Startzeit für das Rendern im Player erreicht werden muss.

Weitere Informationen finden Sie unter [Digital Video Ad Serving Template \(VAST\)](#) auf der IAB-Website.

```
{
  "avails": [
    {
      "ads": [
        {
          "adId": "8104385",
          "duration": "PT15.100000078S",
          "durationInSeconds": 15.1,
          "startTime": "PT17.817798612S",
          "startTimeInSeconds": 17.817,
          "trackingEvents": [
            {
              "beaconUrls": [
                "http://exampleadserver.com/tracking?event=impression"
              ],
              "duration": "PT15.100000078S",
              "durationInSeconds": 15.1,
              "eventId": "8104385",
              "eventType": "impression",
              "startTime": "PT17.817798612S",
              "startTimeInSeconds": 17.817
            },
            {
              "beaconUrls": [
                "http://exampleadserver.com/tracking?event=start"
              ],
              "duration": "PT0S",
              "durationInSeconds": 0.0,
              "eventId": "8104385",
              "eventType": "start",
              "startTime": "PT17.817798612S",
              "startTimeInSeconds": 17.817
            },
            {
              "beaconUrls": [
                "http://exampleadserver.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
  }
]
```

Hybridmodus mit serverseitigen Ad Beacons

MediaTailor unterstützt einen Hybridmodus für die Sitzungsverfolgung. In diesem Modus sendet der Dienst Werbeverfolgungsereignisse im Zusammenhang mit der Wiedergabe aus, stellt aber die gesamte clientseitige Tracking-Nutzlast für die Sitzung zur Verfügung

Um Hybrid-Tracking mithilfe von Wiedergabepräfixen zu aktivieren, initialisieren Sie vom Player aus eine neue MediaTailor Wiedergabe-Sitzung mit einer Anfrage in einem der folgenden Formate gemäß Ihrem Protokoll:

Example: HLS-Format

```
POST master.m3u8
{
  "adsParams": {
    "deviceType": "ipad"
  },
  "reportingMode": "server"
}
```

Example: DASH-Format

```
POST manifest.mpd
{
  "adsParams": {
    "deviceType": "ipad"
  },
  "reportingMode": "server"
}
```

MediaTailor verwaltet die folgenden Tracking-Ereignisse im Hybridmodus:

- Eindruck
- Starten

- Erstes Quartil
- Mittelpunkt
- Drittes Quartil
- Complete
- `breakStart(vmap)`
- `breakEnd(vmap)`

Kundenseitige Ad-Tracking-Integrationen

In diesem Abschnitt werden Integrationen zwischen MediaTailor und verschiedenen clientseitigen Ad-Tracking-Servern beschrieben.

Themen

- [Öffnen Sie Measurement SDK](#)
- [Kostenloser Datazoom-Player SDKs](#)
- [Roku Advertising Framework \(RAF\)](#)
- [TheoPlayer](#)
- [MediaTailor SDK](#)

Öffnen Sie Measurement SDK

Das Open Measurement SDK (OM SDK) des Interactive Advertising Bureau (IAB) ermöglicht die Messung der Sichtbarkeit und Überprüfung von Anzeigen, die in Web-Video- und nativen App-Umgebungen geschaltet werden.

Bei älteren VAST-Dokumenten der Version 3 sollte der Bestätigungscode mit dem Erweiterungstyp zusammen mit dem Erweiterungsknoten geladen werden. AdVerifications Die Wurzel des Erweiterungsknotens ist ein AdVerifications Knoten mit demselben Schema wie das VAST 4.1-Element.

Um die Einführung des OM SDK zu erleichtern, MediaTailor hat sich das Unternehmen mit Datazoom zusammengetan, um kostenlose Player zur Verfügung zu stellen SDKs , die für Open Measurement konfiguriert und verifiziert sind. Weitere Informationen finden Sie unter [Kostenloser Datazoom-Player SDKs](#).

Note

MediaTailor unterstützt derzeit nur VAST Version 3.

Example: Verifizierungsknoten in VAST 3, vor Version 4.1

```
...
<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 extrahiert die AdVerifications Daten aus dem <Extensions> Knoten und platziert sie in dem adVerifications Array in der clientseitigen Tracking-Antwort.

Example: ADVerifications-Array in der clientseitigen Tracking-Antwort

```
{
  "avails": [
    {
      "adBreakTrackingEvents": [],
      "adMarkerDuration": null,
      "ads": [
        {
          "adId": "3062770",
          "adParameters": "",
```

[illegible]

```

    },
    "skipOffset": null,
    "startTime": "PT10.11S",
    "startTimeInSeconds": 10.11,
    "trackingEvents": [
      {
        "beaconUrls": [
          "https://n8ljfs0h09.execute-api.us-west-2.amazonaws.com/v1/impression"
        ],
        "duration": "PT14.982S",
        "durationInSeconds": 14.982,
        "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

Wenden Sie sich an das IAB Tech Lab, um sicherzustellen, dass die Anwendungen jährlich zertifiziert werden, um die Einhaltung der Vorschriften sicherzustellen.

Weitere Informationen zum OM SDK finden Sie unter [Open Measurement SDK](#) auf der Website des IAB Tech Lab.

Kostenloser Datazoom-Player SDKs

Um die Einführung des Players zu erleichtern SDKs, MediaTailor hat es sich mit Datazoom zusammengetan, um kostenlose Player anzubieten SDKs, die mit dem und dem IAB Tech konfiguriert und getestet wurden. [Kundenseitige AWS Elemental MediaTailor Integration mit Google Ad Manager Öffnen Sie Measurement SDK](#)

Das Datazoom Player SDK unterstützt die folgenden Funktionen:

- Live- und VOD-Wiedergabelisten
- DASH- und HLS-Spezifikationen
- Unterstützung von Player-Anbietern für Bitmovin, Exoplayer, Android Media Player, Apple AVPlayer, Brightcove, Chromecast Receiver, Dash.js, hls.js,, Shaka Player, THEO Player, Video.js JWPlayer, Roku und mehr
- IAB Tech Lab Open Measurement-Zertifizierung, sofern für ausgewählte Geräte verfügbar
- Behandlung von Ereignissen per Mausklick
- Dispatcher für Werbeereignisse, z. B. Countdown-Timer für Werbeanzeigen, Anzeigenüberlagerungen und nichtlineare Ereignisse, Beginn und Ende von Werbeunterbrechungen
- Kundenseitiges Werbe-Beaconing
- SDK für die Google Programmatic Access Library (PAL) als optionale Konfigurationseinstellung

Datazoom bietet auch einen kostenpflichtigen Analyse- und Telemetriedienst, den der Spieler unterstützt. SDKs Kunden können die Player-SDK-Telemetrie über die Datazoom-Managementkonsole aktivieren und steuern. [Um auf den Datazoom Player zuzugreifen SDKs und mehr über den Telemetrie- und Analysedienst mit Mehrwert zu erfahren, verwenden Sie die Kontaktinformationen auf der Datazoom-Website.](#)

Roku Advertising Framework (RAF)

Das Roku Ad Framework (RAF) sorgt für ein einheitliches Werberlebnis auf der gesamten Roku-Plattform. Alle Kanäle, einschließlich Videowerbung, müssen die Zertifizierungsanforderungen von Roku für RAF erfüllen. Insbesondere muss die App immer die clientseitige Ereignisauslösung über

RAF verwenden. MediaTailor unterstützt als Anbieter für serverseitige Anzeigeneinfügung (SSAI) die clientseitige Auslösung von Ereignissen. Die RAFX SSAI-Adapter bieten Schnittstellen sowohl zu SSAI-Manifestservern oder Stitchern als auch zu RAF. Zu diesen Schnittstellen gehören:

- Analysieren der `masterURL` Antwort und Extrahieren von `playURLAdURL`, und Anzeigenmetadaten.
- Umwandlung von MediaTailor SSAI-Anzeigenmetadaten in RAF-verwendbare Anzeigenmetadaten und Konfiguration von RAF für die Wiedergabe.
- Beobachtung von Stream-Ereignissen und zeitgesteuerten Metadaten.
- Rechtzeitiger Abgleich der Stream-Ereignisse, der Anzeigen-Metadaten und der Pixel für das Auslöseereignis.
- Pinggen/Abfragen von `RAFAdURL`, wie vom MediaTailor SSAI-Manifestserver gefordert, anschließendes Parsen und Neukonfigurieren von RAF.

Weitere Informationen zu SSAI-Adaptoren für RAF finden Sie unter [Implementieren der serverseitigen Anzeigeneinfügung](#) mithilfe von Roku-Adaptoren auf der Roku-Website.

TheoPlayer

TheoPlayer Die Integration mit MediaTailor macht Folgendes:

- Bietet Funktionen zur Unterstützung der MediaTailor clientseitigen Ereignisverfolgung für HLS und DASH sowohl für VOD- als auch für Live-Workflows.
- Unterstützt das Senden von Tracking-Beacons nur für lineare Anzeigen.
- Deaktiviert die Suche während einer Anzeige. Es gibt jedoch keine Logik für das Abspielen einer Anzeige, wenn der Nutzer nach Ablauf der Werbeunterbrechung sucht.

Weitere Informationen zu SSAI und zur Überprüfung im TheoPlayer Internet, Android, iOS und tvOS SDKs für MediaTailor finden Sie [MediaTailor](#) auf der TheoPlayer Website.

MediaTailor SDK

AWS Elemental unterhält ein JavaScript basiertes Softwareentwicklungskit (SDK). AWS Elemental stellt das SDK so zur Verfügung, wie es ist, ohne stillschweigende Garantie. Verwenden Sie das SDK als Referenz-Demo, um Ihr Onboarding auf die Nutzung zu optimieren. MediaTailor Das SDK zeigt, wie Sie mit der MediaTailor clientseitigen Tracking-API interagieren. Das SDK implementiert clientseitiges Ad-Tracking und Reporting für aktive Spieler. HTML5 Das SDK initialisiert

eine MediaTailor clientseitige Berichtssitzung und fordert dann regelmäßig Informationen zur Anzeigenverfolgung an. Während der Wiedergabe gibt das SDK Anzeigen-Tracking-Ereignisse aus, wenn neue Werbeereignisse erkannt werden.

Das MediaTailor SDK unterstützt die folgenden Funktionen:

- Live- und VOD-Playlisten
- DASH- und HLS-Spezifikationen
- Behandlung von Ereignissen per Mausklick
- Dispatcher für Werbeereignisse
- Benutzerdefinierte Event-Hooks
- Client-seitiges Werbe-Beaconing. Weitere Informationen zum Senden von Ad Beacons finden Sie unter. [Clientseitiges Beaconing](#)

 Note

Reichen Sie ein AWS Support-Ticket ein, um ein JavaScript Beispiel-SDK für zu erhalten MediaTailor. Sie erhalten einen Download-Link für das Paket und seine Dateien.

Durch Werbe-Beacons blättern mit GetTracking

Verwenden Sie den `GetTracking` Endpunkt, um die Anzahl der an einen Player zurückgegebenen Anzeigen einzuschränken. Wenn ein Manifestfenster beispielsweise groß ist und sich über einen langen Zeitraum erstreckt, kann sich die Anzahl der zurückgegebenen Werbe-Beacons auf die Leistung des Players auswirken.

`GetTracking` gibt einen `NextToken` Wert zurück, mit dem Sie die Anzahl der zurückgegebenen Beacons einschränken können, indem Sie die Liste der zurückgegebenen Beacons durchblättern. Sie können die `NextToken` Werte nacheinander durchgehen, um den gewünschten Wert im Feld eines Werbe-Beacons zu finden. `StartTimeInSeconds`

- Beim ersten Aufruf von werden alle möglichen Anzeigen zurückgegeben `GetTracking`, die in das Manifestfenster fallen, einschließlich jeweils eines Werts `NextToken` und.
- Wenn eine `GetTracking` Anfrage kein A enthält `NextToken`, werden alle Anzeigen im Manifestfenster zurückgegeben.

- Wenn eine `GetTracking` Anfrage ein `NextToken` enthält, es `NextToken` aber keine neuen Beacons gibt, die MediaTailor zurückgegeben werden können, wird derselbe Wert zurückgegeben `NextToken`, den Sie in der ursprünglichen Anfrage gesendet haben.
- Wenn einer Anzeige keine Beacons mehr entsprechen, `GetTracking` wird die Anzeige aus der Antwort entfernt.
- Tokens von `GetTracking` laufen nach 24 Stunden ab. Wenn ein `NextToken` Wert älter als 24 Stunden ist, gibt der nächste Aufruf von einem `GetTracking` `NextToken` Nullwert zurück.

Generalisierte Aufrufsequenz von vom Spieler `GetTracking`

Eine `GetTracking` Anfrage vom Client-Spieler ist ein POST mit einem Anforderungstext, der die Werbung `NextToken` und Beacons im Zusammenhang mit dem Token enthält.

```
https://YouMediaTailorUrl/v1/tracking
{
    "NextToken": "value"
    .
    .
    .
}
```

Die allgemeine Reihenfolge für die Verwendung von `GetTracking` with `NextToken` lautet wie folgt:

1. Rufen Sie zum ersten Mal `anGetTracking`.

Alle Anzeigen und Beacons sowie die ersten `NextToken` für nachfolgende Aufrufe werden zurückgegeben.

2. Wenn der Wert Null `NextToken` ist, werden alle Ad Beacons MediaTailor zurückgegeben.
3. Wenn der abgelaufen `NextToken` ist, MediaTailor wird eine HTTP-Rückgabecode 400-Fehlermeldung zurückgegeben.

Rufen Sie erneut auf, `GetTracking` um gültige `NextToken` s abzurufen.

4. Scannen Sie die gesamte Antwort, um herauszufinden, `StartTimeInSeconds` ob sich ein Werbeanzeigen-Beacon im gewünschten Bereich befindet.

5. Rufen Sie erneut `GetTracking` mit dem Wert von `auf`, der dem gewünschten `StartTimeInSeconds` Wert `NextToken` zugeordnet ist.
6. Schauen Sie sich bei Bedarf die zurückgegebenen Anzeigen erneut an, bis Sie genau die Anzeigen gefunden haben, die Sie abspielen möchten.

Erweitertes Beispiel

Dieses Beispiel zeigt, wie Sie mithilfe `GetTracking` von `s` die Anzahl der `NextToken` an einen Player zurückgegebenen Werbe-Beacons einschränken können.

MediaTailor erhält eine `GetTracking` Anfrage. Die Antwort enthält eine Anzeige mit der ID 9935407 und zwei Beacons mit den `StartTimeInSeconds` Werten 52,286 und 48,332 Sekunden.

MediaTailor sendet die JSON-Antwort wie folgt: `NextToken`

```
{
  "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
}
]
}

```

MediaTailor antwortet bei der nächsten GetTracking Anfrage mit dem NextToken Wert: JF57 ITe48t1441mv7 Tm LKu ZLrox Dzflslp 6Bi SNL1 IJmz PVMDN0lqr BYycg MbKEb.

MediaTailor antwortet mit Anzeigen und Beacons, die StartTimeInSeconds denen entsprechen, die NextToken beim vorherigen Anruf eingestellt wurden.

Gehen Sie davon aus, dass die Antwort jetzt zusätzlich zur vorherigen Anzeige mit der ID 9935407 eine weitere Anzeige mit der ID 9235407 enthält. Die Beacons der Anzeigen-ID 9235407 haben die Werte 132,41 und 70.339. StartTimeInSeconds

MediaTailor durchläuft alle Beacons in der Sitzung, um die Beacons mit StartTimeInSeconds mehr als 52,286 Sekunden auszuwählen, bei denen es sich um Beacon 3 und Beacon 4 aus der Anzeige mit der ID 9235407 handelt:

```

{
  "NextToken": "ZkfknvbfsgdgbfSDFRdffg12EdffecFRvhjyjfhdhfnjtsG5SDGN
  "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
            }
          ],
        }
      ],
    }
  ],

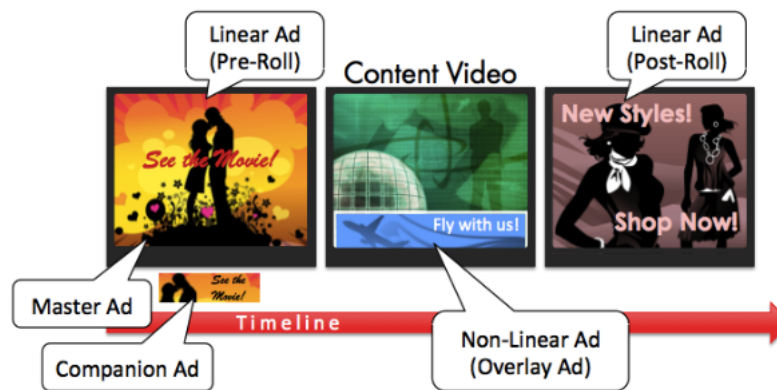
```

```
        "vastAdId": ""
      },
    ],
    "startTime": "PT36.47S",
    "StartTimeInSeconds": 36.47
  }
]
```

Overlay-Anzeigen

Für Live-Streaming-Workflows, bei denen Sie die Monetarisierung erhöhen möchten, ohne das Zuschauererlebnis durch Mid-Roll-Anzeigen zu unterbrechen, können Sie Ihre aktuelle AWS Elemental MediaTailor Integration nutzen, um ein Werbeformat zu entwickeln, das kundenseitig gerendert wird. Diese Art von Werbung wird als Overlay-Werbung bezeichnet. Overlay-Anzeigen sind nichtlineare Videoanzeigen, die in Form von „L-Band-Anzeigen“, „nichtlinearen Videoanzeigen“, „Anzeigen“, „Motion-Overlays“, „picture-in-picture In-Content-Werbung“ oder „Frame-Anzeigen“ geschaltet werden.

MediaTailor erkennt einen SCTE-35-Marker mit Segmentierungstyp als In-Band-Signal für eine Möglichkeit zur Einblendung von Overlay-Werbung. `id=0x38` Die SCTE-35-Markierung veranlasst MediaTailor das Senden einer Anfrage an den Ad Decision Server (ADS), der dann mit einer nichtlinearen Anzeigennutzlast in der VAST-Antwort antwortet. MediaTailor analysiert die VAST-Antwort, um das Einfügen von Overlay-Anzeigen zu unterstützen. MediaTailor führt kein Zusammenfügen linearer Anzeigen durch, sondern signalisiert dem Player, dass eine nichtlineare Overlay-Anzeige zum Abspielen verfügbar ist. Diese Signalisierung ermöglicht es dem Player, die nichtlinearen Anzeigen für die Wiedergabe vom clientseitigen Tracking-Endpunkt abzurufen und zu korrelieren. Der Player kümmert sich dann um die Anzeige, die Berichterstattung und andere Aufgaben im Zusammenhang mit diesen Anzeigen. Beispielsweise kann der Entwickler des Players ein Geräte-SDK eines Anbieters verwenden, das Overlay-Anzeigenformate unterstützt. Weitere Informationen zu clientseitigen Tracking-Integrationen finden Sie unter [Kundenseitige Ad-Tracking-Integrationen](#)



Themen

- [Voraussetzungen für die Verwendung von Overlay-Anzeigen mit MediaTailor](#)
- [Erste Schritte mit Overlay-Anzeigen mit MediaTailor](#)
- [Protokollierung und Metriken für Overlay-Anzeigen in MediaTailor](#)
- [Abrechnung für Overlay-Anzeigen in MediaTailor](#)

Voraussetzungen für die Verwendung von Overlay-Anzeigen mit MediaTailor

Die folgenden Voraussetzungen gelten für die Verwendung von Overlay-Anzeigen mit: MediaTailor

- Der Workflow muss live sein, nicht Video-on-Demand (VOD).
- Die Antwort des Ad Decision Servers (ADS) muss so konfiguriert sein, dass in der VAST-Antwort nur nichtlineare Anzeigen zurückgegeben werden. MediaTailor ignoriert alle linearen Anzeigen für das Ad-Stitching.
- Das Manifest muss eine SCTE-35-Zeitsignalnachricht mit Segmentierungstyp verwenden, `id=0x38` um die Overlay-Ad-Funktion aufzurufen.
- Der Streaming-Anbieter muss die Kontrolle über die Anwendung auf dem Client-Gerät haben und in die clientseitige Tracking-API integriert sein. MediaTailor

Erste Schritte mit Overlay-Anzeigen mit MediaTailor

In diesem Abschnitt wird erläutert, wie Sie mit der Verwendung der Overlay-Anzeigenfunktion von beginnen. MediaTailor Sie richten die SCTE-35-Signalisierung ein, konfigurieren die Antworten des Ad Decision Servers (ADS) und richten die Steuerung auf Sitzungsebene ein.

Themen

- [Overlay-Anzeigen aktivieren](#)
- [Verfolgung von Overlay-Anzeigen mit clientseitigen Metadaten](#)

Overlay-Anzeigen aktivieren

MediaTailor Die Unterstützung für Overlay-Anzeigen ist standardmäßig aktiviert. Ein bestimmter SCTE-35-Werbemarkertyp im Manifest löst das Einfügen einer Overlay-Anzeige aus. Da einige Player das clientseitige Rendern von Overlay-Anzeigen möglicherweise nicht unterstützen, können Sie die Funktion auf Sitzungsebene deaktivieren.

So deaktivieren Sie die Unterstützung von Overlay-Anzeigen mithilfe von HLS- oder DASH-Wiedergabepräfixen:

- Initialisieren Sie vom Player aus eine neue MediaTailor Wiedergabesitzung mit einer Anfrage in einem der folgenden Formate, je nach Protokoll:
 - Beispiel: HLS-Format

```
GET mediatailorURL/v1/master/hashed-account-id/origin-id/asset-id?  
aws.overlayAvails=off
```

- Beispiel: DASH-Format

```
GET mediatailorURL/v1/master/hashed-account-id/origin-id/asset-id?  
aws.overlayAvails=off
```

So deaktivieren Sie die Unterstützung für Overlay-Ads mithilfe des Sitzungsinitialisierungspräfix:

- Konstruieren Sie auf dem Player einen JSON-Nachrichtentext für die Anfrage zur Sitzungsinitialisierung, um: MediaTailor

- Um die Ad-Overlay-Unterstützung zu deaktivieren, fügen Sie ein `overlays` Objekt als Schlüssel der obersten Ebene mit einem Wert von `hinzu. off`. Der Standardwert `overlays` ist `on`.
- (Optional) Geben Sie alle Parameter an, die MediaTailor dann innerhalb eines `adsParams` Objekts an das ADS übergeben werden. Diese Parameter entsprechen `[player_params.param]`-Einstellungen in der ADS-Vorlagen-URL der MediaTailor-Konfiguration.

Example HLS:

```
POST master.m3u8
{
  "adsParams": {
    "deviceType": "ipad"
  },
  "overlayAvails": "off"
}
```

Example DASH:

```
POST manifest.mpd
{
  "adsParams": {
    "deviceType": "androidmobile"
  },
  "overlayAvails": "off"
}
```

Manifeste Signalisierung

MediaTailor löst die Unterstützung für Overlay-Ads aus, wenn ein bestimmter SCTE-35-Marker im Manifest erkannt wird. Das erforderliche Signal ist ein Splice-Befehl vom Typ 6 oder ein Zeitsignal, bei dem es sich um ein Startsignal für Provider Overlay Advertisement handelt. Dieses Signal hat die Segmentierungstyp-ID von `0x38`.

Das folgende Beispiel zeigt den `0x38` SCTE-35-Marker in einem JSON-Objekt.

```
{
  "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
}

```

Das folgende Beispiel zeigt, dass das SCTE-35-Signal als binärer Wert (Basis 32/Hexadezimal) dargestellt wird:

```
0xfc303500000000000000ff00506fe001b78c8001f021d435545490970d4717fd0000dbba009076f7665726c617
```

Das folgende Beispiel zeigt den SCTE-35-Marker sowohl in HLS- als auch in DASH-Manifesten.

Example: HLS-Manifest

```
#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=0xfc30350000000000000000ffff00506fe001b78c8001f021d435545490970d4717fdf00000dbba009076f7665726
#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-Manifest

```

<?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>

```

Antwort des Ad Decision Servers (ADS)

Die ADS-Antwort muss ein gültiges Tracking-Ereignis enthalten. Bei dem Tracking-Ereignis kann es sich mindestens um ein Impression Tracking-Ereignis handeln. Das Tracking-Event sollte mindestens eine NonLinear Anzeige enthalten. Bei dieser Anzeige handelt es sich um eine Overlay-Anzeige in Form einer statischen, HTML- oder iFrame-Ressource.

```
<vmap AdBreak breaktype="linear" breakId="csoverlay"
```

Wenn es sich bei der VAST-Antwort um eine VMAP mit `breakType of handeltnonlinear`, befinden sich die verfügbaren Metadaten im Stammobjekt. `nonLinearAvails` Wenn es sich bei der VAST-Antwort um eine VMAP mit einem `breakType of linear` oder um eine einfache VAST-Antwort ohne VMAP handelt, befinden sich die Avail-Metadaten innerhalb des Stammobjekts. `avails`

Die folgende VAST-Antwort ist eine umschlossene VMAP-Antwort mit dem Wert. `breakType linear`

Unterstützt neben der umschlossenen VMAP-Antwort MediaTailor auch eine umschlossene VMAP-Antwort mit einem `breakType` Wert von `nonlinear` und eine einfache VAST-Antwort.

```

<?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: Quelle des DASH-Manifests für MediaTailor

```

<?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: MediaTailor personalisiertes DASH-Manifest, das eine Werbe-ID-Dekoration enthält

```

<?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:smppte:ul:060E2B34.01040101.01200900.000000000", "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>

```

Verfolgung von Overlay-Anzeigen mit clientseitigen Metadaten

MediaTailor platziert die Overlay-Anzeigen ohne Erfolg `nonLinearAdsList`. Die MediaTailor clientseitige Tracking-API hat zwei Stammobjekte, die als `avails` und `nonLinearAvails` bezeichnet werden. Wenn es sich bei der VAST-Antwort um eine VMAP mit `breakType` of `nonLinear` handelt, befinden sich die verfügbaren Metadaten innerhalb des Stammobjekts `nonLinearAvails`. Wenn es sich bei der VAST-Antwort um eine VMAP mit einem `breakType` of `linear` oder um eine einfache VAST-Antwort ohne VMAP handelt, befinden sich die Avail-Metadaten innerhalb des Stammobjekts `avails`.

Weitere Informationen zur clientseitigen Nachverfolgung finden Sie unter [Kundenseitiges Anzeigen-Tracking](#).

Das folgende Beispiel zeigt eine einfache VAST-Antwort oder VMAP-Antwort mit dem Wert `breakType 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": []
}

```

Das folgende Beispiel zeigt eine einfache VMAP-Antwort mit dem breakType Wert. 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  
}  
]  
}
```

Protokollierung und Metriken für Overlay-Anzeigen in MediaTailor

In diesem Abschnitt werden die Protokollierung und die Messwerte für Overlay-Anzeigen erläutert. MediaTailor Weitere Informationen zum Einrichten der Protokollierung finden Sie unter [Ressourcen überwachen und kennzeichnen AWS Elemental MediaTailor](#).

Themen

- [CloudWatch Logs](#)
- [CloudWatch Metriken](#)

CloudWatch Logs

CloudWatch sammelt die folgenden Protokollinformationen zu Overlay-Anzeigen:

- VAST_RESPONSE- Zeigt Informationen zur Liste der nichtlinearen Anzeigen an.
- FILLED_PROVIDER_OVERLAY- Zeigt Informationen zu den nichtlinearen Anzeigen an.

Note

Das RAW_ADS_RESPONSE ist ein optionales Ereignis, das die ursprüngliche Antwort des ADS anzeigt. Die Verwendung dieses Ereignisses ist besonders in einer Staging- und

Testumgebung hilfreich. Um dieses Ereignis für eine Konfiguration oder ein Konto zu aktivieren, reichen Sie ein Ticket an den AWS Support ein.

CloudWatch Metriken

MediaTailor erfasst Messwerte für Overlay-Anzeigen getrennt von anderen ADS-Metriken. MediaTailor sammelt diese Messwerte, nachdem die Anzeigen erfolgreich aus dem ADS abgerufen wurden. Sie müssen die `GetTracking` API nicht abfragen, um die Metriken zu sammeln.

In der folgenden Tabelle werden CloudWatch Metriken für Overlay-Anzeigen beschrieben:

Metrik	Description
<code>AdDecisionServer.OverlayAds</code>	Die Anzahl der Overlay-Anzeigen, die innerhalb des von Ihnen angegebenen CloudWatch Zeitraums in den ADS-Antworten enthalten waren.
<code>AdDecisionServer.OverlayErrors</code>	Die Anzahl der Antworten ohne 200 HTTP-Statuscode, der leeren Antworten und der Antworten mit Timeout, die innerhalb des von Ihnen angegebenen CloudWatch Zeitraums vom ADS MediaTailor empfangen wurden.
<code>AdDecisionServer.OverlayFilled</code>	Die Anzahl der Avails, die erfolgreich mit mindestens einer Overlay-Anzeige gefüllt wurden: • 1 — Es gibt mindestens eine gültige Anzeige. • 0 — Entweder wurden MediaTailor keine Overlay-Anzeigen angezeigt, oder es ist ein anderer Fehler aufgetreten. <code>SampleCount</code> verfolgt die Anzahl der ausgefüllten Avails.

Metrik	Description
	Sumverfolgt die Anzahl der erfolgreich ausgefüllten Overlay-Avails.
<code>AdDecisionServer.OverlayMinSuggestedDuration</code>	Die Summe der <code>minSuggestedDuration</code> Dauer aller Anzeigen in Millisekunden, die innerhalb des von Ihnen angegebenen MediaTailor Zeitraums vom ADS empfangen wurden. CloudWatch Wenn <code>minSuggestedDuration</code> nicht angegeben, entspricht die angezeigte Dauer der geplanten Dauer.
<code>AdDecisionServer.OverlayLatency</code>	Die Antwortzeit in Millisekunden für Anfragen an das MediaTailor ADS.
<code>AdDecisionServer.OverlayTimeouts</code>	Die Anzahl der Anfragen an das ADS mit Timeout in dem von Ihnen angegebenen CloudWatch Zeitraum.
<code>AdsBilled</code>	Weitere Informationen zu fakturierten Anzeigen finden Sie unter Abrechnung für Overlay-Anzeigen in MediaTailor
<code>Avail.*</code>	Weil MediaTailor keine Planung für Overlay-Anzeigen durchgeführt wird und CloudWatch keine Messwerte angezeigt werden <code>Avail.X</code> .
<code>SkippedReason.*</code>	Weil MediaTailor es keine Planung für Overlay-Anzeigen durchführt, CloudWatch keine Metriken anzeigt. <code>SkippedReason.X</code>

Abrechnung für Overlay-Anzeigen in MediaTailor

MediaTailor stellt Kunden auf der Grundlage der Anzahl der nichtlinearen Anzeigen in der ADS-Antwort eine Rechnung. In dieser Zahl sind auch nichtlineare Anzeigen enthalten, die über die Dauer der Pause hinausgehen. Sobald der MediaTailor Vorrat aufgefüllt ist, werden die ausgefüllten Anzeigen in Rechnung gestellt.

Bei Prefetch-Workflows werden Anzeigen MediaTailor nicht beim Abrufen des Prefetchs in Rechnung gestellt, sondern erst, wenn im Nutzungsfenster für diese Sitzung eine kompatible Anzeige verfügbar ist.

Weitere Abrechnungsinformationen finden Sie unter <https://aws.amazon.com/mediatailor/pricing/>

Werbe-ID-Dekoration

AWS Elemental MediaTailor führt beim Übergang von Inhalten zu Werbeunterbrechungen serverseitiges Ad-Stitching durch. MediaTailor kann das Manifest mit Metadaten zu den zusammengestellten Anzeigen konditionieren. Dies kann die folgenden Vorteile bieten:

Important

Die Dekoration von Ad-IDs ist nicht mit Methoden zur serverseitigen Anzeigeneinfügung (SGAI) kompatibel, da die Felder, mit denen X-AD-CREATIVE-SIGNALING Kopfzeilen gefüllt werden, nur bekannt sind, wenn die Asset-Liste während der Anzeigenwiedergabe abgerufen wird, und nicht, wenn das zwischenspeicherbare Manifest zum ersten Mal geschrieben wird.

- Die Videostartzeit (VST) verbessert sich
- MediaTailor kann ein Hybridmodell aus serverseitiger und clientseitiger Anzeigeneinfügung unterstützen
- In serverseitigen Sitzungen können Zeitlinien für die Wiedergabe mithilfe von Positionsmarkierungen für Anzeigen erstellt werden
- Für clientseitige Sitzungen, für die bereits Wiedergabezeitpläne mit der MediaTailor API erstellt wurden, verbessert sich das Sitzungs-VST, da die Sitzung nicht auf den Aufruf der Tracking-API angewiesen ist, um die Zeitleiste zu erstellen
- Es ist möglich, sowohl serverseitige Anzeigen MediaTailor einzufügen als auch clientseitig gerenderte Anzeigen, die in der Szene angezeigt werden, zu nutzen. Auf diese Weise benötigt das Software Development Kit (SDK) eines Players keine separate Integration, um Ad-Serving-Entitäten direkt für clientseitige Werbung aufzurufen. MediaTailor kann die Anzeigen über das Manifest und die clientseitige Tracking-API verkaufen.

Es gibt Standards, nach denen jedem kreativen Anzeigen-Asset eine eindeutige Kennung zugewiesen wird. Diese Zuordnung ermöglicht es Werbetreibenden, Agenturen, Anbietern und

Publishern, ein kreatives Anzeigen-Asset in ihren unabhängigen Workflows miteinander zu verknüpfen. Da sich die Kennzahlen und die Überwachung von Streams immer weiter verbessern und immer mehr Vertriebspartner serverbasierte Insertion-Architekturen einsetzen, besteht die Notwendigkeit, die Identifikatoren, die einzelnen kreativen Assets innerhalb einer interleaved/stitched Präsentation zugewiesen wurden, genau zu kommunizieren, beispielsweise innerhalb des personalisierten Manifests.

Themen

- [Aktivierung der Signalisierung von Werbe-IDs für Sitzungen](#)
- [Manifeste und Einfügen von Anzeigenmetadaten](#)
- [Interaktionen mit dem Ad Decision Server \(ADS\)](#)
- [Client-seitige Tracking-API](#)

Aktivierung der Signalisierung von Werbe-IDs für Sitzungen

Die Signalisierungsfunktion für die Anzeigen-ID muss während der Sitzungsinitialisierung aktiviert sein. Das Verfahren zur Aktivierung der Funktion unterscheidet sich von der Erstellung von Sitzungen mit dem HLS/DASH Wiedergabepräfix (implizite Sitzungsinitialisierung) und dem Sitzungsinitialisierungspräfix (explizite Sitzungsinitialisierung).

So aktivieren Sie die Anzeigen-ID für die Sitzung mithilfe von Wiedergabepräfixen HLS/DASH

- Initialisieren Sie vom Player aus eine neue MediaTailor Wiedergabesitzung mithilfe einer Anfrage in einem der folgenden Formate gemäß Ihrem Protokoll:
 - Beispiel: HLS-Format

```
GET <mediatailorURL>/v1/master/<hashed-account-id>/<origin-id>/<asset-id>?  
aws.adSignalingEnabled=true
```

- Beispiel: DASH-Format

```
GET <mediatailorURL>/v1/dash/<hashed-account-id>/<origin-id>/<asset-id>?  
aws.adSignalingEnabled=true
```


So aktivieren Sie die Anzeigen-ID für die Sitzung mithilfe des Sitzungsinitialisierungspräfixes

- Erstellen Sie auf dem Player einen JSON-Nachrichtentext für die Anfrage zur Sitzungsinitialisierung, um: MediaTailor
 - Geben Sie innerhalb eines `adsParams` Objekts alle Parameter an, die an das ADS übergeben werden MediaTailor sollen. Diese Parameter entsprechen den `[player_params.param]` Einstellungen in der ADS-Vorlagen-URL der MediaTailor Konfiguration.
 - Um die Signalisierung der Anzeigen-ID zu aktivieren, fügen Sie ein `adSignaling` Objekt als Objekt der obersten Ebene hinzu und fügen Sie darin einen Parameter mit dem Namen `enabled` und dem Wert von `true` hinzu. Der `adSignaling` Standardwert ist `disabled`.
- Beispiel: HLS-Format

```
POST master.m3u8
{
  "adsParams": {
    "deviceType": "ipad"
  },
  "adSignaling": {
    "enabled": "true"
  },
  "reportingMode": "client"
}
```

- Beispiel: DASH-Format

```
POST manifest.mpd
{
  "adsParams": {
    "deviceType": "ipad"
  },
  "adSignaling": {
    "enabled": "true"
  },
  "reportingMode": "client"
}
```

Manifeste und Einfügen von Anzeigenmetadaten

MediaTailor fügt dem Manifest beim Zusammenfügen von Anzeigen die eindeutige ID hinzu, die jedem zusammengestellten Werbemittel zugeordnet ist. MediaTailor ermittelt die eindeutige ID des Werbemittels anhand des `id` Attributwerts dieses Werbemittels in der VAST-Antwort. Wenn dem Creative ein ID-Attributwert fehlt, veröffentlicht MediaTailor es einen leeren Wert (`id=""`).

MediaTailor verwendet ein In-Manifest-Metadaten-Signal, um Abhängigkeiten zwischen der Client-Tracking-API für Anzeigenkreativ-Metadaten und timing/positioning innerhalb des Gesamtzeitplans voneinander zu trennen. Diese Entkopplung reduziert die Wiedergabe-Latenz (insbesondere in VOD-Szenarien), bei denen die Benutzeroberfläche (UI) des Players die Positionen der Werbeunterbrechungen in der Timeline rendert, bevor die Wiedergabe initialisiert wird.

Die hinzugefügten Metadaten haben die folgenden Formen:

- Bei HLS-Manifesten haben die hinzugefügten Metadaten die Form von DATERANGE Tags für jede Anzeige im Verfügbarkeitszeitraum.
- Bei DASH-Manifesten haben die hinzugefügten Metadaten die Form eines Event Elements innerhalb jedes Anzeigenzeitraums.

Der folgende JSON-Nachrichtentext zeigt ein Beispiel für eine VAST-Antwort:

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

Für das obige Beispiel gilt:

- *creativeId* ist der ID Attributwert des Creative Elements für die Anzeige

- **adId** ist entweder die HLS-Sequenznummer, die dem Beginn der Anzeige zugeordnet ist, oder die DASH-Perioden-ID der Anzeige
- **adType** ist entweder `avail` oder `overlay`, basierend auf der VAST-Antwort
- **trackingUri** ist der relative Tracking-Endpunkt für die MediaTailor Sitzung im folgenden Format `../../../tracking/hashed-account-id/origin-id/session-id`
- **customVastData** ist ein Wert, der aus der `creative_signaling` VAST-Erweiterung MediaTailor extrahiert wird. MediaTailor verwendet den Inhalt des CDATA-Knotens, falls vorhanden. Weitere Informationen und ein Beispiel für eine VAST-Antwort finden Sie im [Interaktionen mit dem Ad Decision Server \(ADS\)](#) Abschnitt.

Personalisierung von HLS-Manifesten mit Anzeigenmetadaten

Fügt bei einem HLS-Live-Stream MediaTailor nur Metadaten hinzu, wenn der Stream PROGRAM-DATA-TIME Tags enthält, und zwar mindestens einmal pro Manifestdauer. MediaTailor Fügt PROGRAM-DATE-TIME bei einem Video-on-Demand-Stream (VOD) mindestens ein Segment im personalisierten Manifest hinzu, wobei die Startzeit für jedes VOD-Asset Epoche Null () ist. 1970-01-01T00:00:00Z Wenn das Ursprungsmanifest bereits PROGRAM-DATE-TIME Inhalt enthält, wird dieser Inhalt beibehalten MediaTailor .

MediaTailor personalisiert das Manifest mit den vom Ad Decision Server (ADS) zurückgegebenen Creatives. Enthält für jede Anzeige MediaTailor außerdem ein DATERANGE Tag, das sich über die gesamte Laufzeit der Anzeige erstreckt. Das DATERANGE Tag-Format ähnelt dem Format, das im Abschnitt [Ad Creative Signaling in DASH und HLS](#) in der Version 2023 der technischen Publikation von SVA beschrieben wird.

Das DATERANGE, was MediaTailor generiert wird, hat eindeutige ID-Werte. Um die Eindeutigkeit zu gewährleisten (unter Berücksichtigung der in [Mapping SCTE-35 into](#) angegebenen Richtlinien EXT-X-DATERANGE), MediaTailor wird die MEDIA-SEQUENCE Nummer des ersten Anzeigensegments der Avail mit der Sequenznummer der Anzeige innerhalb des Avails verknüpft.

Bei unzureichend ausgefüllten Werbeunterbrechungen in Konfigurationen, bei denen Slate aktiviert ist, werden die Slate-Segmente an das Ende des Avails MediaTailor angehängt, getrennt durch ein Tag, aber ohne Metadaten. DISCONTINUITY DATERANGE

Fügt für jede Anzeige, die in das personalisierte Manifest eingebunden wird, die kreativen Metadaten MediaTailor hinzu, die als Base64-kodierte Daten in einem benutzerdefinierten Tag dargestellt werden. DATERANGE

Example Linearer HLS-Ursprung (): #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 Linearer HLS-Ursprung ()#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 Lineares personalisiertes HLS-Manifest (mit kreativer Anzeigensignalisierung):

DasDATERANGE, was MediaTailor generiert wird, hat eindeutige ID-Werte. Um die Eindeutigkeit zu gewährleisten (unter Berücksichtigung der in [Mapping SCTE-35 into](#) angegebenen Richtlinien EXT-X-DATERANGE), MediaTailor wird die MEDIA-SEQUENCE Nummer des ersten Anzeigensegments der Avail mit der Sequenznummer der Anzeige innerhalb des Avails verknüpft.

Im folgenden Beispiel verknüpft 421 mit der MediaTailor Positionsnummer der Anzeige. MEDIA-SEQUENCE

```

#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-Ursprung (mit SCTE-Signalen):

```
#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-Ursprung:

```
#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 Personalisiertes VOD HLS-Manifest:

MediaTailor fügt PROGRAM-DATE-TIME VOD-Manifeste hinzu, um sie als Anker für die DATERANGE HLS-Elemente zu verwenden, die auf Anzeigenpositionen hinweisen.

DasDATERANGE, was MediaTailor generiert wird, hat eindeutige ID-Werte. Um die Eindeutigkeit zu gewährleisten (unter Berücksichtigung der in [Mapping SCTE-35 into](#) angegebenen Richtlinien EXT-X-DATERANGE), MediaTailor wird die MEDIA-SEQUENCE Nummer des ersten Anzeigensegments der Avail mit der Sequenznummer der Anzeige innerhalb des Avails verknüpft.

Im folgenden Beispiel verknüpft 421 mit der MediaTailor Positionsnummer der Anzeige. MEDIA-SEQUENCE

```

#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
```

Personalisierung von DASH-Manifesten mit Anzeigenmetadaten

MediaTailor personalisiert das Manifest mit den vom Ad Decision Server (ADS) zurückgegebenen Creatives. Enthält für jede Anzeige MediaTailor auch ein EventStream Element, das sich über die gesamte Laufzeit der Anzeige erstreckt. Das Event Elementformat ähnelt dem Format, das im Abschnitt [Ad Creative Signaling in DASH und HLS](#) in der Version 2023 der technischen Publikation von SVA beschrieben wird.

Bei unzureichend ausgefüllten Werbeunterbrechungen in Konfigurationen, bei denen Slate aktiviert ist, wird der Slate-Zeitraum an das Ende des Verfügbarkeitszeitraums MediaTailor angehängt, jedoch ohne Metadaten EventStream

Fügt für jede Anzeige, die in das personalisierte Manifest eingebunden wird, die kreativen Metadaten MediaTailor hinzu, die als Element innerhalb eines CDATA Elements dargestellt werden. Event

Example Linearer DASH-Ursprung (Inline-SCTE-Attribute):

```
<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_$RepresentationID$_
      $Number$.mp4?m=1676062374" initialization="index_video_$RepresentationID$_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>
    <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"/>
            <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"/>
        </SegmentTimeline>
    </SegmentTemplate>
    <Representation id="1" width="128" height="64" frameRate="48000/1000"
bandwidth="128000" codecs="mp4a.40.2"/>
</AdaptationSet>
</MP4>

```



```

        <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 Personalisiertes lineares DASH-Manifest (mit kreativer Anzeigensignalisierung):

```

<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/
f9f38deca3f14fc4b5ab3cbbd76cfb9e/</BaseURL>

```

```

    <Location>https://777788889999.mediatailor.us-west-2.amazonaws.com/
v1/dash/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/out/v1/
f9f38deca3f14fc4b5ab3cddb76cfb9e/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"/>
            <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:smp: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.mediataylor.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:smppte: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>
    <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>

```

Interaktionen mit dem Ad Decision Server (ADS)

MediaTailor verwendet den Wert des kreativen `id` Attributs aus der VAST-Antwort als Wert für die Signalisierung der Anzeigen-ID. Wenn der `id` Attributwert leer oder in der VAST-Antwort nicht vorhanden ist, MediaTailor wird ein leerer Wert in die Signalisierung der Anzeigen-ID eingefügt.

Example VAST-Antwort:

Die folgende VAST-Beispielantwort enthält einen `id` Attributwert für die lineare Inline-AntwortCreative. MediaTailor extrahiert den Wert aus dem benutzerdefinierten Extension VAST-Element und fügt diesen Wert in die kreativen Metadaten des Manifests ein.

```

<?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>
```

Client-seitige Tracking-API

Das folgende Beispiel zeigt, wie ein Player-SDK die Anzeigenmetadaten im Manifest mit den vollständigen Tracking-Eventdaten in der clientseitigen Tracking-Antwort-Payload mit und verknüpft. `creativeId` `adId`

Example JSON-Nachricht:

```
{
  "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": "UFQ1TTM0Ljk2N1NfMjAyMi0xMS0xOFQwNDZMzo1Mi4yNDUxOTdaXzE%3D",
"nonLinearAvails": []
}
```

Wird verwendet AWS Elemental MediaTailor , um linear zusammengestellte Streams zu erstellen

AWS Elemental MediaTailor Channel Assembly ist ein reiner Dienst, mit dem Sie lineare Streaming-Kanäle erstellen können, indem Sie Ihre vorhandenen Video-on-Demand-Inhalte (VOD) mit Live-Inhalten mischen. MediaTailor berührt niemals Ihre Inhaltssegmente, die direkt von Ihrem Ursprungsserver bereitgestellt werden. MediaTailor Ruft stattdessen die Manifeste von Ihrem Ursprung ab und verwendet sie, um ein Live-Slide-Manifestfenster zusammenzustellen, das auf die zugrunde liegenden Inhaltssegmente verweist. Bei der Kanalzusammenstellung werden Dinge wie die Mediensequenznummer gespeichert, die für eine reibungslose Wiedergabe von Asset zu Asset erforderlich ist. Linear zusammengestellte Streams werden mit geringen Betriebskosten erstellt, indem vorhandene, mit mehreren Bitraten kodierte und verpackte VOD-Inhalte verwendet werden.

Sie können lineare Channel-Assembly-Streams ganz einfach monetarisieren, indem Sie Werbeunterbrechungen in Ihre Programme einfügen, ohne den Inhalt mit SCTE-35-Markern konditionieren zu müssen. Sie können Channel Assembly zusammen mit dem Dienst zur MediaTailor Anzeigeneinfügung oder einem beliebigen serverseitigen Dienst zum Einfügen von Anzeigen verwenden.

Informationen zu den ersten Schritten mit der Kanalmontage finden Sie unter [the section called “Erste Schritte mit der MediaTailor Kanalmontage”](#).

Informationen zur Integration von Channel Assembly in ein Content Delivery Network (CDN) finden Sie unter [Erstellen Sie MediaTailor lineare Kanäle mit Kanalmontage und CDN](#).

Themen

- [Mit Quellstandorten arbeiten](#)
- [Mit Kanälen arbeiten](#)
- [Ein Programm zum Zeitplan eines Kanals hinzufügen](#)
- [Fügen Sie personalisierte Anzeigen und Werbeunterbrechungen in einen Channel-Stream ein](#)
- [Zeitversetzung bei der Wiedergabe eines Kanals](#)

Mit Quellstandorten arbeiten

Ein Quellspeicherort stellt den Ursprungsserver dar, auf dem Ihre Quellinhalte gespeichert sind. Ein Quellstandort kann Amazon S3, ein Standard-Webserver, ein Content Delivery Network (CDN)

wie Amazon CloudFront oder ein Verpackungshersteller wie AWS Elemental MediaPackage sein. MediaTailor ruft Ihre Inhaltsmanifeste vom Quellspeicherort ab und verwendet sie, um den linearen Stream Ihres Kanals zusammenzustellen.

In diesem Thema wird erklärt, wie Sie mit der AWS Elemental MediaTailor Konsole Quellverzeichnisse erstellen und löschen und wie Sie mit VOD-Quellen arbeiten.

Themen

- [Einen Quellspeicherort erstellen](#)
- [Konfiguration der Authentifizierung für Ihren Quellstandort](#)
- [Mit VOD-Quellen arbeiten](#)
- [Arbeiten mit Live-Quellen](#)
- [Verwenden von Paketkonfigurationen](#)
- [Zwischenspeichern von Manifesten](#)

Einen Quellspeicherort erstellen

Im folgenden Verfahren wird erklärt, wie Sie mithilfe der MediaTailor Konsole einen Quellspeicherort erstellen. Informationen zum Erstellen von Quellspeicherorten mithilfe der MediaTailor API finden Sie [CreateSourceLocation](#) in der AWS Elemental MediaTailor API-Referenz.

So erstellen Sie einen Quellspeicherort

1. Öffnen Sie die MediaTailor Konsole unter <https://console.aws.amazon.com/mediatailor/>.
2. Wählen Sie im Navigationsbereich Channel-Assembly > Source Locations aus.
3. Wählen Sie in der Navigationsleiste die Option Quellspeicherort erstellen aus.
4. Geben Sie unter Konfiguration des Quellverzeichnisses einen Namen und die Basis-URL Ihres Ursprungsservers ein:
 - Name: Eine Kennung für Ihren Quellstandort, z. B. my-origin.
 - Basis-URL: Das Protokoll und die Basis-URL des Ursprungsservers, auf dem Ihre Inhalte gespeichert sind, z. B. <https://111111111111.cloudfront.net>. Die URL muss in einem Standard-HTTP-URL-Format mit dem Präfix <http://> oder <https://> vorliegen.

Wählen Sie optional SigV4 für die Amazon S3 S3-Authentifizierung verwenden, wenn Ihr Quellstandort ein Amazon S3 S3-Bucket ist und Sie AWS Signature Version 4 für die Amazon

S3 S3-Zugriffsauthentifizierung verwenden möchten. Weiterführende Informationen finden Sie unter [Konfiguration der Authentifizierung für Ihren Quellstandort](#).

5.

Konfigurieren Sie unter Zugriffskonfiguration optional die Authentifizierung für Ihren Quellstandort:

- Zugriffstyp: Wählen Sie den Authentifizierungstyp aus, der für den Zugriff auf den Inhalt MediaTailor verwendet wird, der am Quellspeicherort gespeichert ist.
- SigV4 für Amazon S3 — MediaTailor verwendet Amazon Signature Version 4 (SigV4), um Anfragen an Ihren Absender zu autorisieren. Weitere Informationen finden Sie unter [the section called “Authentifizieren von Anfragen an Amazon S3 mit SigV4”](#).
- Secrets Manager-Zugriffstoken-Authentifizierung — MediaTailor verwendet Secrets Manager und einen vom AWS KMS Kunden verwalteten Schlüssel, der von Ihnen erstellt wurde, in Ihrem Besitz ist und verwaltet wird, um die Zugriffstoken-Authentifizierung zwischen MediaTailor und Ihrem Ursprung zu erleichtern. Informationen zur Konfiguration der Secrets Manager Manager-Zugriffstoken-Authentifizierung finden Sie unter [the section called “AWS Secrets Manager Arbeitet mit Zugriffstoken-Authentifizierung”](#).
- Header-Name — Geben Sie einen HTTP-Header-Namen an. MediaTailor verwendet den HTTP-Header, um das Zugriffstoken in Inhaltsmanifestanfragen an Ihren Ursprung zu senden. Sie können jeden Header-Namen verwenden, solange er nicht mit x-amz- oder beginnntx-amzn-. Wenn Sie eine Integration mit [MediaPackage CDN-Autorisierung durchführen](#), sollte der Header-Wert wie folgt lautenX-MediaPackage-CDNIdentifizier.
- Geheimer Zeichenkettenschlüssel — Der SecretString Schlüssel, den Sie in Ihrem Secrets Manager Manager-Geheimnis angegeben haben. Wenn Ihr beispielsweise ein Schlüssel- und Wertepaar wie: SecretString enthält{"MyHeaderName": "11111111-2222-3333-4444-111122223333"}, dann MyHeaderName ist es der SecretString Schlüssel, den Sie in dieses Feld eingeben.
- Geheimer ARN — Der ARN des Geheimnisses, das Ihr Zugriffstoken enthält. Eine step-by-step Anleitung finden Sie unter [Schritt 2: Erstellen Sie ein AWS Secrets Manager Geheimnis](#).

6. Konfigurieren Sie unter Konfiguration des Segmentbereitstellungsservers optional einen Server für die Bereitstellung Ihrer Inhaltssegmente:

- Verwenden Sie einen Standardserver für die Segmentzustellung: Geben Sie die Basis-URL des Servers ein, der für die Bereitstellung Ihrer Inhaltssegmente verwendet wird, z. B. ein CDN. Konfigurieren Sie den Standard-Hostnamen für das Segment, wenn Sie für die

Bereitstellung der Inhaltssegmente einen anderen Server als den Quellserver verwenden möchten. Sie können beispielsweise den Zugriff von Spielern auf die Ursprungsmanifeste einschränken, indem Sie eine andere CDN-Konfiguration für die Basis-HTTP-URL (die MediaTailor verwendet, um auf die Manifeste zuzugreifen) und die Standardsegment-Basis-URL (die der Spieler für den Zugriff auf die Inhaltssegmente verwendet) verwenden. Wenn Sie keinen Wert eingeben, wird MediaTailor standardmäßig der Quellserver für die Segmentzustellung verwendet.

- Verwenden Sie benannte Segmentzustellungsserver: Wenn Sie einen Standardserver für die Segmentzustellung konfiguriert haben, können Sie auch zusätzliche Server für die Segmentzustellung konfigurieren. Jeder muss einen eindeutigen Namen und eine Basis-URL haben. Die Basis-URL kann eine vollständige HTTP-URL sein oder ein relativer Pfad wie `/some/path/`. Die Namen werden verwendet, um zu identifizieren, welcher Server verwendet werden soll, wenn MediaTailor er eine Anfrage für Inhaltssegmente erhält. Wenn die Anfrage den Header enthält `X-MediaTailor-SegmentDeliveryConfigurationName` und der Wert des Headers mit einem Namen übereinstimmt, wird die entsprechende Basis-URL verwendet, um den Inhalt bereitzustellen. Wenn der Header nicht in der Anfrage enthalten ist oder wenn er mit keinem Namen übereinstimmt, wird der Standardserver für die Segmentzustellung verwendet.

7. Wählen Sie Quellspeicherort erstellen aus.

8. Um weitere Quellstandorte hinzuzufügen, wiederholen Sie die Schritte 2-6.

Konfiguration der Authentifizierung für Ihren Quellstandort

Verwenden Sie die Zugriffskonfiguration, um die Authentifizierung für Ihren Quellstandort zu konfigurieren. Wenn die Zugriffskonfiguration aktiviert ist, werden Quellmanifeste von Ihrem Ursprung MediaTailor nur abgerufen, wenn die Anfrage zwischen MediaTailor und Ihrem Ursprung autorisiert wurde. Die Zugriffskonfiguration ist standardmäßig ausgeschaltet.

MediaTailor unterstützt die folgenden Authentifizierungstypen:

- SigV4 für Amazon S3 S3-Authentifizierung
- AWS Secrets Manager Zugriffstoken
- SigV4 für die MediaPackage Authentifizierung der Version 2 (v2)

In diesem Kapitel wird erklärt, wie SigV4 für Amazon S3, MediaPackage v2 und AWS Secrets Manager Zugriffstoken für die Quellstandortauthentifizierung verwendet werden.

Für weitere Informationen wählen Sie das entsprechende Thema aus.

Themen

- [Authentifizieren von Anfragen an Amazon S3 mit SigV4](#)
- [Arbeiten mit SigV4 für Version 2 MediaPackage](#)
- [AWS Secrets Manager Arbeitet mit Zugriffstoken-Authentifizierung](#)

Authentifizieren von Anfragen an Amazon S3 mit SigV4

Signature Version 4 (Sigv4) für Amazon S3 ist ein Signaturprotokoll, das zur Authentifizierung von Anfragen an Amazon S3 über HTTPS verwendet wird. Wenn Sie SigV4 für Amazon S3 verwenden, MediaTailor fügt es einen signierten Autorisierungsheader in die HTTPS-Anfrage an den Amazon S3 S3-Bucket ein, der als Ihr Ursprung verwendet wird. Wenn der signierte Autorisierungsheader gültig ist, erfüllt Ihr Absender die Anfrage. Wenn er nicht gültig ist, schlägt die Anfrage fehl.

Allgemeine Informationen zu SigV4 für AWS Key Management Service finden Sie im Thema [Authentifizieren von Anfragen \(AWS Signature Version 4\)](#) in der Amazon S3 S3-API-Referenz.

Note

MediaTailor signiert Anfragen an diese Ursprünge immer mit SigV4.

Voraussetzungen

Wenn Sie die SigV4-Authentifizierung für Amazon S3 für Ihren Quellstandort aktivieren, müssen Sie die folgenden Anforderungen erfüllen:

- Sie müssen MediaTailor den Zugriff auf Ihren Amazon S3 S3-Bucket zulassen, indem Sie [mediatailor.amazonaws.com](#) den Hauptzugriff in IAM gewähren. [Informationen zur Konfiguration des Zugriffs in IAM finden Sie unter Zugriffsverwaltung im Benutzerhandbuch.AWS Identity and Access Management](#)
- Der Service Principal von [mediatailor.amazonaws.com](#) muss berechtigt sein, alle multivarianten Playlisten zu lesen, auf die in den VOD-Quellpaketkonfigurationen verwiesen wird.
- Der Aufrufer der API muss über s3: GetObject IAM-Berechtigungen verfügen, um alle multivarianten Playlisten lesen zu können, auf die in Ihren VOD-Quellpaketkonfigurationen verwiesen wird. MediaTailor

- Ihre Basis-URL für den MediaTailor Quellstandort muss dem Anforderungs-URL-Format von Amazon S3 im virtuellen Hosted-Stil entsprechen. Zum Beispiel `https://bucket-name.s3.Region.amazonaws.com/.key-name` Informationen zum von Amazon S3 gehosteten virtuellen Zugriff finden Sie unter Anfragen im [virtuellen Hosted-Stil](#).

Arbeiten mit SigV4 für Version 2 MediaPackage

Signature Version 4 (Sigv4) für MediaPackage v2 ist ein Signaturprotokoll, das zur Authentifizierung von Anfragen an MediaPackage v2 über HTTP verwendet wird. Wenn Sie SigV4 für MediaPackage v2 verwenden, MediaTailor fügt es einen signierten Autorisierungsheader in die HTTP-Anfrage an den MediaPackage v2-Endpunkt ein, der als Ihr Ursprung verwendet wird. Wenn der signierte Autorisierungsheader gültig ist, erfüllt Ihr Absender die Anfrage. Wenn er nicht gültig ist, schlägt die Anfrage fehl.

Allgemeine Informationen zu Sigv4 für MediaPackage v2 finden Sie im Thema [Authentifizieren von Anfragen \(AWS Signature Version 4\)](#) in der MediaPackage v2-API-Referenz.

Voraussetzungen

Wenn Sie die SigV4-Authentifizierung für MediaPackage Version 2 für Ihren Quellstandort aktivieren, müssen Sie die folgenden Anforderungen erfüllen:

- Sie müssen MediaTailor den Zugriff auf Ihren MediaPackage v2-Endpunkt zulassen, indem Sie `mediatailor.amazonaws.com` in einer Origin-Zugriffsrichtlinie für den Endpunkt den Hauptzugriff gewähren.
- Ihre Basis-URL MediaTailor für Ihren Quellstandort muss ein v2-Endpunkt sein. MediaPackage
- Der Aufrufer der API muss über `mediapackagev2:GetObject` IAM-Berechtigungen verfügen, um alle multivarianten Playlisten lesen zu können, auf die in den Quellpaketkonfigurationen verwiesen wird. MediaTailor

AWS Secrets Manager Arbeitet mit Zugriffstoken-Authentifizierung

MediaTailor unterstützt die Secrets Manager Manager-Zugriffstoken-Authentifizierung. Bei der AWS Secrets Manager Zugriffstoken-Authentifizierung werden ein AWS Key Management Service (AWS KMS) vom Kunden verwalteter Schlüssel und ein AWS Secrets Manager Geheimnis MediaTailor verwendet, das Sie selbst erstellen, besitzen und verwalten, um Anfragen an Ihren Absender zu authentifizieren.

In diesem Abschnitt erklären wir, wie die Secrets Manager Manager-Zugriffstoken-Authentifizierung funktioniert, und geben step-by-step Informationen zur Konfiguration der Secrets Manager Manager-Zugriffstoken-Authentifizierung. Sie können mit der Secrets Manager Manager-Zugriffstoken-Authentifizierung in AWS-Managementkonsole oder programmgesteuert mit arbeiten. AWS APIs

Themen

- [Konfiguration der AWS Secrets Manager Zugriffstoken-Authentifizierung](#)
- [Integration mit MediaPackage Endpunkten, die CDN-Autorisierung verwenden](#)
- [So funktioniert die MediaTailor Secrets Manager Manager-Zugriffstoken-Authentifizierung](#)

Konfiguration der AWS Secrets Manager Zugriffstoken-Authentifizierung

Wenn Sie die AWS Secrets Manager Zugriffstoken-Authentifizierung verwenden möchten, führen Sie die folgenden Schritte aus:

1. Sie [erstellen einen vom AWS Key Management Service Kunden verwalteten Schlüssel](#).
2. Sie [erstellen ein AWS Secrets Manager Geheimnis](#). Das Geheimnis enthält Ihr Zugriffstoken, das in Secrets Manager als verschlüsselter geheimer Wert gespeichert ist. MediaTailor verwendet den vom AWS KMS Kunden verwalteten Schlüssel, um den geheimen Wert zu entschlüsseln.
3. Sie konfigurieren einen AWS Elemental MediaTailor Quellspeicherort für die Verwendung der Secrets Manager Manager-Zugriffstoken-Authentifizierung.

Der folgende Abschnitt enthält step-by-step Anleitungen zur Konfiguration der AWS Secrets Manager Zugriffstoken-Authentifizierung.

Themen

- [Schritt 1: Erstellen Sie einen AWS KMS symmetrischen, vom Kunden verwalteten Schlüssel](#)
- [Schritt 2: Erstellen Sie ein AWS Secrets Manager Geheimnis](#)
- [Schritt 3: Konfigurieren Sie einen MediaTailor Quellstandort mit Zugriffstoken-Authentifizierung](#)

Schritt 1: Erstellen Sie einen AWS KMS symmetrischen, vom Kunden verwalteten Schlüssel

Sie verwenden AWS Secrets Manager , um Ihr Zugriffstoken in Form eines geheimen Codes zu SecretString speichern. Der SecretString wird mithilfe eines AWS KMS symmetrischen, vom Kunden verwalteten Schlüssels verschlüsselt, den Sie erstellen, besitzen und verwalten. MediaTailor

verwendet den symmetrischen, vom Kunden verwalteten Schlüssel, um den Zugriff auf den geheimen Schlüssel zu erleichtern und den geheimen Wert zu verschlüsseln und zu entschlüsseln.

Mit vom Kunden verwalteten Schlüsseln können Sie Aufgaben wie die folgenden ausführen:

- Festlegung und Pflege wichtiger Richtlinien
- Festlegung und Aufrechterhaltung von IAM-Richtlinien und -Zuschüssen
- Aktivieren und Deaktivieren wichtiger Richtlinien
- Rotierendes kryptografisches Schlüsselmateriale
- Hinzufügen von -Tags

Informationen zur Verwendung von Secrets Manager AWS KMS zum Schutz von Geheimnissen finden Sie im Thema [AWS Secrets Manager Anwendungsmöglichkeiten AWS KMS](#) im AWS Key Management Service Entwicklerhandbuch.

Weitere Informationen über kundenverwaltete Schlüssel finden Sie unter [Kundenverwaltete Schlüssel](#) im AWS Key Management Service Developer Guide.

 Note

AWS KMS Für die Nutzung eines vom Kunden verwalteten Schlüssels fallen Gebühren an. Weitere Informationen zur Preisgestaltung finden Sie auf der Seite mit den [AWS Key Management Service Preisen](#).

Sie können einen AWS KMS symmetrischen, vom Kunden verwalteten Schlüssel mithilfe von AWS-Managementkonsole oder programmgesteuert mit dem erstellen. AWS KMS APIs

Einen symmetrischen kundenverwalteten Schlüssel erstellen

Folgen Sie den Schritten zum [Erstellen eines symmetrischen, vom Kunden verwalteten Schlüssels](#) im Entwicklerhandbuch.AWS Key Management Service

Notieren Sie sich den wichtigsten Amazon Resource Name (ARN); Sie benötigen ihn in [Schritt 2: Erstellen Sie ein AWS Secrets Manager Geheimnis](#).

Verschlüsselungskontext

Ein Verschlüsselungskontext ist ein optionaler Satz von Schlüssel-Wert-Paaren, die zusätzliche kontextbezogene Informationen zu den Daten enthalten.

Secrets Manager beinhaltet einen [Verschlüsselungskontext](#) beim Verschlüsseln und Entschlüsseln von `SecretString`. Der Verschlüsselungskontext umfasst den geheimen ARN, der die Verschlüsselung auf dieses spezifische Geheimnis beschränkt. Als zusätzliche Sicherheitsmaßnahme gewährt es MediaTailor in Ihrem Namen einen AWS KMS Zuschuss. MediaTailor wendet eine [GrantConstraints](#) Operation an, die es uns nur ermöglicht, den mit dem geheimen Schlüssel `SecretString` verknüpften ARN zu entschlüsseln, der im Secrets Manager-Verschlüsselungskontext enthalten ist.

Informationen darüber, wie Secrets Manager den Verschlüsselungskontext verwendet, finden Sie im Thema [Verschlüsselungskontext](#) im AWS Key Management Service Entwicklerhandbuch.

Festlegung der Schlüsselrichtlinie

Schlüsselrichtlinien steuern den Zugriff auf den vom Kunden verwalteten Schlüssel. Jeder vom Kunden verwaltete Schlüssel muss über genau eine Schlüsselrichtlinie verfügen, die aussagt, wer den Schlüssel wie verwenden kann. Wenn Sie Ihren vom Kunden verwalteten Schlüssel erstellen, können Sie die Standardschlüsselrichtlinie verwenden. Weitere Informationen finden Sie unter [Authentifizierung und Zugriffskontrolle für AWS KMS](#) im Entwicklerhandbuch zu AWS Key Management Service .

Um Ihren vom Kunden verwalteten Schlüssel mit Ihren MediaTailor Quellstandortressourcen zu verwenden, müssen Sie dem IAM-Prinzipal, der die folgenden API-Operationen aufruft [CreateSourceLocation](#) oder [UpdateSourceLocation](#) verwendet, die Erlaubnis erteilen:

- `kms:CreateGrant`— Fügt einem vom Kunden verwalteten Schlüssel einen Zuschuss hinzu. MediaTailor erstellt einen Zuschuss für Ihren vom Kunden verwalteten Schlüssel, sodass dieser den Schlüssel verwenden kann, um einen Quellspeicherort zu erstellen oder zu aktualisieren, der mit Zugriffstoken-Authentifizierung konfiguriert ist. Weitere Informationen zur Verwendung von [Grants in AWS KMS](#) finden Sie im AWS Key Management Service Developer Guide.

Auf diese Weise können MediaTailor Sie Folgendes tun:

- Rufen Sie an, `Decrypt` damit Ihr Secrets Manager Manager-Geheimnis beim Aufrufen erfolgreich abgerufen werden kann [GetSecretValue](#).

- Rufen Sie `anRetireGrant`, um den Zuschuss zurückzuziehen, wenn der Quellspeicherort gelöscht oder der Zugriff auf den geheimen Schlüssel gesperrt wurde.

Im Folgenden finden Sie ein Beispiel für eine Richtlinienerklärung, die Sie hinzufügen können: MediaTailor

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

Weitere Informationen zur Angabe von Berechtigungen in einer Richtlinie und zur Problembehandlung beim Schlüsselzugriff finden Sie unter [Grants AWS KMS im AWS Key Management Service](#) Developer Guide.

Schritt 2: Erstellen Sie ein AWS Secrets Manager Geheimnis

Verwenden Sie Secrets Manager, um Ihr Zugriffstoken in Form eines durch einen `SecretString` AWS KMS Kunden verwalteten Schlüssels zu speichern. MediaTailor verwendet den Schlüssel, um den zu entschlüsseln. `SecretString` Informationen zur Verwendung von Secrets Manager AWS KMS zum Schutz von Geheimnissen finden Sie im Thema [AWS Secrets Manager Anwendungsmöglichkeiten AWS KMS](#) im AWS Key Management Service Entwicklerhandbuch.

Wenn Sie Origin AWS Elemental MediaPackage als Quelladresse verwenden und die MediaTailor Secrets Manager Manager-Zugriffstoken-Authentifizierung verwenden möchten, gehen Sie wie folgt vor [the section called “Integration mit MediaPackage Endpunkten, die CDN-Autorisierung verwenden”](#).

Sie können ein Secrets Manager-Geheimnis mit dem AWS-Managementkonsole oder programmgesteuert mit dem Secrets Manager erstellen. APIs

So erstellen Sie ein Secret

Folgen Sie den Schritten unter [Secrets erstellen und verwalten mit AWS Secrets Manager](#) im AWS Secrets Manager Benutzerhandbuch.

Beachten Sie bei der Erstellung Ihres Geheimnisses die folgenden Überlegungen:

- Das `KmsKeyId` muss der [Schlüssel-ARN](#) des vom Kunden verwalteten Schlüssels sein, den Sie in Schritt 1 erstellt haben.
- Sie müssen eine angeben [SecretString](#). Das `SecretString` sollte ein gültiges JSON-Objekt sein, das einen Schlüssel und einen Wert enthält, die das Zugriffstoken enthalten. Zum Beispiel `{"MyAccessTokenIdentifier": "112233445566"}`. Der Wert muss zwischen 8 und 128 Zeichen lang sein.

Wenn Sie Ihren Quellstandort mit Zugriffstoken-Authentifizierung konfigurieren, geben Sie den `SecretString` Schlüssel an. MediaTailor verwendet den Schlüssel, um das in der gespeicherte Zugriffstoken zu suchen und abzurufen `SecretString`.

Notieren Sie sich den geheimen ARN und den `SecretString` Schlüssel. Sie werden sie verwenden, wenn Sie Ihren Quellstandort für die Verwendung der Zugriffstoken-Authentifizierung konfigurieren.

Anhängen einer ressourcenbasierten Geheimrichtlinie

Um MediaTailor auf den geheimen Wert zugreifen zu können, müssen Sie dem Geheimnis eine ressourcenbasierte Richtlinie hinzufügen. Weitere Informationen finden Sie unter [Anhängen einer Berechtigungsrichtlinie an ein Secrets Manager Secret](#) im AWS Secrets Manager Benutzerhandbuch.

Im Folgenden finden Sie ein Beispiel für eine Richtlinienerklärung, die Sie hinzufügen können MediaTailor:

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"
    }
  ]
}

```

Schritt 3: Konfigurieren Sie einen MediaTailor Quellstandort mit Zugriffstoken-Authentifizierung

Sie können die Secrets Manager Manager-Zugriffstoken-Authentifizierung mit dem AWS-Managementkonsole oder programmgesteuert mit dem konfigurieren. MediaTailor APIs

So konfigurieren Sie einen Quellstandort mit Secrets Manager Manager-Zugriffstoken-Authentifizierung

Folgen Sie den Anweisungen [Access configuration](#) im AWS Elemental MediaTailor Benutzerhandbuch.

Integration mit MediaPackage Endpunkten, die CDN-Autorisierung verwenden

Wenn Sie Origin AWS Elemental MediaPackage als Quellstandort verwenden, MediaTailor kann die Integration mit MediaPackage Endpunkten erfolgen, die CDN-Autorisierung verwenden.

Gehen Sie wie folgt vor, um die Integration mit einem MediaPackage Endpunkt durchzuführen, der die CDN-Autorisierung verwendet.

Zur Integration mit MediaPackage

1. Führen Sie die Schritte unter [CDN-Autorisierung einrichten](#) im AWS Elemental MediaPackage Benutzerhandbuch aus, falls Sie dies noch nicht getan haben.
2. Schließen Sie das Verfahren in a [the section called “Schritt 1: Erstellen Sie einen AWS KMS symmetrischen, vom Kunden verwalteten Schlüssel”](#).
3. Ändern Sie das Geheimnis, das Sie bei der Einrichtung der MediaPackage CDN-Autorisierung erstellt haben. Ändern Sie das Geheimnis mit den folgenden Werten:
 - Aktualisieren Sie den ARN KmsKeyId mit dem vom Kunden verwalteten Schlüssel, in dem Sie ihn erstellt haben [the section called “Schritt 1: Erstellen Sie einen AWS KMS symmetrischen, vom Kunden verwalteten Schlüssel”](#).

- (Optional) Für können Sie entweder die SecretString UUID auf einen neuen Wert rotieren oder Sie können das vorhandene verschlüsselte Geheimnis verwenden, sofern es sich um ein Schlüssel- und Wertepaar in einem Standard-JSON-Format handelt, z. B. `{"MediaPackageCDNIdentifier": "112233445566778899"}`
4. Führen Sie die Schritte unter [au the section called "Anhängen einer ressourcenbasierten Geheimrichtlinie"](#).
 5. Führen Sie die Schritte unter [au the section called "Schritt 3: Konfigurieren Sie einen MediaTailor Quellstandort mit Zugriffstoken-Authentifizierung"](#).

So funktioniert die MediaTailor Secrets Manager Manager-Zugriffstoken-Authentifizierung

Nachdem Sie einen Quellspeicherort für die Verwendung der Zugriffstoken-Authentifizierung erstellt oder aktualisiert haben, MediaTailor fügt er das Zugriffstoken in einen HTTP-Header ein, wenn Sie Quellinhaltsmanifeste von Ihrem Ursprung anfordern.

Im Folgenden finden Sie eine Übersicht darüber, wie die Secrets Manager Manager-Zugriffstoken-Authentifizierung für die Herkunftsauthentifizierung am Quellstandort MediaTailor verwendet wird:

1. Wenn Sie einen MediaTailor Quellspeicherort erstellen oder aktualisieren, der die Zugriffstoken-Authentifizierung verwendet, MediaTailor sendet eine [DescribeSecret](#)Anfrage an Secrets Manager, um den mit dem Secret verknüpften AWS KMS Schlüssel zu ermitteln. Sie nehmen den geheimen ARN in Ihre Konfiguration für den Zugriff auf den Quellstandort auf.
2. MediaTailor erstellt eine Genehmigung für den vom Kunden verwalteten Schlüssel, MediaTailor sodass dieser mithilfe des Schlüssels auf das in der gespeicherte [Zugriffstoken](#) zugreifen und es entschlüsseln kann. SecretString Der Grant-Name wird lauten. MediaTailor-SourceLocation-*your AWS-Konto ID-source location name*

Sie können den Zugriff auf den Zuschuss jederzeit widerrufen oder ihm den Zugriff auf den vom Kunden verwalteten Schlüssel entziehen MediaTailor. Weitere Informationen finden Sie unter [RevokeGrant](#) in der AWS Key Management Service -API-Referenz.

3. Wenn eine VOD-Quelle erstellt, aktualisiert oder in einem Programm verwendet wird, sendet sie HTTP-Anfragen an die Quellverzeichnisse, MediaTailor um die mit den VOD-Quellen verknüpften Quellinhaltsmanifeste am Quellspeicherort abzurufen. Wenn die VOD-Quelle einem Quellstandort zugeordnet ist, für den ein Zugriffstoken konfiguriert ist, enthalten die Anfragen das Zugriffstoken als HTTP-Header-Wert.

Mit VOD-Quellen arbeiten

Eine VOD-Quelle steht für einen einzelnen Inhalt, z. B. ein Video oder eine Episode eines Podcasts, den Sie zu Ihrem Quellstandort hinzufügen. Du fügst deinem Quellort eine oder mehrere VOD-Quellen hinzu und ordnest dann jede VOD-Quelle einem Programm zu, nachdem du deinen Kanal erstellt hast.

Jede VOD-Quelle muss mindestens eine Paketkonfiguration haben. Eine Paketkonfiguration gibt ein Paketformat, einen Manifestspeicherort und eine Quellgruppe für Ihre VOD-Quelle an. Wenn Sie Ihren Kanal erstellen, verwenden Sie die Quellgruppen der Paketkonfiguration, um die entsprechenden Ausgaben auf Ihrem Kanal zu erstellen. Wenn eine Quelle beispielsweise in zwei verschiedenen Formaten — HLS und DASH — verpackt ist, würdest du zwei Paketkonfigurationen erstellen, eine für DASH und eine für HLS. Dann würden Sie zwei Kanalausgänge erstellen, einen für jede Paketkonfiguration. Jeder Kanalausgang bietet einen Endpunkt, der für Wiedergabeanfragen verwendet wird. Im obigen Beispiel würde der Kanal also einen Endpunkt für HLS-Wiedergabeanfragen und einen Endpunkt für DASH-Wiedergabeanfragen bereitstellen.

Wenn Sie möchten, dass die Offsets der Werbemarkierungen in Ihrem Manifest automatisch erkannt werden, muss jede Werbemarkierung in allen Paketkonfigurationen mit demselben Offset erscheinen und eine Dauer von Null haben. MediaTailor wird bei HLS erkannt DATERANGE und EXT-X-CUE-OUT markiert. Bei DASH erkennt HLS das erste Event-Tag innerhalb jedes EventStream Tags.

Im folgenden Beispiel wird eine Werbeunterbrechungsmöglichkeit aufgrund des DATERANGE Tags mit einer Dauer von 0,0 bei einem Offset von 12000 ms erkannt. Das erste DATERANGE Tag mit einem Offset von 0 ms wird nicht erkannt, da es eine Dauer von 10,0 hat.

```
#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
```

Im folgenden Beispiel wird eine Werbeblock-Opportunity mit einem Offset von 0 ms erkannt, da das EXT-X-CUE-OUT Tag eine Dauer von 0 hat und unmittelbar darauf ein EXT-X-CUE-IN Tag folgt. Das zweite EXT-X-CUE-IN Paar EXT-X-CUE-OUT/wird nicht erkannt, da es eine Dauer von 10 hat.

```
#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
```

Im folgenden Beispiel wird eine Werbeunterbrechungsmöglichkeit mit einem Offset von 0 ms erkannt, da das erste Ereignis in dem Zeitraum EventStream eintritt, der bei PT0 0,000S beginnt. Das zweite Ereignis Event im EventStream wird nicht erkannt.

```
<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>
```

Hinzufügen von VOD-Quellen zu Ihrem Quellstandort

Im folgenden Verfahren wird erklärt, wie Sie VOD-Quellen zu Ihrem Quellspeicherort hinzufügen und Paketkonfigurationen mithilfe der MediaTailor Konsole einrichten. Informationen zum Hinzufügen von VOD-Quellen mithilfe der MediaTailor API finden Sie [CreateVodSource](#) in der AWS Elemental MediaTailor API-Referenz.

Important

Bevor Sie Ihre VOD-Quellen hinzufügen, stellen Sie sicher, dass sie die folgenden Anforderungen erfüllen:

- Die Quellvarianten müssen alle dieselbe Länge haben, wie im Quellmanifest festgelegt.
- Innerhalb einer Paketkonfiguration muss jede Quelle dieselbe Anzahl von untergeordneten Streams haben.

Aufgrund dieser Anforderungen unterstützen wir weder pro Titel noch automatisiertes ABR, da diese Kodierungsmethoden unterschiedliche Manifestlängen und untergeordnete Streams erzeugen können.

Wir empfehlen Ihnen, eine Kodierungsvorlage zu verwenden, die eine Mindestsegmentlänge beinhaltet, um sicherzustellen, dass Ihre codierten Quellen diese Anforderungen erfüllen.

Um VOD-Quellen zu Ihren Quellspeicherorten hinzuzufügen

1. Öffnen Sie die MediaTailor Konsole unter <https://console.aws.amazon.com/mediatailor/>
2. Wählen Sie im Navigationsbereich Channel-Assembly > Source Locations aus.
3. Wählen Sie im Bereich Quellverzeichnisse den Quellspeicherort aus, den Sie in dem [So erstellen Sie einen Quellspeicherort](#) Verfahren erstellt haben.
4. Wählen Sie VOD-Quelle hinzufügen aus.

5. Geben Sie unter VOD-Quelldetails einen Namen für Ihre VOD-Quelle ein:
 - Name: Eine Kennung für Ihre VOD-Quelle, z. B. my-example-video
6. **source-group-name** Geben Sie unter Paketkonfigurationen > Informationen zur Paketkonfiguration ein:

 Note

Die Paketkonfigurationen Ihrer Quelle müssen alle dieselbe Dauer haben, wie im Manifest der Quelle festgelegt. Und alle Quellen innerhalb einer Paketkonfiguration müssen dieselbe Anzahl von untergeordneten Streams haben. Um diese Anforderungen zu erfüllen, empfehlen wir Ihnen, eine Kodierungsvorlage für Ihre Assets zu verwenden. Wir empfehlen, eine Kodierungsvorlage mit einer Mindestsegmentlänge von einer Sekunde zu verwenden. MediaTailor unterstützt kein Streaming pro Titel oder automatisiertes Streaming mit adaptiver Bitrate (ABR), da diese Kodierungsmethoden gegen diese Anforderungen verstoßen.

- Quellgruppe: Geben Sie einen Quellgruppennamen ein, der diese Paketkonfiguration beschreibt, z. B. HLS-4k. Notieren Sie sich diesen Namen. Sie werden ihn angeben, wenn Sie die Ausgabe Ihres Kanals erstellen. Weitere Informationen finden Sie unter [Verwende Quellgruppen mit den Ausgaben deines Kanals](#).
- Typ: Wählen Sie das Paketformat für diese Konfiguration aus. MediaTailor unterstützt HLS und DASH.
- Relativer Pfad: Der relative Pfad von der Basis-HTTP-URL des Quellspeicherorts zum Manifest. Zum Beispiel /my/path/index.m3u8.

 Note

MediaTailor importiert automatisch alle Untertitel und untergeordneten Streams, die in einer übergeordneten multivarianten Playlist enthalten sind. Sie müssen nicht für jede Ihrer Quellversionen (DASH) oder Variantenstreams (HLS) separate Paketkonfigurationen erstellen.

Weitere Informationen zu Paketkonfigurationen finden Sie unter. [Verwenden von Paketkonfigurationen](#)

7. Wählen Sie VOD-Quelle hinzufügen aus.

Wenn Sie weitere VOD-Quellen hinzufügen möchten, wiederholen Sie die Schritte 4-7 des Verfahrens.

Arbeiten mit Live-Quellen

Eine Live-Quelle steht für einen einzelnen Live-Stream, z. B. ein Fußball-Live-Spiel oder eine Nachrichtenübertragung, den Sie zu Ihrem Quellstandort hinzufügen. Nachdem du deinen Kanal erstellt hast, fügst du deinem Quellstandort eine oder mehrere Live-Quellen hinzu und ordnest dann jede Live-Quelle einem Programm zu.

MediaTailor unterstützt die folgenden Arten der linearen Kanalanordnung:

- VOD-Quellen für einen Kanal, der Inhalte enthält VOD-to-live
- Live-Quellen für einen Kanal, der live-to-live Inhalte enthält, die mit Inhalten vermischt sind VOD-to-live

Ein Beispiel für VOD-to-live Inhalte ist ein Kanal, der eine Bibliothek mit VOD-Assets zu einem Live-Stream zusammenfügt. Ein Beispiel für live-to-live Inhalte, die mit VOD-to-live Inhalten gemischt werden, ist ein Kanal, auf dem hauptsächlich VOD-Inhalte gezeigt werden, mit Ausnahme einer nächtlichen Nachrichtenveranstaltung oder einer vorab geplanten Live-Sportveranstaltung. Ein anderes Beispiel für live-to-live Inhalte, die mit VOD-to-live Inhalten gemischt werden, ist ein reiner live-to-live Kanal, dessen Herkunft je nach Tageszeit variiert.

Sie können Live-Quellen verwenden, um einen regionalen Kanal einzurichten, der hauptsächlich nationale Programme zeigt, aber auch regionale Programmüberschreibungen beinhaltet und VOD-Inhalte enthält. Dazu lassen Sie ein encoder/packager Paar für die nationalen Inhalte laufen und lassen dann regionale Encoder laufen, wenn diese Regionen live sind. Anschließend erstellen Sie regionale Kanäle zur Kanalmontage, von denen jeder seinen eigenen Zeitplan hat. Auf diese Weise können die Zuschauer nach Bedarf hin und her wechseln. Dieses Setup hilft Ihnen, die encoding/packaging Kosten zu minimieren.

Jede Live-Quelle muss mindestens eine Paketkonfiguration haben. Eine Paketkonfiguration gibt ein Paketformat, einen Speicherort für das Manifest und eine Quellgruppe für Ihre Live-Quelle an. Wenn Sie Ihren Kanal erstellen, verwenden Sie die Quellgruppen der Paketkonfiguration, um die entsprechenden Ausgaben auf Ihrem Kanal zu erstellen. Wenn eine Quelle beispielsweise in zwei verschiedenen Formaten — HLS und DASH — verpackt ist, würdest du zwei Paketkonfigurationen

erstellen, eine für DASH und eine für HLS. Dann würden Sie zwei Kanalausgänge erstellen, einen für jede Paketkonfiguration. Jeder Kanalausgang bietet einen Endpunkt, der für Wiedergabeanfragen verwendet wird. In diesem Beispiel bietet der Kanal einen Endpunkt für HLS-Wiedergabeanfragen und einen Endpunkt für DASH-Wiedergabeanfragen.

Allgemeine Anforderungen für die Verwendung von Live-Quellen

Wenn Sie Live-Quellen verwenden, müssen Ihre Inhalte den folgenden allgemeinen Anforderungen entsprechen:

- **HLS-Live-Quellen** — Sie müssen `#EXT-X-PROGRAM-DATE-TIME` Tags für das erste Segment im Manifestfenster und für jede Unterbrechung angeben.
- **HLS** — Sie müssen Anzeigenmarkierungen als konfigurieren. `DATERANGE`
- **Fenster mit Quellmanifest** — Wir empfehlen, ein Manifestfenster mit einer Dauer zu verwenden, die mindestens so lang ist wie das Manifestfenster auf Ihrem MediaTailor Channel Assembly-Kanal. Es hat sich bewährt, ein Manifestfenster mit einer Dauer von 30 Sekunden oder länger als das Manifestfenster auf dem Channel Assembly-Kanal zu verwenden.
- **Passen Sie die Zieldauer an die Dauer der vorhandenen Quellen an.**
- **Passen Sie die Anzahl der untergeordneten Playlisten an die Anzahl der vorhandenen Quellen an.**

Konfigurationen

Wenn Sie andere Mediendienste von AWS Elemental als Teil Ihres Workflows für Live-Quellen verwenden, empfehlen wir, bei der Einrichtung Ihrer MediaPackage Konfiguration die bewährten Methoden zu befolgen. In der folgenden Tabelle wird beschrieben, wie Sie MediaPackage Einstellungen auf der Grundlage des von Ihnen verwendeten Streaming-Standards konfigurieren.

MediaPackage Einrichtung für Live-Quellen

Standard	Einstellung	Wert	Notwendigkeit	Hinweise
HLS	Endpunkttyp	Apple HLS	Erforderlich, es sei denn, Sie verwenden CMAF	Um HLS-Jobs zuzuordnen ts AWS Elemental MediaConvert
HLS	Endpunkttyp	CMAF	Erforderlich, sofern Sie	Um mp4 AWS Elemental

Standard	Einstellung	Wert	Notwendigkeit	Hinweise
			Apple HLS nicht verwenden	MediaConv ert HLS-Jobs abzugleichen
HLS	ProgramDateTimeIntervalSeconds	1	Erforderlich	Sie müssen für jedes Segment einen Wert angeben#EXT-X-PROGRAM-DATE-TIME , um Wiedergabeprobleme bei Unterbrechungen zu vermeiden.
HLS	PlaylistWindowSeconds	30 Sekunden länger als das Channel-Assembly-Manifestfenster	Erforderlich	
HLS	AdMarkers	DATERANGE	Erforderlich beim Durchgeben von Werbemarkierungen	
HLS	IncludeIframeOnlyStream	Disabled	Empfohlen	
DASH	ManifestLayout	FULL	Empfohlen	

Standard	Einstellung	Wert	Notwendigkeit	Hinweise
DASH	SegmentTemplateFormat	NUMBER_WITH_TIMELINE oder TIME_WITH_TIMELINE	Empfohlen	NUMBER_WITH_DURATION wird nicht unterstützt.
DASH	ManifestWindowSeconds	30 Sekunden länger als das Channel-Assembly-Manifestfenster	Erforderlich	
DASH	PeriodTriggers	ADS	Erforderlich beim Durchgeben von Werbemarkierungen	

Live-Quellen zu Ihrem Quellstandort hinzufügen

Im folgenden Verfahren wird erklärt, wie Sie mit der MediaTailor Konsole Live-Quellen zu Ihrem Quellstandort hinzufügen und Paketkonfigurationen einrichten. Informationen zum Hinzufügen von Live-Quellen mithilfe der MediaTailor API finden Sie [CreateLiveSource](#) in der AWS Elemental MediaTailor API-Referenz.

Important

Bevor Sie Ihre Live-Quellen hinzufügen, stellen Sie sicher, dass innerhalb einer Paketkonfiguration jede Quelle dieselbe Anzahl von untergeordneten Streams hat.

Um Live-Quellen zu Ihren Quellspeicherorten hinzuzufügen

1. Öffnen Sie die MediaTailor Konsole unter <https://console.aws.amazon.com/mediatailor/>.
2. Wählen Sie im Navigationsbereich Channel-Assembly > Source Locations aus.

3. Wählen Sie im Bereich Quellverzeichnisse den Quellspeicherort aus, den Sie in dem [So erstellen Sie einen Quellspeicherort](#) Verfahren erstellt haben.
4. Wählen Sie auf der Registerkarte Live-Quellen die Option Live-Quelle hinzufügen aus.
5. Geben Sie unter Live-Quelldetails einen Namen für Ihre Live-Quelle ein:
 - Name: Eine Kennung für Ihre Live-Quelle, z. my-example-videoB.
6. **source-group-name** Geben Sie unter Paketkonfigurationen > Informationen zur Paketkonfiguration ein:

 Note

Innerhalb einer Paketkonfiguration müssen alle VOD-Quellen und Live-Quellen dieselbe Anzahl an untergeordneten Streams haben. Wir empfehlen, dass Sie Ihre Quell-Streams auf die gleiche Weise konfigurieren.

- Quellgruppe: Geben Sie einen Quellgruppennamen ein, der diese Paketkonfiguration beschreibt, z. B. HLS-4k. Notieren Sie sich diesen Namen. Sie werden ihn angeben, wenn Sie die Ausgabe Ihres Kanals erstellen. Weitere Informationen finden Sie unter [Verwenden Quellgruppen mit den Ausgaben deines Kanals](#).
- Typ: Wählen Sie das Paketformat für diese Konfiguration aus. MediaTailor unterstützt HLS und DASH.
- Relativer Pfad: Der relative Pfad von der Basis-HTTP-URL des Quellspeicherorts zum Manifest. Zum Beispiel/my/path/index.m3u8.

 Note

MediaTailor importiert automatisch alle Untertitel und untergeordneten Streams, die in einer übergeordneten multivarianten Playlist enthalten sind. Sie müssen nicht für jede Ihrer Quellversionen (DASH) oder Variantenstreams (HLS) separate Paketkonfigurationen erstellen.

Weitere Informationen zu Paketkonfigurationen finden Sie unter. [Verwenden von Paketkonfigurationen](#)

7. Wählen Sie Live-Quelle hinzufügen aus.

Wenn Sie weitere Live-Quellen hinzufügen möchten, wiederholen Sie die Schritte 4-6 des Verfahrens.

Verwenden von Paketkonfigurationen

Eine Paketkonfiguration ist eine Darstellung der Quelle, die die verschiedenen Verpackungsmerkmale enthält, die für die Wiedergabe auf verschiedenen Geräten erforderlich sind. Angenommen, Sie haben eine Quelle mit drei Paketformaten: HLS mit DRM, DASH mit Segment-Timeline-Adressierung und HLS mit CMAF-Segmenten.

Bei der Channel-Assembly werden Ihre Quellen nicht neu verpackt. Wenn Sie mehrere Paketformate für eine bestimmte Quelle einbeziehen möchten, müssen Sie jedes Paketformat am Quellspeicherort verfügbar machen und den Pfad zu jedem Paketformat angeben.

Jedes Paketkonfigurationsobjekt muss Folgendes enthalten:

- Relativer Pfad — Der vollständige Pfad zum Paketformat der Quelle, relativ zum Quellspeicherort. Zum Beispiel `/my/path/index.m3u8`.
- Quellgruppe — Der Name der Quellgruppe, die verwendet wird, um Paketkonfigurationen mit der Ausgabe eines Kanals zu verknüpfen.
- Typ — Entweder HLS oder DASH.

Nachdem Sie einen Kanal erstellt haben, müssen Sie auch jede Quellgruppe deklarieren, die Sie für die Ausgabe des Kanals verwenden möchten.

Zwischenspeichern von Manifesten

MediaTailor speichert die Quell-Playlisten regelmäßig und bei Gelegenheit im Cache, um die Leistung und Zuverlässigkeit der Kanalmontage zu verbessern. Manchmal ist die zwischengespeicherte Version im Vergleich zur Originalversion an Ihrem Quellspeicherort veraltet. Rufen Sie auf MediaTailor, um die Aktualisierung der zwischengespeicherten Version der Quelle zu erzwingen. [UpdateVodSource](#) Verwenden Sie diesen Aufruf beispielsweise, wenn sich die eingebetteten Pfade in Ihrer Quelle ändern. Stellen Sie sicher, dass Sie immer eine up-to-date Version der Quelle an Ihrem Quellspeicherort verfügbar haben, auch wenn Sie nur wenige Anfragen von erhalten MediaTailor.

Mit Kanälen arbeiten

Ein Kanal fügt Ihre Quellmanifeste zu einem linearen Stream zusammen. Jeder Kanal enthält einen oder mehrere Ausgänge, die Ihren Paketkonfigurationen entsprechen.

Zuerst erstellen Sie einen Kanal und fügen dann Ihre VOD-Quellen und Live-Quellen zum Zeitplan des Kanals hinzu, indem Sie Programme erstellen. Jedes Programm ist mit einer VOD-Quelle oder einer Live-Quelle verknüpft.

Themen

- [Erstellen Sie mit der MediaTailor Konsole einen Kanal](#)
- [Verwende Quellgruppen mit den Ausgaben deines Kanals](#)
- [Löschen Sie einen Kanal mithilfe der Konsole MediaTailor](#)

Erstellen Sie mit der MediaTailor Konsole einen Kanal

Im folgenden Verfahren wird beschrieben, wie Sie mithilfe der MediaTailor Konsole einen Kanal erstellen.

So erstellen Sie einen Kanal

1. Öffnen Sie die MediaTailor Konsole unter <https://console.aws.amazon.com/mediatailor/>.
2. Wählen Sie im Navigationsbereich Channel-Assembly > Channels aus.
3. Wählen Sie in der Navigationsleiste die Option Kanal erstellen aus.
4. Geb unter Kanaldetails Details zu deinem Kanal ein:
 - Name: Gib einen Namen für deinen Kanal ein.
 - Stufe: Die Stufe bestimmt, welche Funktionen der Kanal unterstützt und wie viel der Betrieb des Kanals kostet. Weitere Informationen zur Preisgestaltung finden Sie auf der [Preisseite von Channel Assembly](#). MediaTailor unterstützt die folgenden Stufen:
 - Basic — Die Basic-Stufe unterstützt sowohl den linearen als auch den Loop-Wiedergabemodus und unterstützt keine Live-Quellen.
 - Standard — Die Standard-Stufe unterstützt Live-Quellen und erfordert den linearen Wiedergabemodus.

Wenn Sie in den Kanaldetails die Option Standard auswählen, können Sie die Zielgruppen unter Zielgruppendedetails definieren. Diese Zielgruppen werden für ProgramRules verwendet, wenn Sie AudienceMedia für Ihr Standardprogramm erstellen.

- Wählen Sie Hinzufügen aus.
 - Geben Sie den Namen der Zielgruppe in das Textfeld ein. Er muss zwischen 1 und 32 alphanumerische Zeichen lang sein.
 - Wählen Sie Bestätigen aus.
 - Wählen Sie Weiter aus.
- Wiedergabemodus: Der Wiedergabemodus legt das Wiedergabeverhalten des Kanals fest. MediaTailor unterstützt die folgenden Wiedergabemodi:
- Loop — Die Programme im Zeitplan werden back-to-back in einer Endlosschleife abgespielt. Nachdem das letzte Programm in einem Zeitplan abgespielt wurde, kehrt die Wiedergabe zum ersten Programm zurück. Die Wiedergabe wird in einer Endlosschleife fortgesetzt, bis Sie den Kanal beenden.
 - Linear — Jedes Programm im Zeitplan wird einmal abgespielt, back-to-back.

5.

Wählen Sie für Filler Slate den Quellspeicherort, der auf den Slate-Speicherort verweist, und den Namen der VOD-Quelle, die als Slate verwendet werden soll. MediaTailor verwendet das Slate, um Lücken zwischen Programmen im Zeitplan zu füllen. Wenn die Dauer der Slate kürzer als die Dauer der Lücke zwischen den Programmen ist, wird die MediaTailor Slate wiederholt. Sie müssen das Feld Filler Slate konfigurieren, wenn Ihr Kanal den linearen Wiedergabemodus verwendet. MediaTailor unterstützt Filler Slate für den Loop-Wiedergabemodus nicht.

6. Wählen Sie Weiter aus.

7. Geben Sie die Zielgruppendedetails in den Programmregeln an.

8. Wenn Sie in den Kanaldetails die Option Standard auswählen, können Sie die Zielgruppen unter Zielgruppendedetails definieren. Diese Zielgruppen werden für Programmregeln verwendet, wenn Sie AudienceMedia für Ihr Standardprogramm erstellen:

- Wählen Sie Hinzufügen aus, fügen Sie im Textfeld eine Zielgruppe hinzu und wählen Sie dann Bestätigen aus.

 Note

Geben Sie einen Namen ein, der nicht länger als _ 32 alphanumerische Zeichen ist.

- **Ausgabebetyp:** Wählen Sie das Streaming-Format für den Kanal aus. DASH und HLS werden unterstützt.
 - **Quellgruppe:** Geben Sie den Namen der Quellgruppe ein, die Sie in Ihrer Paketkonfiguration erstellt haben, wie unter beschrieben. [Hinzufügen von VOD-Quellen zu Ihrem Quellstandort](#)
9. Klicken Sie auf Weiter.
 10. Geben Sie unter Manifesteinstellungen zusätzliche Informationen zu Ihren Manifesteinstellungen ein:
 - **Manifestfenster (Sekunden):** Das Zeitfenster in Sekunden, das in jedem Manifest enthalten ist. Der Mindestwert ist 30 Sekunden und der Höchstwert ist 3600 Sekunden.
 - **Art der Anzeigenmarkierung (nur HLS-Ausgaben):** Die Art der Anzeigen-Tags, die in Werbeunterbrechungen von VOD-Programmen erscheinen. Wählen Sie `Daterange` diese Option aus, um Werbeunterbrechungen mit Tags in VOD-Programme MediaTailor einzufügen. `EXT-X-DATERANGE` Wählen Sie `Scte35 Enhanced` diese Option aus, um Werbeunterbrechungen mithilfe `EXT-X-CUE-OUT` von `UND`-Tags in VOD-Programme MediaTailor einzufügen. `EXT-X-CUE-IN` Weitere Informationen zu diesen Tag-Typen finden Sie unter [SCTE-35-Nachrichten für Werbeunterbrechungen](#). Leitet bei Live-Workflows MediaTailor immer `DATERANGE` Tags und keine erweiterten `Scte35`-Tags weiter, unabhängig vom ausgewählten Anzeigen-Markup-Typ.
 11. Wenn Sie mehrere Kanalausgänge konfigurieren möchten, wählen Sie unter Ausgänge die Option Hinzufügen. Konfigurieren Sie dann die Details für Ihre Ausgabe, indem Sie die Schritte 6 und 7 in diesem Verfahren ausführen.
 12. Wählen Sie Weiter aus.
 13. Wählen Sie unter Kanalrichtlinie die IAM-Richtlinieneinstellungen Ihres Kanals aus:
 - **Keine Kanalrichtlinie anhängen:** Beschränken Sie die Wiedergabe nur auf Personen, die Zugriff auf die Anmeldeinformationen dieses Kontos haben.
 - **Benutzerdefinierte Richtlinie anhängen:** Definieren Sie Ihre eigene Richtlinie und beschränken Sie den Zugriff auf so wenige oder so viele, wie Sie möchten.
 - **Öffentliche Richtlinie anhängen:** Akzeptiere alle eingehenden Kundenanfragen an die Ausgabe eines Kanals. Sie müssen diese Option verwenden, wenn Sie die MediaTailor Anzeigeneinfügung verwenden möchten.
 14. Wählen Sie Weiter aus.
 15. Überprüfen Sie Ihre Einstellungen im Bereich Überprüfen und erstellen.

16. Wählen Sie Create channel (Channel erstellen).

Note

Kanäle werden im gestoppten Zustand erstellt. Dein Kanal ist erst aktiv, wenn du ihn mit der MediaTailor Konsole oder der MediaTailor StartChannel API startest.

Verwende Quellgruppen mit den Ausgaben deines Kanals

Eine Quellgruppe verknüpft eine Paketkonfiguration mit einem Ausgang auf einem Kanal. Wenn Sie die Paketkonfiguration auf der Quelle erstellen, identifizieren Sie den Namen der Quellgruppe. Wenn Sie dann die Ausgabe auf dem Kanal erstellen, geben Sie denselben Namen ein, um die Ausgabe mit der Paketkonfiguration zu verknüpfen. VOD-Quellen und Live-Quellen, die einem Programm auf einem Kanal hinzugefügt werden, müssen zu der Quellgruppe gehören, die in der Ausgabe angegeben ist.

Beispiel:

- Die VOD-Quellen 1 und 2 haben beide drei Paketkonfigurationen mit den Quellgruppen HLS, DASH und HLS-4k.
- VOD-Quelle 3 hat zwei Paketkonfigurationen mit den Quellgruppen HLS und DASH.

Wenn Kanal A zwei Ausgänge mit den Quellgruppen HLS und DASH hat, kann der Kanalausgang alle drei VOD-Quellen verwenden. Das liegt daran, dass die VOD-Quellen 1, 2 und 3 alle Paketkonfigurationen mit den Quellgruppenbezeichnungen HLS und DASH haben.

Wenn Kanal B zwei Ausgänge mit den Quellgruppen HLS und HLS-4k hat, kann er die VOD-Quelle 1 und 2 verwenden, aber nicht 3. Das liegt daran, dass die VOD-Quellen 1 und 2 beide Paketkonfigurationen mit den Quellgruppenbezeichnungen HLS und HLS-4k haben.

Wenn Kanal C einen einzigen Ausgang mit der Quellgruppe DASH hat, kann er alle drei VOD-Quellen verwenden. Alle drei VOD-Quellen haben Paketkonfigurationen mit der DASH-Quellgruppe.

Löschen Sie einen Kanal mithilfe der Konsole MediaTailor

Gehen Sie wie folgt vor, um Ihren Kanal zu löschen.

Um deinen Kanal zu löschen

1. Öffne die MediaTailor Konsole unter <https://console.aws.amazon.com/mediatailor/>.
2. Wählen Sie im Navigationsbereich Channel-Assembly > Channels aus.
3. Wählen Sie den Kanal aus, den Sie löschen möchten.
4. Wenn dein Kanal läuft, wähle im Drop-down-Menü Aktionen die Option Stopp aus. Du musst deinen Kanal beenden, bevor du ihn löschen kannst.
5. Wenn dein Kanal gestoppt ist, wähle im Drop-down-Menü Aktionen die Option Löschen aus.

Ein Programm zum Zeitplan eines Kanals hinzufügen

Jedes Programm enthält eine VOD-Quelle oder eine Live-Quelle, die Teil eines Quellenverzeichnisses in Ihrem Konto ist. Du fügst deine Programme zum Zeitplan deines Kanals hinzu, um die Reihenfolge zu kontrollieren, in der sie im Stream deines Kanals abgespielt werden.

Ein Programm, das eine VOD-Quelle enthält, kann mit einer oder mehreren Werbeunterbrechungen konfiguriert werden. Jede Werbeunterbrechung enthält einen Slate, bei dem es sich um eine VOD-Quelle von einem Quellort handelt. Um den Werbeblock zu erstellen, fügen Sie den Slate mit einem Abstand von Millisekunden in das Programm ein.

Themen

- [Ein Programm innerhalb eines Kanalzeitplans mithilfe der Konsole erstellen MediaTailor](#)
- [Definition von Zielgruppenkohorten und alternativen Inhalten mithilfe der Programmregeln](#)
- [Generierung zielgruppenspezifischer Manifeste](#)

Ein Programm innerhalb eines Kanalzeitplans mithilfe der Konsole erstellen MediaTailor

Im folgenden Verfahren wird beschrieben, wie Sie mithilfe der MediaTailor Konsole ein Programm innerhalb des Zeitplans Ihres Kanals erstellen. Außerdem wird beschrieben, wie Werbeunterbrechungen konfiguriert werden, die optional sind. Informationen zum Erstellen von Programmen mithilfe der MediaTailor API finden Sie [CreateProgram](#) in der AWS Elemental MediaTailor API-Referenz.

Um ein Programm hinzuzufügen

1. Öffnen Sie die MediaTailor Konsole unter <https://console.aws.amazon.com/mediatailor/>.
2. Wählen Sie im Navigationsbereich Kanalmontage > Kanäle aus.
3. Wählen Sie im Bereich Kanäle den Kanal aus, den Sie in dem [So erstellen Sie einen Kanal](#) Verfahren erstellt haben.
4. Geben Sie in den Programmdetails Details zu Ihrem Programm ein:
 - Name: Dies ist der Name des Programms, das du deinem Kanal hinzufügst.
 - Quelltyp: Legt fest, welche Art von Quellvideo das Programm abspielt. Diese Option ist nur für Standardkanäle verfügbar.
 - VOD — Das Programm spielt eine video-on-demand Quelle ab, z. B. eine zuvor aufgezeichnete TV-Episode.
 - Live — Das Programm spielt eine Live-Quelle ab, z. B. eine Live-Nachrichtenübertragung.
 - Name des Quellverzeichnisses: Der Quellspeicherort, MediaTailor der dem Programm zugeordnet ist.
 - Wenn Sie „Bestehenden Quellspeicherort auswählen“ wählen, wählen Sie im Menü „Quellspeicherort auswählen“ einen Quellspeicherort aus. Sie können auch anhand des Namens nach Ihrem Quellstandort suchen. Dies ist hilfreich, wenn Sie über eine große Anzahl von Quellstandorten verfügen.
 - Wenn Sie Geben Sie den Quellspeicherort ein wählen, suchen Sie anhand des Namens nach Ihrem Quellspeicherort.
 - Name der VOD-Quelle: Der Name der VOD-Quelle, die dem Programm MediaTailor zugeordnet ist:
 - Wenn Sie Select a existing VOD-Quelle wählen, wählen Sie einen VOD-Quellennamen aus der Liste der VOD-Quellen, die Ihrem Konto zugeordnet sind. Sie können auch anhand des Namens nach Ihrer VOD-Quelle suchen. Dies ist hilfreich, wenn Sie über eine große Anzahl von VOD-Quellen verfügen.
 - Wenn Sie „Nach Namen suchen“ wählen, suchen Sie anhand des Namens nach Ihrer Live-Quelle.
 - Name der Live-Quelle: Der Name der Live-Quelle, die dem Programm zugeordnet werden soll. Diese Option ist nur verfügbar, wenn Sie Live als Quelltyp ausgewählt haben.
 - Wenn Sie „Bestehenden Quellspeicherort auswählen“ wählen, wählen Sie im Menü „Quellspeicherort auswählen“ einen Quellspeicherort aus. Sie können auch anhand des

Namens nach Ihrem Quellstandort suchen. Dies ist hilfreich, wenn Sie über eine große Anzahl von Quellstandorten verfügen.

- Wenn Sie Geben Sie den Quellspeicherort ein wählen, suchen Sie anhand des Namens nach Ihrem Quellspeicherort.
- Name der VOD-Quelle: Der Name der VOD-Quelle, die dem Programm MediaTailor zugeordnet ist:
 - Wenn Sie „Vorhandene Live-Quelle auswählen“ wählen, wählen Sie einen Namen für die Live-Quelle aus der Liste der Live-Quellen aus, die mit Ihrem Konto verknüpft sind. Sie können Ihre Live-Quelle auch anhand des Namens suchen. Dies ist hilfreich, wenn Sie über eine große Anzahl von Live-Quellen verfügen.
 - Wenn Sie „Nach Namen suchen“ wählen, suchen Sie anhand des Namens nach Ihrer Live-Quelle.

5. Wählen Sie Weiter aus, um zur Registerkarte „Konfiguration planen“ zu wechseln.

6. Definieren Sie unter Wiedergabekonfiguration, wann ein Programm im Zeitplan Ihres Kanals abgespielt wird:

- Dauer in Millisekunden: Definiert die Dauer des Programms in Millisekunden. Diese Option ist nur für Programme verfügbar, die Live-Quellen verwenden.
- Übergangstyp: Definiert die Übergänge von Programm zu Programm im Zeitplan:
 - Relativ: Das Programm wird entweder vor oder nach einem anderen Programm im Zeitplan abgespielt. Diese Option ist nur für Programme verfügbar, die VOD-Quellen verwenden.
 - Absolut: Das Programm wird zu einer bestimmten Uhrzeit abgespielt. MediaTailor bemüht sich nach besten Kräften, das Programm zu der von Ihnen angegebenen Uhrzeit abzuspielen. MediaTailor startet die Wiedergabe des Programms an einer gemeinsamen Segmentgrenze zwischen dem vorherigen Programm oder Slate. Diese Option ist nur für Kanäle verfügbar, die für die Verwendung des linearen Modus konfiguriert sind [Wiedergabemodus: Der Wiedergabemodus legt das Wiedergabeverhalten des Kanals fest. MediaTailor unterstützt die folgenden Wiedergabemodi:](#).
- Startzeit des Programms: Bei absoluten Übergangstypen ist dies die Uhrzeit, zu der das Programm während der Uhrzeit, zu der die Wiedergabe des Programms geplant ist. Wenn Sie dieses Programm zu einem laufenden linearen Kanal hinzufügen, müssen Sie eine Startzeit eingeben, die 15 Minuten oder später als die aktuelle Uhrzeit liegt.
- Relative Position: Wählen Sie aus, wo das Programm relativ zu einem anderen Programm in den Zeitplan eingefügt werden soll. Sie können „Vor dem Programm“ oder „Nach

dem Programm“ wählen. Diese Einstellung gilt nicht, wenn dies das erste Programm im Programm deines Kanals ist.

- Wenn Sie „Bestehendes Programm auswählen“ wählen, wählen Sie im Menü „Bestehendes Programm verwenden“ den Programmnamen aus einer vordefinierten Liste der nächsten 100 Programme aus, die vom Kanal wiedergegeben werden.
- Wenn du „Nach einem Programm anhand des Namens suchen“ auswählst, gib den Namen eines bestehenden Programms in deinem Kanal ein.

Wenn du Werbeunterbrechungen zu deinem Programm hinzufügen möchtest, fahre mit dem nächsten Schritt fort. Werbeunterbrechungen können nur für Programme konfiguriert werden, die VOD-Quellen verwenden. Bei Live-Quellen werden Werbeunterbrechungen in DASH-Manifesten und Werbeunterbrechungen in HLS-Manifesten, die das EXT-X-DATERANGE Tag verwenden, automatisch weitergegeben.

7. Wählen Sie Weiter aus, um zu Werbeunterbrechungen hinzuzufügen zu gelangen.
8. Wähle Werbeunterbrechung hinzufügen aus. Konfigurieren Sie unter Werbeunterbrechungen die Einstellungen für die Werbeunterbrechung:
 - Name des Speicherorts der Slate-Quelle: Wählen Sie einen vorhandenen Quellspeicherort auswählen und wählen Sie den Quellspeicherort aus, an dem Ihr Slate gespeichert ist, den Sie zuvor in dieser Aufgabe erstellt haben.
 - Name der VOD-Quelle: Wählen Sie „Eine vorhandene VOD-Quelle auswählen“ und wählen Sie die VOD-Quelle aus, die Sie für Slate verwenden und die Sie zuvor in dieser Aufgabe hinzugefügt haben. Die Dauer des Slate bestimmt die Dauer der Werbepause.
 - Offset in Millisekunden: Dieser Wert bestimmt die Startzeit der Werbepause in Millisekunden als Offset im Verhältnis zum Beginn des Programms. Geben Sie einen Wert ein, der kürzer als die Dauer der VOD-Quelle ist und der an einer Segmentgrenze für alle Titel innerhalb der VOD-Quelle des Programms (alle Audio-, Video- und Untertitelspuren) ausgerichtet ist. Andernfalls wird die Werbeunterbrechung übersprungen. Wenn Sie beispielsweise 0 eingeben, entsteht eine Pre-Roll-Werbeunterbrechung, die vor Programmbeginn abgespielt wird.
 - Verfügbare Zahl: MediaTailor [schreibt, in welche dieser Wert geschrieben wird](#) `splice_insert.avail_num`, wie in Abschnitt 9.7.3.1 der SCTE-35-Spezifikation, [Digital Program Insertion Cueing Message](#), definiert. Der Standardwert lautet 0. Die Werte müssen zwischen 0 und 256 (einschließlich) liegen.
 - Erwartete Verfügbarkeit: MediaTailor schreibt diesen Wert `splice_insert.avails_expected`, wie in Abschnitt 9.7.3.1 der SCTE-35-Spezifikation

definiert. Der Standardwert lautet 0. Die Werte müssen zwischen 0 und 256 (einschließlich) liegen.

- ID des Splice-Ereignisses: MediaTailor schreibt diesen Wert `insplice_insert.splice_event_id`, wie in Abschnitt 9.7.3.1 der SCTE-35-Spezifikation definiert. Der Standardwert ist 1.
- Eindeutige Programm-ID: MediaTailor schreibt diesen Wert `insplice_insert.unique_program_id`, wie in Abschnitt 9.7.3.1 der SCTE-35-Spezifikation definiert. Der Standardwert lautet 0. Die Werte müssen zwischen 0 und 256 (einschließlich) liegen.

9. Wählen Sie für einen linearen Standardkanal die Option Weiter aus, um zu Alternative Medien einstellen zu wechseln.

Weitere Informationen zur Verwendung MediaTailor zum Erstellen alternativer Medien finden Sie unter [Alternative Medien erstellen](#).

Weiterführende Informationen MediaTailor zur Personalisierung Ihrer Werbeunterbrechungen finden Sie unter [Fügen Sie personalisierte Anzeigen und Werbeunterbrechungen in einen Channel-Stream ein](#).

10. Wählen Sie Weiter aus, um zu Überprüfen und erstellen zu wechseln.
11. Wählen Sie Programm hinzufügen aus.

Weiterführende Informationen MediaTailor zur Personalisierung Ihrer Werbeunterbrechungen finden Sie unter [Fügen Sie personalisierte Anzeigen und Werbeunterbrechungen in einen Channel-Stream ein](#).

12.

 **Important**

Wenn Sie bei Kanälen, die sich in einer Endlosschleife befinden, die Programmliste für eine Sendung ändern, die innerhalb der nächsten 10 Minuten geplant ist, wird die Änderung erst in der nächsten Endlosschleife sichtbar.

Geben Sie unter Programmdetails Details zu Ihrem Programm ein:

- Name: Dies ist der Name des Programms, das du deinem Kanal hinzufügst.
- Quelltyp: Bestimmt, welche Art von Quelle das Programm wiedergibt. Diese Option ist nur für Standardkanäle verfügbar.

- VOD — Das Programm spielt eine VOD-Quelle ab, z. B. eine zuvor aufgezeichnete TV-Episode.
- Live — Das Programm spielt eine Live-Quelle ab, z. B. eine Live-Nachrichtenübertragung.
- Name des Quellverzeichnisses: Der Quellspeicherort, der dem Programm zugeordnet werden soll.

Wenn Sie „Vorhandenen Quellspeicherort auswählen“ wählen, wählen Sie im Dropdownmenü „Quellspeicherort auswählen“ einen Quellspeicherort aus. Sie können Ihren Quellstandort auch anhand des Namens suchen. Dies ist hilfreich, wenn Sie über eine große Anzahl von Quellstandorten verfügen.

Wenn Sie Geben Sie den Quellspeicherort ein wählen, suchen Sie anhand des Namens nach Ihrem Quellspeicherort.

- Name der VOD-Quelle: Der Name der VOD-Quelle, die dem Programm zugeordnet werden soll.

Wenn Sie Select a existing VOD-Quelle wählen, wählen Sie einen VOD-Quellennamen aus der Liste der VOD-Quellen, die Ihrem Konto zugeordnet sind. Sie können Ihre VOD-Quelle auch anhand des Namens suchen. Dies ist hilfreich, wenn Sie über eine große Anzahl von VOD-Quellen verfügen.

Wenn Sie „Nach Namen suchen“ wählen, suchen Sie anhand des Namens nach Ihrer VOD-Quelle.

- Name der Live-Quelle: Der Name der Live-Quelle, die dem Programm zugeordnet werden soll. Diese Option ist nur verfügbar, wenn Sie Live als Quelltyp ausgewählt haben.

Wenn Sie „Vorhandene Live-Quelle auswählen“ wählen, wählen Sie einen Namen für die Live-Quelle aus der Liste der Live-Quellen aus, die mit Ihrem Konto verknüpft sind. Sie können Ihre Live-Quelle auch anhand des Namens suchen. Dies ist hilfreich, wenn Sie über eine große Anzahl von Live-Quellen verfügen.

Wenn Sie „Nach Namen suchen“ wählen, suchen Sie anhand des Namens nach Ihrer Live-Quelle.

13. Lege unter Wiedergabekonfiguration fest, wann ein Programm im Zeitplan deines Kanals abgespielt wird:

- Dauer in Millisekunden: Definiert die Dauer des Programms in Millisekunden. Diese Option ist nur für Programme verfügbar, die Live-Quellen verwenden.

- **Übergangstyp:** Definiert die Übergänge von Programm zu Programm im Zeitplan.
 - **Relativ** — Das Programm wird entweder vor oder nach einem anderen Programm im Zeitplan abgespielt. Diese Option ist nur für Programme verfügbar, die VOD-Quellen verwenden.
 - **Absolut** — Das Programm wird zu einer bestimmten Uhrzeit abgespielt. MediaTailor bemüht sich nach besten Kräften, das Programm zu der von Ihnen angegebenen Uhrzeit abzuspielen. Wir starten die Wiedergabe des Programms an einer gemeinsamen Segmentgrenze zwischen dem vorherigen Programm oder Slate. Diese Option ist nur für Kanäle verfügbar, die für die Verwendung von konfiguriert sind [linear playback mode](#).

 Note

Beachten Sie das folgende Verhalten bei absoluten Übergangstypen:

- Wenn das vorherige Programm im Zeitplan eine Dauer hat, die über die Wanduhrzeit hinausgeht, wird das vorherige MediaTailor Programm an der gemeinsamen Segmentgrenze gekürzt, die der Wanduhr am nächsten ist.
 - Wenn zwischen den Programmen im Zeitplan Lücken bestehen, MediaTailor wird abgespielt. [filler slate](#) Wenn die Dauer der Slate kürzer als die Dauer der Lücke ist, wird die Slate in einer MediaTailor Schleife wiederholt.
- **Startzeit des Programms** — Bei absoluten Übergangstypen die Uhrzeit, zu der das Programm auf der Wanduhr abgespielt werden soll. Wenn Sie dieses Programm zu einem laufenden linearen Kanal hinzufügen, müssen Sie eine Startzeit eingeben, die 15 Minuten oder später als die aktuelle Uhrzeit liegt.
 - **Relative Position:** Wählen Sie aus, wo das Programm relativ zu einem anderen Programm in den Zeitplan eingefügt werden soll. Sie können „Vor dem Programm“ oder „Nach dem Programm“ wählen. Diese Einstellung gilt nicht, wenn dies das erste Programm im Programm Ihres Kanals ist.
 - **Relatives Programm:** Der Name des Programms, das verwendet werden soll, um das neue Programm davor oder danach einzufügen. Diese Einstellung gilt nicht, wenn dies das erste Programm im Zeitplan Ihres Kanals ist.

Wenn Sie „Bestehendes Programm auswählen“ wählen, wählen Sie im Dropdownmenü „Bestehendes Programm verwenden“ den Namen des Programms aus einer vordefinierten Liste der nächsten 100 Programme aus, die vom Kanal wiedergegeben werden.

Wenn du „Nach einem Programm anhand des Namens suchen“ auswählst, gib den Namen eines bestehenden Programms in deinem Kanal ein.

Wenn du Werbeunterbrechungen zu deinem Programm hinzufügen möchtest, fahre mit dem nächsten Schritt fort. Werbeunterbrechungen können nur für Programme konfiguriert werden, die VOD-Quellen verwenden. Bei Live-Quellen werden Werbeunterbrechungen in DASH-Manifesten und Werbeunterbrechungen in HLS-Manifesten, die das EXT-X-DATERANGE Tag verwenden, automatisch weitergegeben.

14. Wählen Sie Werbeunterbrechung hinzufügen aus. Konfigurieren Sie unter Werbeunterbrechungen die Einstellungen für die Werbeunterbrechung:

- Name des Speicherorts der Slate-Quelle: Wählen Sie einen vorhandenen Quellspeicherort auswählen und wählen Sie den Quellspeicherort aus, an dem Ihr Slate gespeichert ist, den Sie zuvor in diesem Tutorial erstellt haben.
- Name der VOD-Quelle: Wählen Sie „Eine vorhandene VOD-Quelle auswählen“ und wählen Sie die VOD-Quelle aus, die Sie für Slate verwenden und die Sie zuvor in diesem Tutorial hinzugefügt haben. Die Dauer des Slate bestimmt die Dauer der Werbepause.
- Für Offset in Millisekunden: Dieser Wert bestimmt die Startzeit der Werbeunterbrechung in Millisekunden als Offset im Verhältnis zum Beginn des Programms. Geben Sie einen Wert ein, der kürzer als die Dauer der VOD-Quelle ist und der an einer Segmentgrenze für alle Titel innerhalb der VOD-Quelle des Programms (alle Audio-, Video- und Untertitelspuren) ausgerichtet ist. Andernfalls wird die Werbeunterbrechung übersprungen. Wenn Sie beispielsweise 0 eingeben, entsteht eine Pre-Roll-Werbeunterbrechung, die vor Programmbeginn abgespielt wird.

 Note

Wenn in Ihrer VOD-Quelle Werbemarken wie DATERANGE oder EXT-X-CUE-OUT für HLS und EventStream für DASH mit einer Dauer von Null MediaTailor erkannt werden, können Sie den Offset dieser Werbemarken aus dem Drop-down-Menü auswählen, der als Offset für die Werbeunterbrechung verwendet werden soll. Damit eine Werbemöglichkeit erkannt wird, muss sie in allen Paketkonfigurationen innerhalb einer VOD-Quelle mit demselben Offset vorhanden sein und ihre Dauer muss Null sein.

- Nachrichtentyp: Der SCTE-35-Anzeigentyp. Wählen Sie entweder `SPLICE_INSERT` oder `TIME_SIGNAL`:
 - `SPLICE_INSERT`: Stellt mithilfe von Splice-Insert-Parametern grundlegende Metadaten zum Werbeblock bereit.
 - `TIME_SIGNAL`: Stellt erweiterte Metadaten mithilfe von Segmentierungsdeskriptoren bereit. Weitere Hinweise zu den Unterschieden zwischen Nachrichtentypen finden Sie unter [the section called “SCTE-35-Nachrichten für Werbeunterbrechungen”](#)

Für den Nachrichtentyp `SPLICE_INSERT`:

- Für die Avail-Nummer wird in diese geschriebensplice_insert.avail_num, wie in Abschnitt 9.7.3.1 der SCTE-35-Spezifikation definiert. Der Standardwert ist 0. Die Werte müssen zwischen und (einschließlich) liegen. 0 256
- Für Avail Expected ist dies wie in Abschnitt 9.7.3.1 der SCTE-35-Spezifikation definietsplice_insert.avails_expected, geschrieben. Der Standardwert ist 0. Die Werte müssen zwischen und (einschließlich) liegen. 0 256
- Für die Splice-Event-ID wird dies geschriebensplice_insert.splice_event_id, wie in Abschnitt 9.7.3.1 der SCTE-35-Spezifikation definiert. Der Standardwert ist 1.
- Bei der eindeutigen Programm-ID wird in diese geschriebensplice_insert.unique_program_id, wie in Abschnitt 9.7.3.1 der SCTE-35-Spezifikation definiert. Der Standardwert ist 0. Die Werte müssen zwischen und (einschließlich) liegen. 0 256

Für den Nachrichtentyp `TIME_SIGNAL`:

- Für die Segmentierungsereignis-ID wird in diese geschriebensegmentation_descriptor.segmentation_event_id, wie in Abschnitt 10.3.3.1 der SCTE-35-Spezifikation definiert. Der Standardwert ist 1.
- Bei der Segmentierungstyp-ID wird in diese geschriebensegmentation_descriptor.segmentation_type_id, wie in Abschnitt 10.3.3.1 der SCTE-35-Spezifikation definiert. Der Standardwert ist (0x30). 48 Die Werte müssen zwischen 0 und 256 (einschließlich) liegen.
- Für Segmentierung UPID wird dies geschriebensegmentation_descriptor.segmentation_upid, wie in Abschnitt 10.3.3.1 der SCTE-35-Spezifikation definiert. Der Wert muss eine hexadezimale Zeichenfolge sein, die die Zeichen und enthält. 0-9 A-F Der Standardwert ist eine leere Zeichenfolge.

- Für den Typ Segmentation UPID wird dies `segmentation_descriptor.segmentation_upid_type`, wie in Abschnitt 10.3.3.1 der SCTE-35-Spezifikation definiert. Der Standardwert ist (0x0E). Die Werte müssen zwischen 0 und 256 (einschließlich) liegen.
- Für die Segmentnummer wird dies `segmentation_descriptor.segment_num`, wie in Abschnitt 10.3.3.1 der SCTE-35-Spezifikation definiert. Der Standardwert ist 0. Die Werte müssen zwischen 0 und 256 (einschließlich) liegen.
- Bei den erwarteten Segmenten wird entsprechend der Definition in Abschnitt 10.3.3.1 der SCTE-35-Spezifikation darauf geschrieben. `segmentation_descriptor.segments_expected` Der Standardwert ist 0. Die Werte müssen zwischen 0 und 256 (einschließlich) liegen.

15. Wählen Sie Programm hinzufügen.

Weiterführende Informationen MediaTailor zur Personalisierung Ihrer Werbeunterbrechungen finden Sie unter [Fügen Sie personalisierte Anzeigen und Werbeunterbrechungen in einen Channel-Stream ein](#).

Note

Wenn dein Kanal mindestens eine Ausgabe mit einem Enhanced SCTE35 Werbemarkup-Typ hat, kannst du Metadaten für Werbeunterbrechungen einreichen. MediaTailor schreibt die eingereichten Schlüssel-Wert-Paare in das EXT-X-ASSET Tag für deine Werbepause.

Definition von Zielgruppenkohorten und alternativen Inhalten mithilfe der Programmregeln

Mit den Programmregeln kannst du Zielgruppenkohorten für einen Kanal definieren und alternative Medien angeben, die für diese Zielgruppen abgespielt werden sollen. Sie können einer Zielgruppe für ein Programm eine oder mehrere alternative Inhaltsquellen zuordnen. Nach dem Ende des Programms werden die standardmäßigen Zielgruppeninhalte wiedergegeben, sofern Sie keine weiteren alternativen Medien angeben.

Die Programmregeln sind auf Kanälen der Stufe STANDARD mit dem Wiedergabemodus LINEAR verfügbar. MediaTailor-Kanäle unterstützen alternative Medien für alle VOD-Quellen und Live-Quellen.

Ein Anwendungsbeispiel finden Sie unter [Programmregeln verwenden mit AWS MediaTailor](#).

Zielgruppen definieren

Definieren Sie Zielgruppen auf einem Kanal, indem Sie bei der Konfiguration eines MediaTailor Kanals die Zielgruppe nacheinander eingeben. Sie können dies entweder über die MediaTailor Konsole oder die MediaTailor CreateChannel API tun. Jede Zielgruppe muss zwischen 1 und 32 alphanumerische Zeichen lang sein. Wenn die für die Zielgruppen angegebenen Werte ungültig sind, schlägt die Anfrage fehl.

Sie können Zielgruppen nur auf Kanälen der Stufe STANDARD im Wiedergabemodus LINEAR definieren.

Wenn Sie die Zielgruppen aktualisieren müssen, können Sie dies entweder über die MediaTailor Konsole oder die MediaTailor UpdateChannel API tun.

Wenn Sie die ProgramRules Funktion verwenden, stellen Sie sicher, dass die in CreateProgram oder die UpdateProgram Anfrage AudienceMedia definierte Zielgruppe die im Kanal definierte vorhandene Zielgruppe enthält.

Alternative Medien erstellen

In der folgenden Aufgabe wird erklärt, wie alternative Medien mithilfe der MediaTailor Konsole definiert werden. Informationen zur Definition alternativer Medien mithilfe der MediaTailor API finden Sie [CreateProgram](#) in der AWS Elemental MediaTailor API-Referenz.

So definieren Sie alternative Medien in einem neuen Programm:

1. Öffnen Sie die MediaTailor Konsole unter <https://console.aws.amazon.com/mediatailor/>.
2. Wählen Sie im Navigationsbereich Kanalmontage > Kanäle aus.
3. Wählen Sie den Kanalnamen aus, dem Sie alternative Medien hinzufügen möchten.
4. Erstellen Sie ein Programm. Weitere Informationen finden Sie unter [Ein Programm innerhalb eines Kanalzeitplans mithilfe der Konsole erstellen MediaTailor](#).
5. Alternative Medien konfigurieren:
 - Wählen Sie im Feld Zielgruppen die Option Hinzufügen aus, um die Zielgruppe auszuwählen, für die Sie alternative Medien definieren.
 - Wählen Sie im Menü „Zielgruppe“ eine für den Kanal definierte Zielgruppe aus.

- Wähle Alternative Medien hinzufügen aus, um mit der Definition alternativer Medien für das Programm zu beginnen.
- MediaTailor erstellt ein Feld für alternative Medien 1. Dies ist der erste Inhalt, der im Programm als alternative Medien MediaTailor wiedergegeben wird.
- Im Feld Alternative Medien 1:
 - Wählen Sie einen Quellspeicherort aus.
 - Wählen Sie entweder einen VOD - oder einen Live-Quellentyp aus:

Für VOD

- Wählen Sie VOD als Quelltyp aus.
- (Optional) Geben Sie einen Clip-Bereich an. Bei VOD-Quellen, einschließlich VOD-Quellen für alternative Medien, können Sie festlegen, dass ein Teil einer VOD-Quelle abgespielt werden soll, wobei der Ausschnitt vom Anfang and/or bis zum Ende der Quelle ausgeschnitten wird. Geben Sie die Anfangs- und Endversätze in Millisekunden an.
- (Optional) a\ Werbeunterbrechungen hinzufügen. Dies erfolgt auf die gleiche Weise wie beim Erstellen von Programmen. Weitere Informationen finden Sie unter [Ein Programm innerhalb eines Kanalzeitplans mithilfe der Konsole erstellen MediaTailor](#).

Für Live

- Wählen Sie Live als Quelltyp aus.
- Wählen Sie eine Live-Quelle aus.
- Geben Sie eine Startzeit in Millisekunden der Uhrzeit ein, zu der diese Live-Quelle starten soll. Die Live-Quelle wird nur innerhalb des Zeitrahmens des Standardprogramms abgespielt, für das sie definiert ist. Wenn die Startzeit vor dem Start des Standardprogramms liegt, beginnt sie erst, wenn das Standardprogramm dies tut. Wenn die Startzeit nach dem Ende des Standardprogramms liegt, MediaTailor wird die Live-Quelle nicht abgespielt.
- Geben Sie eine Dauer in Millisekunden ein. Die Dauer muss mindestens 10 Minuten lang sein.
- Zusätzliche alternative Medien können diesem Programm für das Publikum hinzugefügt werden, indem Sie erneut Alternative Medien hinzufügen auswählen. Dadurch wird ein weiteres Feld mit der Bezeichnung Alternative Medien 2 erstellt. Sie können bis zu 5 alternative Medienquellen pro Programm und Zielgruppe angeben.

- Wenn Sie mit der Definition alternativer Medien für alle gewünschten Zielgruppen fertig sind, wählen Sie Weiter und fahren Sie mit der Erstellung des Programms fort.

Weitere Informationen finden Sie unter [Ein Programm innerhalb eines Kanalzeitplans mithilfe der Konsole erstellen MediaTailor](#).

 Note

Alternative Medien werden nur in dem Zeitraum abgespielt, für den sie definiert sind. Wenn der gesamte alternative Inhalt den Standardinhalt überschreitet, MediaTailor wird er gekürzt. MediaTailor spielt alternative Medien in der Reihenfolge ab, in der sie definiert sind. Die Startzeiten für Live-Alternativmedien haben immer Vorrang und zuvor geplante VOD-Quellen oder Live-Quellen werden gekürzt. Jede Zeit, die nicht mit alternativen Medien für ein Publikum gefüllt ist, wird mit der vom Kanal definierten Füllliste gefüllt

- Um Zielgruppenmedien für andere Zielgruppen zu definieren, wählen Sie neben Zielgruppen erneut Hinzufügen aus. Wählen Sie die neu erstellte Zielgruppe aus, legen Sie die Zielgruppen-ID fest und fügen Sie alternative Medien hinzu, wie oben beschrieben. Bis zu 5 Zielgruppen können alternative Medien in einem Programm verwenden.

Generierung zielgruppenspezifischer Manifeste

Verwenden Sie den `aws.mediatailor.channel.audienceId` Abfrageparameter, um ein Manifest für eine bestimmte Zielgruppe abzurufen. Dieser Abfrageparameter kann dynamisch von Ihrem CDN angehängt oder durch einen Aufruf an Ihr Content- oder Kundenverwaltungssystem hinzugefügt werden. Sie müssen die Zuordnung einer bestimmten Playback-Sitzung zu einer externen Sitzung beibehalten. `audienceId` MediaTailor Dadurch wird anstelle des Standardinhalts ein zielgruppenspezifisches Manifest abgerufen, in dem alle alternativen Medien für diese Zielgruppe definiert sind. Es ist wichtig, dass, sobald ein Manifest für eine bestimmte Zielgruppe angefordert wird, der Player das Manifest immer mit derselben Zielgruppen-ID anfordert, da es sonst zu Wiedergabefehlern kommen kann.

Wenn eine Anfrage für eine Zielgruppe gestellt wird, die auf dem Kanal nicht existiert, wird ein 404-Fehler MediaTailor zurückgegeben.

Example Ein Manifest für eine Zielgruppe abrufen

```
https:// prefix>.channel-assembly.mediatailor.us-  
west-2.amazonaws.com/v1/channel/ExampleChannel/index_dash.mpd?  
aws.mediatailor.channel.audienceId=Seattle
```

Fügen Sie personalisierte Anzeigen und Werbeunterbrechungen in einen Channel-Stream ein

Mit kannst du lineare Streams zur Kanalmontage monetarisieren MediaTailor, indem du Werbeunterbrechungen in deine Programme einfügst, ohne den Inhalt mit SCTE-35-Markern zu konditionieren. Sie können Channel Assembly mit dem MediaTailor Ad Insertion Service oder mit jeder beliebigen serverseitigen Anzeigeneinfügung (SSAI) verwenden.

In den folgenden Themen wird gezeigt, wie Sie personalisierte Anzeigen und Werbeunterbrechungen in den linearen Stream Ihres Kanals einfügen.

Themen

- [Die Anzeigeneinfügung einrichten mit MediaTailor](#)
- [SCTE-35-Nachrichten für Werbeunterbrechungen](#)

Die Anzeigeneinfügung einrichten mit MediaTailor

Um personalisierte Werbung in den Stream deines Kanals einzufügen, ist die Endpunkt-URL deines Kanals die Inhaltsquelle dafür AWS Elemental MediaTailor. In dieser Anleitung erfahren Sie, wie Sie das Einfügen von Anzeigen einrichten MediaTailor .

Voraussetzungen

Überprüfen Sie zu Beginn, ob die folgenden Anforderungen erfüllt sind:

- Bereiten Sie Ihre HLS- und DASH-Streams für die MediaTailor Anzeigenschaltung vor.
 - Falls Sie noch keine Inhaltsstreams vorbereitet haben, finden Sie weitere Informationen [Schritt 2: Bereiten Sie einen Stream vor](#) im Thema Erste Schritte mit der MediaTailor Anzeigeneinfügung.
- Haben Sie einen Ad Decision Server (ADS).
- Konfigurieren Sie die Einstellungen für Werbeunterbrechungen im Programm. Weitere Informationen finden Sie im [Configuring ad breaks for your program](#) Verfahren.

Als bewährte Methode sollten Sie die Verwendung eines Content Delivery Network (CDN) zwischen der Kanalzusammenstellung und der MediaTailor Anzeigeneinblendung in Betracht ziehen. Der Dienst zum Einfügen von MediaTailor Anzeigen kann zusätzliche Anfragen an die Herkunft generieren. Daher empfiehlt es sich, Ihr CDN so zu konfigurieren, dass es die Manifeste aus der Channel-Assembly als Proxy weiterleitet, und dann das CDN mit dem Präfix der URLs Inhaltsquellen-URL verwenden.

Für das Einfügen von Anzeigen konfigurieren MediaTailor

Im Folgenden wird gezeigt, wie Sie die MediaTailor Konsoleneinstellungen so konfigurieren, dass Sie personalisierte Werbung in den Stream Ihres Kanals einfügen können.

Um das Einfügen MediaTailor von Anzeigen zu konfigurieren

1. Öffnen Sie die MediaTailor Konsole unter <https://console.aws.amazon.com/mediatailor/>.
2. Wählen Sie im Navigationsbereich Konfigurationen aus.
3. Geben Sie unter Erforderliche Einstellungen die grundlegenden erforderlichen Informationen zu Ihrer Konfiguration ein:
 - Name: Der Name Ihrer Konfiguration.
 - Inhaltsquelle: Gib die Wiedergabe-URL aus der Ausgabe deines Kanals ein, abzüglich des Dateinamens und der Erweiterung. Weiterführende Informationen zur MediaTailor Konfiguration finden Sie unter [Erforderliche Einstellungen](#).
 - Ad Decision Server: Geben Sie die URL für Ihr ADS ein.
4. Sie können optional die Konfigurationsalias, die Personalisierungsdetails und die erweiterten Einstellungen konfigurieren. Informationen zu diesen Einstellungen finden Sie unter [Optionale Konfigurationseinstellungen](#)
5. Wählen Sie in der Navigationsleiste Konfiguration erstellen aus.

Nachdem Sie die MediaTailor Anzeigeneinfügung eingerichtet haben, können Sie auch Werbeunterbrechungen einrichten. Detaillierte Anweisungen finden Sie unter [Erste Schritte mit der MediaTailor Anzeigenschaltung](#).

SCTE-35-Nachrichten für Werbeunterbrechungen

Mit können Sie einen Inhaltskanal erstellen MediaTailor, der auf dem Quellstandort und den VOD-Quellressourcen basiert. Anschließend können Sie für jedes Programm im Zeitplan eines Kanals eine

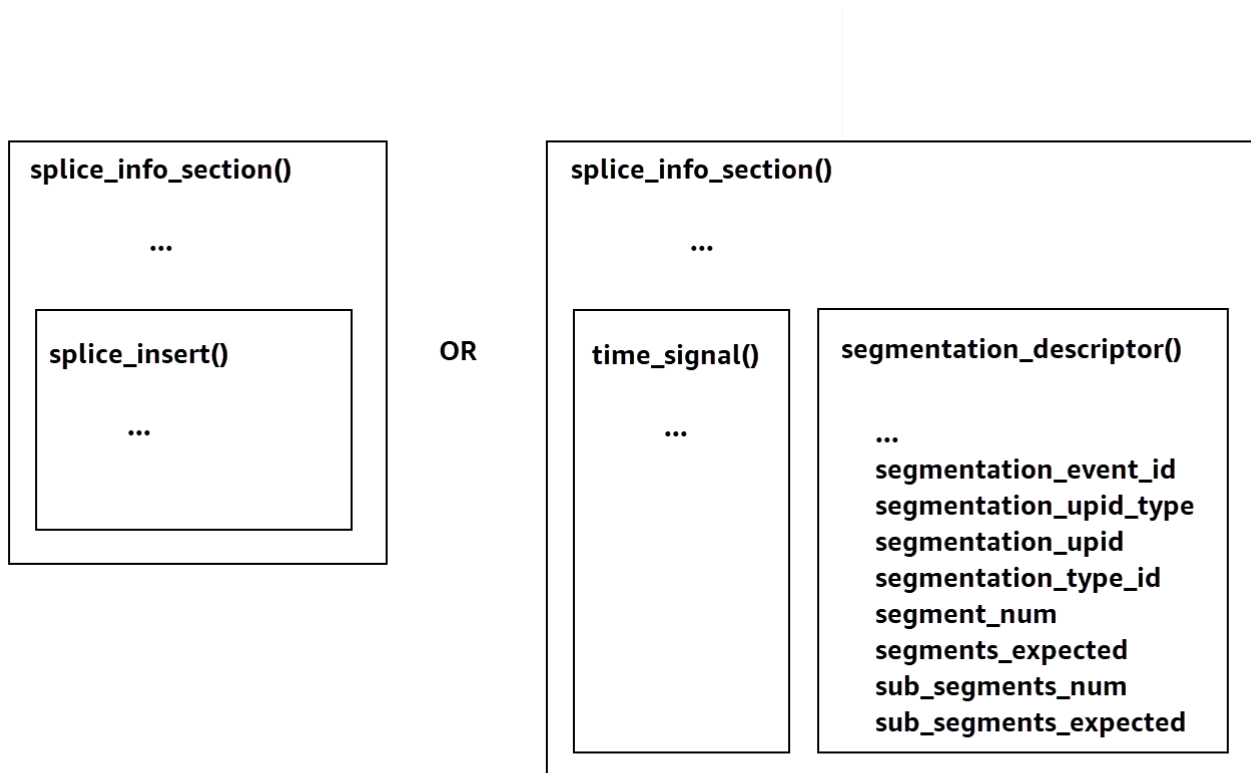
oder mehrere Werbeunterbrechungen einrichten. Sie verwenden Nachrichten, die auf der SCTE-35-Spezifikation basieren, um den Inhalt auf Werbeunterbrechungen vorzubereiten. Sie können beispielsweise SCTE-35-Nachrichten verwenden, um Metadaten zu den Werbeunterbrechungen bereitzustellen. Weitere Informationen zur SCTE-35-Spezifikation finden Sie unter [Digital Program Insertion Cueing Message](#).

Sie können die Werbeunterbrechungen auf eine von zwei Arten einrichten:

- Eine `time_signal` SCTE-35-Nachricht mit einer Nachricht anhängen.
`segmentation_descriptor` Diese `segmentation_descriptor` Nachricht enthält erweiterte Metadatenfelder wie Inhaltskennungen, die mehr Informationen über die Werbeunterbrechung enthalten. MediaTailor schreibt die Anzeigenmetadaten als Teil der SCTE-35-Daten der EXT-X-DATERANGE (HLS) oder EventStream (DASH) -Werbemarkierung in das Ausgabemanifest.
- Anhängen einer `splice_insert` SCTE-35-Nachricht, die grundlegende Metadaten zum Werbeblock enthält.
- HLS:
 - Wenn der Markup-Typ „Werbung“ lautet `Daterange`, werden Werbeunterbrechungen als EXT-X-DATERANGE Tags im MediaTailor Manifest angegeben.
 - Wenn der Markup-Typ „Werbung“ lautet `Scte35 Enhanced`, werden Werbeunterbrechungen mithilfe der folgenden Tags MediaTailor angegeben:
 - MediaTailor platziert ein Zeichen EXT-X-CUE-OUT auf dem ersten Segment des Anzeigenfensters, was auf einen Schnitt vom Inhalt bis zur Werbeunterbrechung hinweist. Es enthält die erwartete Dauer der Werbeunterbrechung, z. `EXT-X-CUE-OUT:Duration=30` B.
 - `>EXT-X-ASSET`: Dieses Tag erscheint im selben Segment wie die Metadaten der Werbeunterbrechung, die AdBreak bei der Erstellung oder Aktualisierung des Programms angegeben wurden, EXT-X-CUE-OUT und enthält diese. Es enthält CAID immer.
 - `EXT-0ATCLS-SCTE35`: Dieses Tag erscheint im selben Segment wie die EXT-X-CUE-OUT SCTE-35-Nachricht und enthält Base64-kodierte Byte.
 - `EXT-X-CUE-OUT-CONT`: Dieses Tag erscheint in jedem nachfolgenden Segment innerhalb der Werbeanzeige und enthält Informationen zur Dauer und zur verstrichenen Zeit. Es enthält auch die Base64-kodierte SCTE-35-Nachricht und die CAID
 - `EXT-X-CUE-IN`: Dieses Tag erscheint im ersten Inhaltssegment, nachdem die Werbepause vorbei ist, und weist auf einen Schnitt von einer Werbepause zurück zum Inhalt hin.

Die folgende Abbildung zeigt die zwei Möglichkeiten, Werbeunterbrechungen in einem Kanal mithilfe von SCTE-35-Nachrichten einzurichten:

- Verwenden Sie eine `splice_insert()` Nachricht, um Werbeunterbrechungen mit grundlegenden Metadaten einzurichten.
- Verwenden Sie eine `time_signal()` Nachricht zusammen mit einer `segmentation_descriptor()` Nachricht, um Werbeunterbrechungen mit detaillierteren Metadaten einzurichten.



Informationen zur Verwendung `time_signal` finden Sie in Abschnitt 9.7.4 der SCTE-35-Spezifikation 2022, [Digital](#) Program Insertion Cueing Message.

Die Informationen zum Werbeunterbruch werden in den SCTE-35-Ausgabedaten angezeigt. `splice_info_section` Mit MediaTailor können Sie eine einzelne Nachricht mit einer einzelnen `segmentation_descriptor` `time_signal` Nachricht koppeln.

Note

Wenn Sie eine `segmentation_descriptor` Nachricht senden, müssen Sie sie als Teil des `time_signal` Nachrichtentyps senden. Die `time_signal` Nachricht enthält nur das `splice_time` Feld, das anhand eines bestimmten MediaTailor Zeitstempels erstellt wird.

In der folgenden Tabelle werden die Felder beschrieben, die für jede MediaTailor `segmentation_descriptor` Nachricht erforderlich sind. [Weitere Informationen finden Sie in Abschnitt 10.3.3.1 der SCTE-35-Spezifikation 2022, die Sie auf der ANSI Webstore-Website erwerben können.](#)

segmentation_descriptor Erforderliche Felder für eine Nachricht

Feld	Typ	Standardwert	Description
<code>segmentation_event_id</code>	Ganzzahl	1	Das ist geschrieben an <code>segmentation_descriptor.segmentation_event_id</code> .
<code>segmentation_upid_type</code>	Ganzzahl	14 (0x0E)	Das ist geschrieben. <code>segmentation_descriptor.segmentation_upid_type</code> Der Wert muss zwischen 0 und 256 (einschließlich) liegen.
<code>segmentation_upid</code>	Zeichenfolge	"" (leere Zeichenfolge)	Das ist geschrieben an <code>segmentation_descriptor.segmentation_upid</code> .

Feld	Typ	Standardwert	Description
			Der Wert muss eine hexadezimale Zeichenfolge sein, die die Zeichen 0-9 und enthält. A-F
segmentation_type_id	Ganzzahl	48 (0 x 30)	Das ist geschrieben. segmentation_descriptor.segmentation_type_id Der Wert muss zwischen 0 und 256 (einschließlich) liegen.
segment_num	Ganzzahl	0	Das ist geschrieben. segmentation_descriptor.segment_num . Der Wert muss zwischen 0 und 256 (einschließlich) liegen.
segments_expected	Ganzzahl	0	Das ist geschrieben. segmentation_descriptor.segments_expected . Der Wert muss zwischen 0 und 256 (einschließlich) liegen.

Feld	Typ	Standardwert	Description
sub_segment_num	Ganzzahl	null	Das ist geschriebensegmentation_descriptor.sub_segment_num. Der Wert muss zwischen 0 und 256 (einschließlich) liegen.
sub_segments_expected	Ganzzahl	null	Das ist geschriebensegmentation_descriptor.sub_segments_expected. Der Wert muss zwischen 0 und 256 (einschließlich) liegen.

Die folgende Tabelle zeigt die Werte, die für einige segmentation_descriptor Nachrichtfelder MediaTailor automatisch festgelegt werden.

Werte, die von MediaTailor für die Felder einer **segmentation_descriptor** Nachricht festgelegt wurden

Feld	Typ	Wert
segmentation_event_cancel_indicator	Boolesch	Wahr
program_segmentation_flag	Boolesch	Wahr
delivery_not_restricted_flag	Boolesch	Wahr

MediaTailor setzt immer den Wert `segmentation_duration_flag` auf `True`. MediaTailor füllt das `segmentation_duration` Feld mit der Dauer des Statusinhalts in Ticks aus.

 Note

Beim MediaTailor Senden der `time_signal` Nachrichten wird das `splice_command_type` Feld in der `splice_info_section` Nachricht auf 6 (0x06) gesetzt.

In der HLS-Ausgabe enthält das EXT-X-DATERANGE Ausgabe-Tag für eine AdBreak `time_signal` Nachricht ein SCTE-35 Feld, das auf die serialisierte Version der Nachricht festgelegt ist.

`splice_info_section` Das folgende EXT-X-DATERANGE Tag zeigt beispielsweise die serialisierte Version der Nachricht: `splice_info_section`

```
#EXT-X-DATERANGE:ID="\1",START-DATE="\2020-09-25T02:13:20Z",DURATION=300.0,SCTE35-  
OUT=0xFC002C0000000000000000FF00506800000000001602144355454900000000100E000019BFCC00E00300000000
```

In der DASH-Ausgabe enthält das EventStream Ausgabeelement für eine AdBreak `time_signal` Nachricht ein `scte35:SpliceInfoSection` Element mit `scte35:TimeSignal` und `scte35:SegmentationDescriptor` Elementen als untergeordneten Elementen. Das `scte35:TimeSignal` Element hat ein `scte35:SpliceTime` untergeordnetes Element, und das `scte35:SegmentationDescriptor` Element hat ein untergeordnetes `scte35:SegmentationUpid` Element. Die folgende DASH-Ausgabe zeigt beispielsweise die Struktur des untergeordneten EventStream Elements des Elements:

```
<EventStream schemeIdUri="urn:scte:scte35:2013:xml" timescale="900000">
  <Event duration="270000000">
    <scte35:SpliceInfoSection protocolVersion="0" ptsAdjustment="0" tier="4095">
      <scte35:TimeSignal>
        <scte35:SpliceTime ptsTime="0" />
      </scte35:TimeSignal>
      <scte35:SegmentationDescriptor segmentNum="0"
segmentationDuration="270000000" segmentationEventCancelIndicator="false"
segmentationEventId="1" segmentationTypeId="48" segmentsExpected="0">
        <scte35:SegmentationUpid segmentationUpidFormat="hexBinary"
segmentationUpidType="14">012345</scte35:SegmentationUpid>
      </scte35:SegmentationDescriptor>
    </scte35:SpliceInfoSection>
  </Event>
</EventStream>
```

Sie haben mehr über die Verwendung von SCTE-35-Nachrichten zur Einrichtung von Werbeunterbrechungen bei der Kanalzusammenstellung gelernt, die Struktur und die erforderlichen Felder für diese Nachrichten sowie Beispiele für HLS- und DASH-Ausgaben, die die SCTE-35-Nachrichten enthalten, gelernt.

Zeitversetzung bei der Wiedergabe eines Kanals

Mit MediaTailor Channel Assembly können Sie die Wiedergabe eines Kanals zeitlich verschieben. Time-Shifting ermöglicht es Zuschauern, die Inhalte eines Kanals zu einem anderen Zeitpunkt als der ursprünglichen Sendezeit anzusehen. Ein Zuschauer kann beispielsweise von Anfang an damit beginnen, sich eine Sendung anzusehen, auch wenn sie der Sendung erst beitreten, nachdem sie gestartet wurde. Oder ein Zuschauer kann eine Sendung pausieren und sie später wieder ansehen.

Um die Wiedergabe eines Kanals zeitlich zu verschieben, fügen Sie der Wiedergabe-URL des Kanals einen `start` Parameter hinzu. Der `start` Parameter gibt die Zeit an, zu der die Wiedergabe beginnen soll, relativ zur aktuellen Zeit. Wenn die aktuelle Uhrzeit beispielsweise 14:00 Uhr ist und Sie die Wiedergabe ab 13:00 Uhr starten möchten, würden Sie den `start` Parameter auf einstellen `-3600`, was -1 Stunde in Sekunden entspricht.

Der `start` Parameter akzeptiert sowohl positive als auch negative Werte in Sekunden:

- Negative Werte geben eine Zeit in der Vergangenheit an, relativ zur aktuellen Zeit. `start=-3600` bedeutet zum Beispiel „Wiedergabe von vor 1 Stunde starten“.
- Positive Werte geben eine Zeit in der future an, relativ zur aktuellen Zeit. `start=3600` bedeutet zum Beispiel „Wiedergabe ab 1 Stunde in der future starten“.

Das folgende Beispiel zeigt, wie der `start` Parameter zur Wiedergabe-URL eines Kanals hinzugefügt wird:

```
https://a1b2c3d4e5f6.mediapackage.us-west-2.amazonaws.com/out/v1/examplechannel/index.m3u8?start=-3600
```

In diesem Beispiel beginnt die Wiedergabe vor 1 Stunde.

Der `start` Parameter unterliegt den folgenden Einschränkungen:

- Das maximale Zeitfenster liegt 24 Stunden in der Vergangenheit oder future.
- Time-Shifting ist nur für Kanäle verfügbar, die den linearen Wiedergabemodus verwenden.

- Time-Shifting ist für Kanäle, die den Loop-Wiedergabemodus verwenden, nicht verfügbar.

Wenn Sie einen `start` Parameter angeben, der außerhalb des verfügbaren Time-Shift-Fensters liegt, wird ein MediaTailor Fehler zurückgegeben.

Verwendung eines CDN zur Optimierung der MediaTailor Anzeigenpersonalisierung und Inhaltsbereitstellung

AWS Elemental MediaTailor funktioniert effektiv als eigenständiger Dienst, aber die Integration in ein Content Delivery Network (CDN) wie Amazon CloudFront oder andere Drittanbieter CDNs kann Ihre Streaming-Workflows erheblich verbessern. Eine CDN-Integration ist besonders nützlich, wenn Sie Inhalte für ein großes, geografisch verteiltes Publikum bereitstellen müssen oder wenn Sie eine konsistente Anzeigenbereitstellung in verschiedenen Regionen sicherstellen möchten. AWS

Ohne ein CDN stellen Ihre Zuschauer eine direkte Verbindung her, um MediaTailor personalisierte Manifeste und Anzeigensegmente zu erhalten. Dies kann zu einer erhöhten Latenz führen, insbesondere bei Zuschauern, die weit von der AWS Region entfernt sind, in der Ihre MediaTailor Konfiguration bereitgestellt wird. Darüber hinaus kann es bei Direktverbindungen zu MediaTailor einer erhöhten Auslastung kommen, was sich negativ auf die Leistung auswirken kann.

Weitere Informationen zu MediaTailor Konzepten und Workflows finden Sie unter [Was ist AWS Elemental MediaTailor?](#).

Bei der Integration eines CDN mit ist es wichtig MediaTailor, die korrekte Handhabung von CORS (Cross-Origin Resource Sharing) zu konfigurieren, um Probleme zu vermeiden, die zu Wiedergabefehlern in webbasierten Playern führen können. Die richtige CORS-Konfiguration ist sowohl für Anzeigensegmente als auch für Inhaltssegmente unerlässlich. Anzeigensegmente sind zwar anfälliger für CORS-Probleme, aber die Anwendung einer konsistenten CORS-Behandlung für alle Segmenttypen gewährleistet ein zuverlässiges Wiedergabeerlebnis. Eine ausführliche Anleitung zur Konfiguration des CDN-Routing-Verhaltens mit der richtigen CORS-Handhabung finden Sie unter [Serienreife Konfiguration für CloudFront MediaTailor](#).

Die CDN-Integration ermöglicht auch erweiterte Funktionen für die Parameterübergabe und dynamisches Routing. Hinweise zur Weitergabe von Abfrageparametern CDNs für Autorisierung und Routing finden Sie unter [MediaTailor Manifest-Abfrageparameter](#). Informationen zum dynamischen Ad-Server- und Origin-Routing mithilfe von Konfigurationsaliasen finden Sie unter [MediaTailor dynamische Anzeigenvariablen für ADS-Anfragen](#).

Wenn Sie ein CDN zwischen Ihren Zuschauern platzieren, MediaTailor bietet dies die folgenden Vorteile:

- Reduzieren Sie die Latenz, indem Sie Inhalte von Randstandorten aus der Nähe der Zuschauer bereitstellen

- Verbessern Sie die Skalierbarkeit, indem Sie die Last auf die globale Infrastruktur des CDN verteilen
- Verbessern Sie die Zuverlässigkeit durch redundante Bereitstellungspfade
- Optimieren Sie die Kosten, indem Sie den ursprünglichen Verkehr reduzieren
- Implementieren Sie erweiterte Funktionen wie Media Quality-Aware Routing (MQAR) für eine verbesserte Streaming-Qualität

Themen

- [Wählen Sie das richtige CDN für Ihre Bedürfnisse](#)
- [Planen Sie Ihre CDN-Integration für AWS Elemental MediaTailor](#)
- [Richten Sie die CDN-Integration ein mit MediaTailor](#)
- [Richten Sie SSAI mit einem CDN für personalisierte Videowerbung ein](#)
- [Erstellen Sie MediaTailor lineare Kanäle mit Kanalmontage und CDN](#)
- [Integrieren MediaTailor mit MediaPackage und CDN](#)
- [Integration AWS Elemental MediaTailor mit Amazon CloudFront](#)
- [Drittanbieter CDNs für die MediaTailor Anzeigenbereitstellung einrichten](#)
- [Leitfaden zur Leistungsoptimierung für CDN und Integrationen MediaTailor](#)
- [Überwachen Sie den Betrieb und die Leistung von MediaTailor CDN](#)
- [Testen und Validieren für CDN und Integrationen MediaTailor](#)
- [Probleme mit MediaTailor der CDN-Integration beheben](#)
- [Analyse des CDN-Integrationsprotokolls und Fehlercode-Referenz für MediaTailor](#)
- [Automatisieren MediaTailor und CDN mit CloudFormation](#)
- [Serienreife Konfiguration für CloudFront MediaTailor](#)
- [Holen Sie sich Support und Hilfe zur Fehlerbehebung für CDN und Integrationen MediaTailor](#)

Wählen Sie das richtige CDN für Ihre Bedürfnisse

Die Wahl des richtigen Content Delivery Network (CDN) -Anbieters ist eine wichtige Entscheidung, die sich auf die Leistung der Inhaltsbereitstellung, die Kosten und das Zuschauererlebnis auswirken kann. AWS Elemental MediaTailor Berücksichtigen Sie bei der Auswahl eines CDN für Ihre MediaTailor Implementierung die folgenden Faktoren:

Geografische Abdeckung

Wählen Sie ein CDN mit starker Präsenz in Regionen, in denen sich Ihre Zielgruppe befindet. Verschiedene CDN-Anbieter haben in verschiedenen geografischen Regionen unterschiedliche Stärken.

Integration mit Diensten AWS

Amazon CloudFront bietet die engste Integration mit MediaTailor und anderen AWS Services, was die Einrichtung und Verwaltung vereinfachen kann. Drittanbieter bieten CDNs möglicherweise weitere Vorteile wie spezielle Funktionen zur Videobereitstellung oder eine stärkere Präsenz in bestimmten Regionen.

Videospezifische Funktionen

Halten Sie Ausschau nach Funktionen CDNs, die speziell für die Videoübertragung entwickelt wurden, wie z. B. adaptive Bitratenoptimierung, Videokomprimierung und Analysen, die sich auf das Zuschauererlebnis konzentrieren.

Struktur der Kosten

Vergleichen Sie die Preismodelle der Anbieter und berücksichtigen Sie dabei Faktoren wie Verkehrsvolumen, geografische Verteilung und Funktionsanforderungen. Einige CDNs bieten Mengenrabatte oder Rabatte für feste Nutzung an, die möglicherweise Ihren Nutzungsmustern entsprechen.

Support für erweiterte Funktionen

Vergewissern Sie sich, dass das von Ihnen gewählte CDN die Funktionen unterstützt, die Sie benötigen, z. B. Token-Authentifizierung, geografische Beschränkung, Zusammenklappen von Anfragen und korrekte Header-Weiterleitung.

Weitere Informationen zu bestimmten CDN-Anbietern und deren Integration finden Sie in den folgenden Ressourcen MediaTailor.

- [CloudFront-Integration](#) für Amazon CloudFront
- [CDN-Setup eines Drittanbieters](#) für CDN-Drittanbieter

Die folgenden Themen bieten umfassende Anleitungen zur Konfiguration MediaTailor mit einem CDN für optimale Leistung.

Planen Sie Ihre CDN-Integration für AWS Elemental MediaTailor

Mit einer CDN-Integration für AWS Elemental MediaTailor können Sie das Zuschauererlebnis verbessern und die Latenz reduzieren. Wenn Sie ein Content Delivery Network (CDN) implementieren, können Sie Inhalte von Orten aus bereitstellen, die Ihren Zuschauern näher sind. Dies gewährleistet schnellere Ladezeiten, bessere Skalierbarkeit und eine konsistente Anzeigenbereitstellung in verschiedenen geografischen Regionen.

Sie benötigen eine angemessene Planung, bevor Sie ein CDN mit AWS Elemental MediaTailor implementieren. Dieser Abschnitt führt Sie durch die wichtigsten Planungsbereiche. Sie gehen auf diese Bereiche ein, bevor Sie mit der eigentlichen Konfiguration beginnen. Diese Schritte helfen Ihnen dabei, ein optimales Seherlebnis für Ihr Publikum zu schaffen.

Wenn Sie Ihre Planung abgeschlossen haben, finden Sie Anweisungen MediaTailor zur step-by-step Implementierung unter [CDN integrieren mit](#).

Informationen zu MediaTailor Kontingenten, die sich auf Ihre CDN-Planung auswirken können, finden Sie unter [Kontingente in AWS Elemental MediaTailor](#). Informationen zu CloudFront Kontingenten finden Sie unter [Kontingente](#) im CloudFront Entwicklerhandbuch.

Bevor Sie mit der Planung beginnen, sollten Sie sich darüber im Klaren sein, wie die MediaTailor Interaktion mit einem CDN abläuft:

1. Die Zuschauer fordern Inhalte über dein CDN an, anstatt direkt von MediaTailor
2. Das CDN leitet Manifestanfragen zur Personalisierung weiter MediaTailor .
3. Das CDN speichert Inhaltssegmente und Anzeigensegmente von Edge-Standorten aus im Cache und stellt sie bereit.

Diese Architektur reduziert die Auslastung und stellt MediaTailor gleichzeitig sicher, dass Zuschauer personalisierte Anzeigen mit minimaler Latenz erhalten.

Wenn Sie die Manifest-Terminologie verstehen, können Sie Ihr CDN richtig konfigurieren. Verschiedene Streaming-Protokolle verwenden spezifische Manifeststrukturen, die sich darauf auswirken, wie Sie das Caching und das Routing einrichten:

- HLS-Manifeste — Wenn Sie mit HLS-Streams arbeiten, kümmern Sie sich um Folgendes:
 - Multivariante Playlist: Konfigurieren Sie Ihr CDN so, dass diese Manifeste der obersten Ebene zur Personalisierung weitergeleitet werden. MediaTailor

- **Medien-Playlist:** Lege die entsprechenden Caching-Regeln für diese Manifeste fest, die Links zu Inhaltssegmenten enthalten.
- **DASH-Manifeste** — Wenn Sie mit DASH-Streams arbeiten, kümmern Sie sich um Folgendes:
 - **MPD (Media Presentation Description):** Konfigurieren Sie Ihr CDN so, dass es diese Manifeste entsprechend Ihren Personalisierungsanforderungen verarbeitet.

Der CDN-Planungsprozess umfasst die folgenden wichtigen Schritte, die sich jeweils auf eine bestimmte Aufgabe konzentrieren:

- [Schätzen Sie die Datenverkehrsanforderungen für CDN und Integrationen ab MediaTailor:](#) Berechnen Sie die erwartete Parallelität Ihrer Zuschauer und die Bandbreitenanforderungen.
- [Konfigurieren Sie Optimierungsstrategien für CDN und Integrationen MediaTailor:](#) Konfigurieren Sie Ihr CDN für eine optimale Inhaltsbereitstellung und Anzeigenpersonalisierung.
- [Passen Sie die Planung für CDN und Integrationen an MediaTailor:](#) Passen Sie Ihre CDN-Strategie an Ihren spezifischen Workflow an. MediaTailor
- [Richten Sie die Überwachung und Skalierung für CDN und Integrationen ein MediaTailor:](#) Implementieren Sie Überwachungs- und Skalierungsstrategien für eine zuverlässige Leistung.
- [Optimieren Sie die Kosten für CDN und Integrationen MediaTailor:](#) Bringen Sie Leistung und Kosteneffizienz in Einklang.
- [Testen Sie Ihre Implementierung für CDN und Integrationen MediaTailor:](#) Testen Sie Ihre CDN-Integration vor der Produktionsbereitstellung gründlich.

Schätzen Sie die Datenverkehrsanforderungen für CDN und Integrationen ab MediaTailor

Um Ihre CDN-Integration (Content Delivery Network) genau zu dimensionieren mit AWS Elemental MediaTailor:

1. Berechnen Sie die erwartete Parallelität Ihrer Zuschauer anhand von historischen Daten oder ähnlichen Ereignissen. Planen Sie zusätzliche Kapazität ein, die über Ihren Ausgangswert hinausgeht, um unerwartete Leistungsspitzen zu bewältigen. Aktuelle Skalierungsempfehlungen erhalten Sie von Ihrem AWS Account-Team. Sie können auch [Kontingente in AWS Elemental MediaTailor](#) anzeigen.

2. Identifizieren Sie Muster mit Spitzenzugriffen und potenzielle Spitzen in Ihrem Inhaltsplan. Berücksichtigen Sie Faktoren wie:
 - Live-Sportveranstaltungen oder Staffelpremieren
 - Marketingkampagnen oder Werbeveranstaltungen
 - Zeitzonunterschiede für globale Zielgruppen
 - Zuschauermuster an Feiertagen oder zu bestimmten Jahreszeiten
3. Ermitteln Sie Ihre Bandbreitenanforderungen, indem Sie die Anzahl der Zuschauer mit den Stream-Bitraten multiplizieren. Arbeiten Sie mit Ihrem CDN-Anbieter zusammen, um die entsprechende Kapazität zu berechnen. Basieren Sie diese Berechnung auf Ihren spezifischen Inhalts-Bitraten und der erwarteten Zielgruppengröße. Fügen Sie den Overhead für Anzeigensegmente und Manifestanfragen hinzu, wie es Ihr Anbieter empfiehlt.
4. Arbeiten Sie mit Ihrem CDN-Anbieter zusammen, um eine ausreichende Edge-Kapazität in Ihren Zielregionen sicherzustellen.

Stellen Sie sicher, dass Ihre Kapazität für die Anzeigenschaltung der Nachfrage der Zuschauer entspricht, indem Sie die folgenden spezifischen Maßnahmen ergreifen:

1. Überprüfen Sie Ihr aktuelles Kontingent für Anfragen zur Anzeigenschaltung in der [Konsole „Servicekontingente“](#). Sehen Sie sich die aktuellen Servicebeschränkungen an, um zu erfahren, wie viele Nutzer gleichzeitig von Ihrer Konfiguration unterstützt werden können.
2. Bei Ereignissen mit hohem Datenaufkommen fordern Sie über die [Service Quotas-Konsole](#) ein erhöhtes Kontingent an.
3. Wenn Sie mit mehr als 500.000 gleichzeitigen Zuschauern rechnen, wenden Sie sich mindestens 2 Wochen vor Ihrer Veranstaltung an den [AWS-Support](#). Auf diese Weise können AWS Sie sicherstellen, dass ausreichend Kapazität für Ihre Anforderungen an die Anzeigenpersonalisierung vorhanden ist.

Weitere Informationen zur Implementierung der Kapazitätsplanung in Ihrem Workflow finden Sie unter Optimieren der Anzeigenbereitstellung [mit Prefetch-Planung](#) für Ereignisse mit hohem Besucheraufkommen.

Konfigurieren Sie Optimierungsstrategien für CDN und Integrationen MediaTailor

Wenn Sie Ihre Traffic-Schätzung abgeschlossen haben, konfigurieren Sie Ihr Content Delivery Network (CDN), um die Inhaltsbereitstellung und Anzeigenpersonalisierung mit zu optimieren. AWS Elemental MediaTailor Diese Optimierungen tragen dazu bei, eine reibungslose Wiedergabe zu gewährleisten und gleichzeitig gezielte Werbung aufrechtzuerhalten.

Implementieren Sie die folgenden spezifischen CDN-Optimierungen:

1. Konfigurieren Sie die Origin Shield-Funktionen in Ihrem CDN, um die Belastung zu reduzieren MediaTailor und die Caching-Effizienz zu verbessern. Origin Shield fungiert als zwischengeschaltete Caching-Ebene, die:

- Konsolidieren Sie mehrere Viewer-Anfragen zu einer einzigen ursprünglichen Anfrage
- Reduzieren Sie die Anzahl redundanter Anfragen auf MediaTailor
- Verbessern Sie die Antwortzeiten für den zwischengespeicherten Inhalt

Implementierungsdetails zur Einrichtung von Origin Shield mit CloudFront findest du unter [Origin Shield verwenden](#) im CloudFront Entwicklerhandbuch.

2. Lege die entsprechenden Werte für Time To Live (TTL) für verschiedene Inhaltstypen fest. TTL bestimmt, wie lange das CDN Inhalte zwischenspeichert. Nach Ablauf dieser Zeit fordert das CDN eine neue Kopie vom Ursprung an:

- Manifestiert:
 - 30 Sekunden für das Einfügen der Anzeige
 - 5-10 Sekunden für die Kanalmontage

Bei der Anzeigeneinfügung werden MediaTailor Manifeste mit Werbung angezeigt, die auf den Betrachter zugeschnitten ist. Wenn eine Playlist oder MPD zwischengespeichert und auf dem falschen Wiedergabegerät bereitgestellt wird, kann es auf dem Gerät zu Wiedergabe- oder Tracking-Problemen kommen.

- Inhaltssegmente: 24 oder mehr Stunden (diese ändern sich selten und du kannst sie aggressiv zwischenspeichern, um die ursprüngliche Auslastung zu reduzieren)
- Anzeigensegmente: 24 oder mehr Stunden (Anzeigeninhalte werden in der Regel von mehreren Zuschauern wiederverwendet und können für längere Zeiträume zwischengespeichert werden)

Umfassende TTL-Empfehlungen und Strategien zur Caching-Optimierung für alle MediaTailor Workflows finden Sie unter [Caching-Optimierung für CDN und Integrationen MediaTailor](#)

Ausführliche Anweisungen zur Konfiguration des Cache-Verhaltens finden Sie unter [Einstellungen für das Cache-Verhalten](#) im CloudFront Entwicklerhandbuch. CloudFront

3. Stellen Sie CDN-Edge-Knoten in der Nähe Ihrer Zuschauerpopulationen bereit. Arbeiten Sie mit Ihrem CDN-Anbieter zusammen, um:
 - Identifizieren Sie anhand der demografischen Daten der Zuschauer die optimalen Standorte der Edge-Knoten
 - Sorgen Sie für ausreichende Kapazitäten in jeder Region
 - Überwachen Sie die Edge-Leistung und passen Sie sie nach Bedarf an

Anleitungen zur Implementierung finden Sie unter [CloudFront Edge-Standorte](#), um die verfügbaren Edge-Standorte für Ihre Zielgruppenregionen zu ermitteln.

4. Für globale Zielgruppen sollten Sie die Implementierung einer Multi-CDN-Strategie in Betracht ziehen. Dieser Ansatz:
 - Nutzt mehrere CDN-Anbieter, um die Zuverlässigkeit zu verbessern
 - Leitet Zuschauer zum CDN mit der besten Leistung für ihren Standort
 - Bietet Failover-Optionen bei CDN-Ausfällen
 - Kann die Kosten optimieren, indem verschiedene Preismodelle genutzt werden

Einzelheiten zur Implementierung finden Sie im Blog AWS Networking & Content Delivery unter [Multi-CDN-Strategien](#).

Passen Sie die Planung für CDN und Integrationen an MediaTailor

Verschiedene AWS Elemental MediaTailor Workflows haben besondere Anforderungen, die sich auf die Planung von Content Delivery Networks (CDN) auswirken. Workflow-spezifische Anleitungen finden Sie unter [Arbeiten mit Konfigurationen](#).

Passen Sie Ihren Kapazitätsplan an Ihren spezifischen MediaTailor Workflow an:

Für Workflows zum Einfügen von MediaTailor Anzeigen

1. Konfigurieren Sie Ihr CDN so, dass es personalisierte Manifeste ohne Zwischenspeicherung verarbeitet. Dadurch wird sichergestellt, dass jeder Zuschauer einzigartige, zielgerichtete Anzeigen erhält. Die Anzeigen basieren auf ihrem Profil und ihrem Betrachtungskontext.
2. Passen Sie Ihren Ad Decision Server (ADS) so an, dass er hohe Anforderungsvolumen bewältigen kann. Anleitungen zur ADS-Konfiguration finden Sie unter [Anzeigeneinfügung mit MediaTailor](#). Beachten Sie:
 - Anforderungen an die Reaktionszeit für Ihren Anwendungsfall
 - Erwartete Kapazität für gleichzeitige Zuschauer
 - Anforderungen an Redundanz und Failover
 - Anforderungen an die geografische Verteilung
3. Implementieren Sie das Zusammenklappen von Anfragen auf CDN-Ebene, um synchronisierte Werbeunterbrechungsanfragen zu bearbeiten. Beim Zusammenklappen von Anfragen werden mehrere identische Anfragen zu einer einzigen ursprünglichen Anfrage zusammengefasst. Dies ist entscheidend bei:
 - Live-Sportveranstaltungen, bei denen viele Zuschauer gleichzeitig auf Werbeunterbrechungen klicken
 - Premieren beliebter Fernsehsendungen mit synchronisierten Werbeunterbrechungen
 - Aktuelle Nachrichtenereignisse, die plötzliche Zuschauerzahlen nach sich ziehen

Einzelheiten zur Implementierung finden Sie unter [Origin-Failover](#) zur Konfiguration der Bearbeitung von Anfragen bei Spitzenlasten.

Für Workflows zur MediaTailor Kanalmontage

1. Berechnen Sie die Kapazitätsanforderungen anhand der Anzahl der Kanäle und ihrer Bitraten. Hinweise zur Kapazitätsplanung für die Kanalmontage finden Sie unter [Kanalmontage in MediaTailor](#). Beachten Sie:
 - Gesamtzahl der Kanäle
 - Anforderungen an die Bitrate pro Kanal
 - Erwartetes gleichzeitiges Laden des Viewers
 - Anforderungen an die geografische Verteilung

2. Konfigurieren Sie Ihr CDN so, dass es vorhersehbare Verkehrsmuster auf der Grundlage veröffentlichter Zeitpläne verarbeitet. Die Kanalzusammenstellung weist in der Regel vorhersehbarere Muster auf als die Anzeigeneinfügung aus folgenden Gründen:
 - Die Zeitpläne für die Programmierung sind im Voraus bekannt
 - Das Verhalten der Zuschauer folgt etablierten Mustern
 - Der Inhalt ändert sich nicht dynamisch pro Zuschauer
3. Stellen Sie sicher, dass Ihr Absender über ausreichend Bandbreite verfügt, um eine konsistente Kanalausgabe zu gewährleisten. Implementieren Sie:
 - Redundante Ursprungsserver für Kanäle mit hoher Verfügbarkeit
 - Automatischer Failover zwischen primärem Ursprung und Backup-Ursprung
 - Überwachung zur Erkennung von Leistungsproblemen am Ursprung

Hinweise zur Implementierung finden Sie unter [Einrichtung von Ursprungsredundanz](#) zur Schaffung einer belastbaren Ausgangsinfrastruktur.

Für kombinierte Workflows MediaTailor

1. Passen Sie Ihre Infrastruktur so an, dass sie die kombinierten Verkehrsmuster beider Dienste bewältigen kann. Hinweise zu kombinierten Workflows finden Sie unter [AWS Elemental MediaTailor Zum Einfügen von Anzeigen verwenden](#). Bedenken Sie Folgendes:
 - Grundvoraussetzungen für die Kanalmontage
 - Overhead-Anforderungen für das Einfügen von Anzeigen
 - Verkehrsmuster zu Spitzenzeiten
 - Redundanzanforderungen
2. Konfigurieren Sie separate CDN-Verhaltensweisen für die lineare Bereitstellung von Inhalten und die dynamische Anzeigeneinblendung. Diese Trennung ermöglicht Ihnen:
 - Optimieren Sie die Caching-Richtlinien für jeden Inhaltstyp unabhängig voneinander
 - Leitet Anfragen je nach Inhaltstyp an geeignete Quellen weiter
 - Überwachen Sie die Leistungskennzahlen für jeden Workflow separat
3. Richten Sie das richtige Routing zwischen Edge und CDNs Origin ein, um eine optimale Leistung aufrechtzuerhalten. Erwägen Sie die Verwendung von:
 - Verschiedene Quellpfade für Inhaltssegmente (/content/*) und Anzeigensegmente (/ads/*)

- Geografisches Routing zur Optimierung der Latenz für verschiedene Regionen

Einzelheiten zur Implementierung finden Sie unter [Cache-Verhalten konfigurieren](#), um pfadbasierte Routing- und Caching-Regeln einzurichten.

Richten Sie die Überwachung und Skalierung für CDN und Integrationen ein MediaTailor

Effektive Überwachungs- und Skalierungsstrategien sind entscheidend für die Aufrechterhaltung einer optimalen Leistung und eines optimalen Zuschauererlebnisses bei der Integration Ihres AWS Elemental MediaTailor Content Delivery Network (CDN). Implementieren Sie diese Ansätze, um sicherzustellen, dass Ihre CDN-Integration zuverlässig und skalierbar funktioniert.

Implementieren Sie die folgenden Überwachungs- und Skalierungsstrategien:

1. Konfigurieren Sie die Überwachung für diese wichtigen Kennzahlen. Hinweise zu geeigneten Zielwerten finden Sie unter [Monitoring MediaTailor with Amazon CloudWatch](#) und die Best Practices Ihres CDN-Anbieters:
 - CDN-Cache-Trefferquoten (legen Sie Basiskennzahlen und Ziele auf der Grundlage Ihres Inhaltstyps und Ihrer Bereitstellungsmuster fest)
 - Volumen der ursprünglichen Anfragen (überwachen Sie die Muster während des normalen Betriebs, um Ausgangswerte für die Erkennung von Anomalien festzulegen)
 - Fehlerquoten nach Fehlertyp (definieren Sie Schwellenwerte auf der Grundlage Ihrer Service-Level-Ziele und Best Practices) MediaTailor
 - Reaktionszeiten (legen Sie geeignete Latenzziele auf der Grundlage Ihrer Anforderungen an das Zuschauererlebnis und der geografischen Verteilung fest)

Detaillierte Implementierungsanweisungen finden Sie unter [CloudWatch Dashboards erstellen](#), um Ihre MediaTailor und CDN-Metriken gemeinsam zu visualisieren.

2. Richten Sie Warnmeldungen für unerwartete Verkehrsmuster oder Leistungseinbußen ein. Konfigurieren Sie Schwellenwerte auf der Grundlage Ihrer Basiskennzahlen und Service-Level-Ziele. Anleitungen zum Einrichten von Benachrichtigungen finden Sie unter [CloudWatch Amazon-Alarme erstellen](#). Erwägen Sie die Überwachung:
 - Signifikante Abweichungen von den Basis-Cache-Trefferquoten (normalerweise wird eine Warnung angezeigt, wenn sie unter 85 bis 90% fällt)

- Plötzlicher Anstieg des ursprünglichen Anforderungsvolumens (Warnung bei einem Anstieg von 30% oder mehr gegenüber dem Ausgangswert)
- Spitzenwerte bei der Fehlerrate, die Ihre definierten Schwellenwerte überschreiten (in der Regel 1—2% bei 4xx-Fehlern, 0,5% bei 5xx-Fehlern)
- Verschlechterung der Reaktionszeit über akzeptable Werte (typischerweise > 500 ms für Manifeste, > 200 ms für Segmente)

Implementierungsbeispiele finden Sie unter [CloudWatch Konzepte](#) für die Erstellung effektiver Monitoring-Dashboards und Warnmeldungen.

3. Erstellen Sie Skalierungspläne für vorhersehbare Ereignisse mit hohem Besucheraufkommen. Ihre Pläne sollten die folgenden Schlüsselemente enthalten:
 - Erhöhung der Kapazität vor der Veranstaltung (24-48 Stunden vor Veranstaltungsbeginn)
 - Schrittweiser Zuwachs der Zuschauer (in der Regel 10-20% der erwarteten Zuschauerzahl pro 5-Minuten-Intervall)
 - Regionale Kapazitätsverteilung auf der Grundlage der Zuschauerzahl (Zuteilung der Kapazität proportional zur erwarteten regionalen Zuschauerzahl)
 - Verfahren zur Skalierung nach der Veranstaltung (Aufrechterhaltung der Spitzenkapazität für 30-60 Minuten nach Abschluss der Veranstaltung)

Implementierungsleitlinien zur Skalierung bei stark frequentierten Ereignissen finden Sie im Media Blog unter [Einrichten eines stabilen end-to-end Live-Workflows](#). AWS

4. Implementieren Sie Failover- und Redundanzmaßnahmen für kritische Streams, darunter:
 - CDN-Bereitstellungen in mehreren Regionen (mindestens zwei Regionen für kritische Inhalte)
 - Backup-Ursprungsserver (konfiguriert mit automatisierten Integritätsprüfungen alle 30 Sekunden)
 - Automatisierte Failover-Trigger auf der Grundlage von Integritätsprüfungen (in der Regel nach 2-3 fehlgeschlagenen Prüfungen)
 - Wiederherstellungsverfahren für verschiedene Ausfallszenarien (dokumentiert mit spezifischen Reaktionszeitzielen)

Ausführliche Implementierungsschritte finden Sie unter [Optimieren der Hochverfügbarkeit mit CloudFront Origin-Failover](#).

Optimieren Sie die Kosten für CDN und Integrationen MediaTailor

Die Kosten für Content Delivery Network (CDN) können je nach Verkehrsmuster, geografischer Verteilung und Nutzung von Funktionen erheblich variieren. AWS Elemental MediaTailor Aktuelle Preisinformationen finden Sie unter [CloudFront Preise](#). Sie können auch in der Dokumentation Ihres CDN-Anbieters nachschlagen.

Bringen Sie die Leistung mit der Kosteneffizienz in Einklang, indem Sie die folgenden Strategien anwenden:

1. Analysieren Sie Ihre CDN-Datenverkehrsmuster, um die kostengünstigste Preisstufe auszuwählen. Erkundigen Sie sich bei Ihrem CDN-Anbieter über die folgenden Faktoren:

- Datenübertragungsvolumen nach Region und Zeitraum
- Muster für Manifeste und Segmente anfordern
- Anforderungen an die geografische Verteilung
- Spitzennutzungsmuster im Vergleich zu durchschnittlichen Nutzungsmustern

Wenn Sie Hilfe bei der Kostenanalyse benötigen, verwenden Sie den [AWS Preisrechner](#), um Ihre CloudFront Kosten auf der Grundlage Ihrer spezifischen Nutzungsmuster zu schätzen.

2. Für vorhersehbare Workloads sollten Sie die Vereinbarungen mit reservierter Kapazität mit Ihrem CDN-Anbieter prüfen. Diese Vereinbarungen können unter anderem folgende Vorteile bieten:

- Ermäßigte Tarife für festgelegte Nutzungsvolumen
- Vorhersehbare monatliche Kosten für die Budgetierung
- Vorrangige Unterstützung und Kapazitätszuweisung

Erkundigen Sie sich bei Ihrem CDN-Anbieter, ob die reservierte Kapazität für Ihr Nutzungsverhalten geeignet ist. Weitere Informationen finden Sie unter [CloudFront Premium-Funktionen](#). CloudFront Hier finden Sie Informationen zu den Optionen für reservierte Kapazitäten.

3. Optimieren Sie die Kosten für ausgehenden Datenverkehr, indem Sie den Datenverkehr zwischen Ihrem CDN-Anbieter MediaTailor und Ihrem CDN-Anbieter ausgleichen. Zu den Strategien gehören:

- Maximieren Sie die Cache-Trefferquoten, um ursprüngliche Anfragen zu reduzieren
- Verwenden Sie Origin Shield, um Anfragen zu konsolidieren
- Implementieren Sie Komprimierung, um das Datenübertragungsvolumen zu reduzieren

• Wählen Sie CDN-Regionen, die den AWS regionalen Preisen entsprechen

Implementierungshinweise zur Komprimierung finden Sie im CloudFront Entwicklerhandbuch unter [Bereitstellen komprimierter Dateien](#).

4. Implementieren Sie geeignete Caching-Strategien für verschiedene Inhaltstypen, um die Anzahl der ursprünglichen Anfragen zu reduzieren. Hinweise zur Cache-Optimierung finden Sie unter [Verbesserung der Cache-Trefferquoten](#). Konzentrieren Sie sich auf:

- Inhaltssegmente (Sie können diese für längere Zeiträume zwischenspeichern)
- Anzeigensegmente (werden in der Regel von mehreren Zuschauern wiederverwendet)
- Statische Inhalte wie Player-Dateien und Bilder

Verbesserte Cache-Trefferquoten reduzieren die Herstellungskosten erheblich. Arbeiten Sie mit Ihrem CDN-Anbieter zusammen, um die Cache-Konfigurationen für Ihre spezifischen Inhaltsmuster zu optimieren. Ausführliche Implementierungsschritte finden Sie unter [Cache-Verhalten konfigurieren](#).

Testen Sie Ihre Implementierung für CDN und Integrationen MediaTailor

Sorgen Sie für zuverlässige Zuschauererlebnisse, indem Sie Ihre CDN-Integration (AWS Elemental MediaTailor Content Delivery Network) vor der Produktionsbereitstellung gründlich testen. Richtiges Testen hilft dabei, Probleme zu erkennen und zu lösen, bevor sie sich auf Ihr Publikum auswirken. Anleitungen zu Testmethoden finden Sie unter [CloudFront Distributionen testen](#). Sie können auch die Testdokumentation Ihres CDN-Anbieters lesen.

Folgen Sie diesen Testschritten, um Ihre CDN-Integration zu validieren:

1. Erstellen Sie eine Testumgebung, die Ihre Produktionskonfiguration widerspiegelt. Schließen Sie ein:
 - Identische CDN-Einstellungen und Cache-Verhalten
 - Fügen Sie repräsentative Inhalte mit verschiedenen Bitraten und Formaten hinzu
 - Konfigurieren Sie einen Test- und Entscheidungsserver mit Beispiel-Anzeigenantworten
 - Richten Sie Überwachungs- und Warnkonfigurationen ein

Anleitungen zur step-by-step Implementierung finden Sie unter [Erstellen einer Staging-Distribution](#) im CloudFront Entwicklerhandbuch.

2. Führen Sie Lasttests durch, um Ihre Kapazitätsschätzungen zu überprüfen. Anleitungen zu Lasttests finden Sie unter [Überwachung MediaPackage](#). Die Testszenarien sollten Folgendes beinhalten:

- Schrittweise Erhöhung der Zuschauerzahl (in der Regel alle 5 Minuten um 10 bis 20% des erwarteten Spitzenwerts)
- Plötzliche Verkehrsspitzen auf der Grundlage Ihrer erwarteten Spitzenlasten (simulieren Sie, dass 50% der Zuschauer zu Spitzenzeiten innerhalb von 60 Sekunden beitreten)
- Verlängerte Spitzenlastperioden (Aufrechterhaltung der Spitzenlast für mindestens 30-60 Minuten)
- Geografische Verteilung, die zu Ihrer Zielgruppe passt (verteilen Sie den Test-Traffic entsprechend den erwarteten Standorten der Zuschauer)

Stellen Sie sicher, dass die Antwortzeiten unter den Zielschwellenwerten bleiben. In der Regel bedeutet dies weniger als 500 ms für Manifeste und weniger als 200 ms für Segmente. Die Fehlerquoten sollten unter 1% bleiben. Einzelheiten zur Implementierung von Tools und Methoden für Lasttests finden Sie im Blog AWS Networking & Content Delivery CloudFront unter [Auslastungstests mit](#).

3. Testen Sie Failover-Szenarien, um die Zuverlässigkeit sicherzustellen. Simulieren Sie:

- Ausfälle des Origin-Servers (vollständiger Ausfall und teilweise Verschlechterung)
- Ausfälle am CDN-Edge-Standort (Test mit Weiterleitung des Datenverkehrs zu Backup-Standorten)
- Nichtverfügbarkeit des Ad Decision Servers (Test mit Timeouts von 5-10 Sekunden)
- Probleme mit der Netzwerkkonnektivität (simulieren Sie Paketverlust und erhöhte Latenz)

Arbeiten Sie mit Ihrem CDN-Anbieter zusammen, um geeignete Ziele für die Failover-Reaktionszeit für Ihren Anwendungsfall festzulegen. In der Regel bedeutet dies weniger als 3 Sekunden, bis der Failover abgeschlossen ist. Anleitungen zur Implementierung von Failover-Tests finden Sie unter [Origin Failover im CloudFront Developer](#) Guide.

4. [Implementieren Sie bei größeren Ereignissen schrittweise Hochlaufstrategien auf AWS der Grundlage der Richtlinien für Lasttests:](#)

- Staffelnde Startzeiten für verschiedene Zielgruppensegmente. Erlauben Sie beispielsweise zunächst Premium-Abonnenten mit 15-minütigen Intervallen zwischen den Zielgruppen.
- Verwenden Sie Techniken zum Vorwärmen, um die Belastung schrittweise zu erhöhen. Das Vorwärmen beinhaltet:
 - Füllen Sie die CDN-Caches 2-4 Stunden vor der Veranstaltung mit beliebten Inhalten

- Erhöhen Sie den synthetischen Traffic schrittweise auf 20 bis 30% des erwarteten Spitzenwerts, bis die Systeme wieder betriebsbereit sind
- Testen Sie alle Komponenten unter realistischen Lastbedingungen mit realem Inhalt
- Überwachen Sie die Systemleistung während der gesamten Anlaufphase und verfolgen Sie dabei:
 - Cache-Trefferquoten und Reaktionszeiten (Ziel-Trefferquote > 90%, Reaktionszeit < 500 ms)
 - Fehlerraten und Ausgangslast (Fehlerraten unter 1% beibehalten, Ursprungs-CPU < 70%)
 - Fehlerraten und Ursprungslast (Fehlerraten unter 1% beibehalten, ursprüngliche CPU 98% erfolgreiche Personalisierung)
 - Kennzahlen zur Nutzererfahrung (Ziel <2 Sekunden Startzeit, < 0,5% Zurückweisung)
- Halten Sie einen Notfallplan für unerwartete Verkehrsspitzen bereit. Ihr Plan sollte die folgenden wesentlichen Komponenten enthalten:
 - Verfahren zur Kapazitätsskalierung in Notfällen mit dokumentierten Schritten zur Kapazitätserhöhung um 50 bis 100% innerhalb von 15 Minuten
 - Backup-CDN-Aktivierungsprotokolle mit der Fähigkeit, 20-50% des Datenverkehrs auf sekundäres CDN zu verlagern
 - Vereinfachtes Fallback für die Anzeigeneinblendung, um die Targeting-Parameter von 10 oder mehr auf 3 bis 5 wichtige Parameter zu reduzieren
 - Kommunikationspläne für Stakeholder mit vordefinierten Benachrichtigungsvorlagen und Kontaktlisten

Fahren Sie nach Abschluss Ihrer Tests mit den Schritten zur [Implementierung Ihrer CDN-Integration](#) für die Produktionsbereitstellung fort.

Beheben Sie häufig auftretende Probleme bei CDN und Integrationen MediaTailor

Bewältigen Sie häufig auftretende Probleme bei der Integration von Content Delivery Network (CDN), AWS Elemental MediaTailor bevor sie sich auf Ihre Zuschauer auswirken. Dieser Abschnitt hilft Ihnen dabei, typische Probleme zu identifizieren und zu lösen, die bei der CDN-Integration mit auftreten.

AWS Elemental MediaTailor

Umfassende Anleitungen zur Fehlerbehebung finden Sie unter [Problembehandlung MediaTailor und CloudFront Problembehandlungsverteilungen](#).

Behebung MediaTailor offensichtlicher Lieferprobleme

Wenn Zuschauer Probleme mit der Wiedergabe haben oder falsche Werbung sehen, überprüfe diese häufigen Probleme im Zusammenhang mit dem Manifest:

- **Falsche Caching-Einstellungen:** Wenn dein CDN personalisierte Manifeste zwischenspeichert, sehen Zuschauer möglicherweise Werbung, die für andere Nutzer bestimmt ist.

Lösung: Konfigurieren Sie Ihr CDN mit einer Cache-TTL von 0 für Manifestanfragen an MediaTailor

- **Fehler bei der ursprünglichen Anfrage:** Wenn Ihr CDN keine Verbindung herstellen kann MediaTailor, schlagen Manifestanfragen fehl.

Lösung: Überprüfen Sie die Netzwerkkonnektivität zwischen Ihrem CDN und MediaTailor Stellen Sie sicher, dass Ihr CDN die richtigen Header ordnungsgemäß weiterleitet.

- **Probleme mit Sitzungsparametern:** Fehlende oder falsche Sitzungsparameter können zu Personalisierungsfehlern führen.

Lösung: Stellen Sie sicher, dass Ihr Player alle erforderlichen Sitzungsparameter korrekt an die Manifestanfragen anhängt.

Behebung von Problemen bei der MediaTailor Segmentzustellung

Wenn Inhalts- oder Anzeigensegmente nicht richtig geladen werden, untersuchen Sie die folgenden häufig auftretenden Probleme:

- **Umschreiben von Segmentpfaden:** Eine falsche CDN-Konfiguration kann dazu führen, dass das Segment falsch umgeschrieben wird. URLs

Lösung: Stellen Sie sicher, dass Ihr CDN das Segment korrekt verarbeitet. URLs Stellen Sie sicher, dass Pfade nicht so geändert werden, dass Verweise beschädigt werden.

- **CORS-Konfiguration:** Fehlende oder falsche CORS-Header können verhindern, dass Browser Segmente laden.

Lösung: Konfigurieren Sie Ihr CDN so, dass die entsprechenden CORS-Header für Segmentanfragen übergeben werden.

- **Cache-Fehlschläge:** Bei Ereignissen mit hohem Datenaufkommen können mehrere Cache-Fehlschläge die Ursprungsserver überfordern.

Lösung: Implementieren Sie Funktionen zum Zusammenklappen von Anfragen und zum Origin Shield, um die Auslastung der Quellen bei Datenverkehrsspitzen zu reduzieren.

Behebung von MediaTailor CDN-Leistungsproblemen

Wenn Zuschauer zwischenspeichern oder langsam geladen werden, überprüfen Sie diese leistungsbezogenen Probleme:

- Niedrige Cache-Trefferquote: Wenn dein CDN häufig Inhalte vom Ursprungsserver anfordert, wird die Leistung beeinträchtigt.

Lösung: Analysieren Sie die Cache-Trefferquoten nach Inhaltstyp und passen Sie die TTL-Einstellungen an, um die Caching-Effizienz zu verbessern.

- Geografische Verteilung: Bei Zuschauern, die weit von CDN-Edge-Standorten entfernt sind, kann es zu einer erhöhten Latenz kommen.

Lösung: Überprüfen Sie die Verteilung Ihrer CDN-Edge-Standorte. Erhöhen Sie die Kapazität in Regionen mit hoher Zuschauerkonzentration.

- Kapazitätsbeschränkungen bei der Herkunft: Wenn Ihre Ursprungsserver überlastet sind, erhöhen sich die Antwortzeiten.

Lösung: Implementieren Sie die Beschränkung der ursprünglichen Anfragen. Sie können auch die Ausgangskapazität erhöhen oder das Caching verbessern, um die Ausgangslast zu reduzieren.

Weitere Unterstützung bei der Fehlerbehebung finden Sie unter [Problembehandlung MediaTailor](#) oder wenden Sie sich an den AWS Support. Implementierungsleitfäden zur Lösung häufiger CDN-Probleme finden Sie im Media Blog unter [Debuggen Ihres Content Delivery Network](#). AWS

Richten Sie die CDN-Integration ein mit MediaTailor

Dieser Abschnitt enthält Anleitungen zur AWS Elemental MediaTailor Integration in ein Content Delivery Network (CDN).

Eine effektive CDN-Integration mit MediaTailor ist für die Bereitstellung hochwertiger Streaming-Erlebnisse mit personalisierten Anzeigen in großem Maßstab unerlässlich. Dieser Leitfaden führt Sie durch den gesamten Prozess der Einrichtung, Konfiguration und Optimierung Ihrer CDN-Integration.

Weitere Informationen finden Sie unter den folgenden Links:

- Informationen zur Weitergabe von Abfrageparametern CDNs für Autorisierung und Anzeigenausrichtung finden Sie unter [MediaTailor Manifest-Abfrageparameter](#).
- Informationen zur erweiterten Weiterleitung mithilfe dynamischer Variablen und Konfigurationsaliasnamen finden Sie unter [MediaTailor dynamische Anzeigenvariablen für ADS-Anfragen](#).
- Informationen zum Erstellen von MediaTailor Konfigurationen finden Sie unter [AWS Elemental MediaTailor Zum Einfügen von Anzeigen verwenden](#).
- Informationen zum Erstellen einer CloudFront Distribution finden Sie unter [Creating a Distribution](#) im CloudFront Developer Guide.

Aus Gründen der Übersichtlichkeit wird in diesem Handbuch die folgende Terminologie verwendet, um die verschiedenen Arten von Manifesten zu beschreiben:

- HLS-Manifeste:
 - Multivariante Playlist: Das Manifest der obersten Ebene, das Links zu Medien-Playlisten enthält
 - Medien-Playlist: Das Manifest der zweiten Ebene mit Links zu Inhaltssegmenten
- DASH-Manifestationen:
 - MPD (Media Presentation Description): Die Standardbezeichnung für DASH-Manifeste

Die Integration MediaTailor mit einem CDN bietet die folgenden Vorteile:

- Geringere Latenz für Zuschauer
- Verbesserte Skalierbarkeit für Ereignisse mit hohem Besucheraufkommen
- Verbesserte Zuverlässigkeit durch redundante Bereitstellungspfade
- Optimierte Kosten durch Reduzierung des Ausgangsverkehrs
- DDoBesserer Schutz vor S-Angriffen

Komponenten und Anforderungen für die CDN-Integration

Eine erfolgreiche CDN-Integration mit MediaTailor beinhaltet die Konfiguration der folgenden Schlüsselkomponenten:

Verhalten beim CDN-Routing

Regeln, die festlegen, wie verschiedene Arten von Anfragen (Manifeste, Inhaltssegmente, Anzeigensegmente) über Ihr CDN weitergeleitet werden.

CDN-Zuordnung in MediaTailor

Eine Konfiguration MediaTailor, die sicherstellt, dass Manifeste auf Ihre CDN-Domain verweisen, anstatt direkt auf Originalserver zu verweisen.

Sicherheitseinstellungen

Konfigurationen zum Schutz Ihrer Inhalte und Infrastruktur, einschließlich Transportsicherheit, Zugriffskontrolle und Überwachung.

Testen und Validieren

Verfahren zur Überprüfung, ob Ihre CDN-Integration ordnungsgemäß funktioniert, bevor Sie sie in der Produktion einsetzen.

Voraussetzungen für die CDN-Integration

Bevor Sie Ihre CDN-Integration konfigurieren, stellen Sie sicher, dass Sie über Folgendes verfügen:

1. Eine MediaTailor Konfiguration mit den folgenden Einstellungen:

- Ihr Inhaltsursprung als Inhaltsquelle
- Ihr ADS als Entscheidungsserver für Werbeanzeigen

Sie benötigen auch den Ursprung und ADS URLs für die CDN-Integrationsschritte.

2. Zugriff auf die Konfigurationsoberfläche Ihres CDN

3. Verständnis der spezifischen Terminologie Ihres CDN für Verhalten, Regeln und Cache-Einstellungen

4. Kenntnis Ihrer Inhaltsstruktur, einschließlich der für Segmente verwendeten Dateierweiterungen (z. B. .ts, .mp4 oder .m4s)

Schritte zur Einrichtung der CDN-Integration

Der Prozess der Integration MediaTailor mit einem CDN folgt diesen allgemeinen Schritten:

1. Konfigurieren Sie das CDN-Routing-Verhalten — Richten Sie Ihr CDN so ein, dass verschiedene Arten von Anfragen entsprechend weitergeleitet werden.
2. Konfigurieren Sie das CDN-Mapping unter MediaTailor — Aktualisieren Sie Ihre MediaTailor Konfiguration, um Ihre CDN-Domainnamen zu verwenden.
3. Implementieren Sie bewährte Sicherheitsmethoden — Konfigurieren Sie Sicherheitseinstellungen, um Ihre Inhalte und Infrastruktur zu schützen.
4. Testen Sie Ihre Integration — Stellen Sie sicher, dass Ihre CDN-Integration ordnungsgemäß funktioniert.

Erforderliche Header für MediaTailor die CDN-Integration

MediaTailor Damit Ihr CDN korrekt funktioniert, müssen Sie Ihr CDN so konfigurieren, dass es bestimmte HTTP-Header weiterleitet. Diese Header sind für eine einwandfreie Funktionalität, einschließlich Komprimierung, Geräteerkennung, Anzeigenpersonalisierung und Geo-Targeting, unerlässlich.

Konfigurieren Sie Ihr CDN so, dass die folgenden Header weitergeleitet werden an: MediaTailor

Accept-Encoding

Zweck: Für die Komprimierungsfunktion erforderlich

Details: Dieser Header gibt an MediaTailor , welche Komprimierungsmethoden der Client unterstützt. MediaTailor verwendet diese Informationen, um Manifeste nach Möglichkeit zu komprimieren, wodurch die Bandbreitennutzung reduziert und die Leistung verbessert wird. Ältere Geräte, die keine Komprimierung unterstützen, senden diesen Header nicht und MediaTailor geben unkomprimierte Manifeste zurück.

User-Agent

Zweck: Erforderlich für die Geräteerkennung und die Personalisierung von Werbung

Details: MediaTailor Analysiert den User-Agent-Header, um den Gerätetyp, den Browser und die Funktionen des Client-Geräts zu identifizieren. Diese Informationen werden für gezielte Werbung, gerätespezifische Optimierungen und zur Sicherstellung der Kompatibilität mit verschiedenen Playback-Clients verwendet.

Host

Zweck: Erforderlich für die korrekte Weiterleitung von Anfragen

Details: Der Host Header stellt sicher, dass Anfragen an den richtigen MediaTailor Endpunkt weitergeleitet werden. Dies ist besonders wichtig in Umgebungen mit mehreren Mandanten und bei der Verwendung benutzerdefinierter Domänenkonfigurationen.

Viele CDNs, darunter Amazon CloudFront, leiten den Host Header standardmäßig nicht weiter. Für CloudFront Benutzer: Anweisungen [Cache-Verhalten konfigurieren](#) zur Konfiguration finden Sie unter.

X-Forwarded-For

Zweck: Erforderlich für Client-IP-Erkennung und Geo-Targeting

Details: Dieser Header behält die ursprüngliche Client-IP-Adresse bei, wenn Anfragen Ihr CDN passieren. MediaTailor verwendet diese Informationen für die geografische Ausrichtung von Anzeigen, Analysen und die Einhaltung regionaler Inhaltsbeschränkungen.

Important

Alle vier Header sind für die volle MediaTailor Funktionalität erforderlich. Wenn einer dieser Header fehlt, kann dies zu einer eingeschränkten Funktionalität führen, einschließlich:

- Manifeste können nicht komprimiert werden (Accept-Encoding fehlt)
- Schlechte Anzeigenausrichtung und Probleme mit der Gerätekompatibilität (fehlender User-Agent)
- Fehler beim Routing von Anfragen (fehlender Host)
- Ungenaueres Geo-Targeting und Analysen (fehlen) X-Forwarded-For

CDN-spezifische Konfigurationsanweisungen finden Sie in den Abschnitten Routing-Verhalten und Caching, die auf diese Header-Liste verweisen.

Die folgenden Themen enthalten detaillierte Anweisungen für jeden Aspekt der CDN-Integration.

Themen

- [Richten Sie das CDN-Routing-Verhalten ein für MediaTailor](#)
- [Richten Sie das CDN-Mapping ein in MediaTailor](#)
- [Bewährte Sicherheitsmethoden für die CDN-Integration für MediaTailor](#)

Richten Sie das CDN-Routing-Verhalten ein für MediaTailor

In diesem Abschnitt wird erklärt, wie Sie Ihr Content Delivery Network (CDN) so einrichten, dass verschiedene Arten von Anfragen entsprechend für die AWS Elemental MediaTailor Integration weitergeleitet werden. Durch die richtige Routingkonfiguration wird sichergestellt, dass Manifestanfragen, Inhaltssegmente und Anzeigensegmente korrekt behandelt werden.

Die Konfiguration des CDN-Routing-Verhaltens ist ein entscheidender Schritt bei der Schaffung einer effizienten Pipeline für die Inhaltsbereitstellung. Durch die Einrichtung spezifischer Routing-Regeln für verschiedene Inhaltstypen können Sie das Caching optimieren, die Bereitstellungsleistung verbessern und sicherstellen, dass die personalisierte Anzeigeneinblendung korrekt funktioniert.

Erweiterte Routing-Szenarien mit dynamischen Variablen und Konfigurationsaliasnamen finden Sie unter. [MediaTailor dynamische AnzeigenvARIABLEN für ADS-Anfragen](#) Hinweise zur Beibehaltung von Abfrageparametern beim CDN-Routing finden Sie unter. [MediaTailor Manifest-Abfrageparameter](#)

Important

Wenn CORS-Header nicht in das zwischengespeicherte Objekt oder in die CDN-Antwort an die Zuschauer aufgenommen werden, kann dies zu Wiedergabefehlern führen.

Konfiguration des CDN-Routing-Verhaltens

Richten Sie Ihr CDN so ein, dass verschiedene Arten von Anfragen entsprechend weitergeleitet werden.

Routing von Inhaltssegmenten

Das Routing von Inhaltssegmenten leitet Anfragen für Ihre tatsächlichen Inhaltssegmente an Ihren Ursprungsserver weiter. Wie das Routing von Anzeigensegmenten erfordert auch das Routing von Inhaltssegmenten eine korrekte CORS-Konfiguration, um eine reibungslose Wiedergabe in webbasierten Playern zu gewährleisten.

Eine ausführliche Anleitung zur Konfiguration finden Sie im CloudFront Beispiel unter [Priorität 4: Verhalten beim Herkunftspfad des Inhalts](#). Dieses Beispiel enthält spezifische Einstellungen, die Sie für andere verwenden CloudFront oder für andere anpassen sollten CDNs.

Zu den wichtigsten Konfigurationsanforderungen für das Routing von Inhaltssegmenten gehören:

- Verwenden Sie Pfadmuster, die Ihren Dateierweiterungen für Inhaltssegmente entsprechen (wie *.ts*.mp4, oder*.m4s)
- Anfragen an Ihren Inhaltsursprung weiterleiten (z. B. einen Amazon S3 S3-Bucket oder einen MediaPackage Endpunkt)
- Für ein optimales Verhältnis zwischen Cache-Treffern sollten Sie nur Abfragezeichenfolge-Parameter angeben, die dazu führen, dass Ihr Absender die Antwort im Cache-Schlüssel ändert und die Ursprungsanfrage weiterleitet
- Wenden Sie eine geeignete Cache-Richtlinie mit TTL-Werten von mehr als 24 Stunden an
- Fügen Sie Ihren Zuschauern CORS-Antwort-Header hinzu

Routing von Anzeigensegmenten

Das Routing von Anzeigensegmenten ist entscheidend für die Bereitstellung personalisierter Werbung für Zuschauer. Bei der Konfiguration des Routing von Anzeigensegmenten müssen Sie die korrekte Handhabung von CORS (Cross-Origin Resource Sharing) implementieren, um Probleme zu vermeiden, die zu Wiedergabefehlern in webbasierten Playern führen können.

Eine ausführliche Anleitung zur Konfiguration finden Sie im CloudFront Beispiel unter. [Priorität 0: Pfadverhalten von Anzeigensegmenten](#) Dieses Beispiel enthält spezifische Einstellungen, die Sie für andere verwenden CloudFront oder für andere anpassen sollten CDNs.

Zu den wichtigsten Konfigurationsanforderungen für das Routing von Anzeigensegmenten gehören:

- Verwenden Sie das Pfadmuster /tm/* speziell für MediaTailor Anzeigensegmente
- Anfragen weiterleiten an segments.mediatailor.*region*.amazonaws.com
- Um ein optimales Verhältnis zwischen Cache-Treffern zu erzielen, sollten Sie keine Header, Cookies oder Abfragezeichenfolge-Parameter für Viewer-Anforderungen in den Cache-Schlüssel oder in die ursprüngliche Anfrage aufnehmen
- Wenden Sie eine geeignete Cache-Richtlinie mit TTL-Werten von mehr als 24 Stunden an
- Fügen Sie Ihren Zuschauern CORS-Antwort-Header hinzu

Weiterleitung von Anfragen per Manifest

Verwenden Sie die folgenden allgemeinen Einstellungen, um multivariante Playlisten-, Medien-Playlisten- und MPD-Anfragen an MediaTailor weiterzuleiten. Informationen zur CloudFront Konfiguration finden Sie unter. [Konfiguration des Manifest-Cache-Verhaltens](#)

1. Erstellen Sie in Ihrer CDN-Konfigurationsoberfläche Verhalten für verschiedene Manifesttypen.
2. Stellen Sie Pfadmuster so ein, dass sie den Dateierweiterungen für multivariante Playlisten und Medien-Playlisten (* .m3u8 für HLS) und MPD-Dateierweiterungen (für DASH) entsprechen. * .mpd
3. Konfigurieren Sie die Ursprungseinstellung in Ihrem CDN so, dass sie auf Ihren Konfigurationsendpunkt verweist. MediaTailor
4. Deaktivieren Sie für das Einfügen von Anzeigen das Zwischenspeichern von personalisierten multivarianten Playlisten, Medien-Playlisten und. MPDs Da durch das Einfügen von Anzeigen personalisierte Manifeste bereitgestellt werden, sollte Ihr CDN diese nicht zwischenspeichern. Wenn ein anderes als das vorgesehene Wiedergabegerät eine zwischengespeicherte Playlist oder MPD empfängt, kann dies zu Problemen bei der Wiedergabe oder Nachverfolgung führen. Umfassende Anleitungen zum Zwischenspeichern, einschließlich TTL-Empfehlungen für alle Inhaltstypen, finden Sie unter. [Caching-Optimierung für CDN und Integrationen MediaTailor](#)
5. Konfigurieren Sie die Header-Weiterleitung für alle Header. Die Mindestanforderungen finden Sie unter [Erforderliche Header für MediaTailor die CDN-Integration](#).
6. Aktivieren Sie die Weiterleitung von Abfragezeichenfolgen, um Parameter für die Anzeigenausrichtung zu übergeben.

HLS multivariant playlist

Multivariante HLS-Playlist-Anfragen folgen diesen Formaten:

```
https://<playback-endpoint>/v1/index/<hashed-account-id>/<origin-id>/<index>.m3u8
```

Beispiel:

```
https://777788889999.mediataylor.us-east-1.amazonaws.com/v1/master/  
a1bc06b59e9a570b3b6b886a763d15814a86f0bb/Demo/assetId.m3u8
```

HLS media playlist

Anfragen für HLS-Medien-Playlisten folgen diesen Formaten:

```
https://<playback-endpoint>/v1/manifest/<hashed-account-id>/<session-id>/  
<manifestNumber>.m3u8
```

Spieleranfragen an `https://CDN_Hostname/some/path/asset.m3u8` werden auf der `https://mediataylor.us-west-2.amazonaws.com/v1/session/configuration/`

endpoint Grundlage des Schlüsselworts *.m3u8 in der Anfrage an den MediaTailor Pfad weitergeleitet.

Beispiel:

```
https://777788889999.mediatailor.us-east-1.amazonaws.com/v1/manifest/  
a1bc06b59e9a570b3b6b886a763d15814a86f0bb/c240ea66-9b07-4770-8ef9-7d16d916b407/0.m3u8
```

DASH MPD

DASH-MPD-Anfragen folgen diesen Formaten:

```
https://<playback-endpoint>/v1/dash/<hashed-account-id>/<origin-id>/<assetName>.mpd
```

Spieleranfragen an `https://CDN_Hostname/some/path/asset.mpd` werden auf der `https://mediatailor.us-west-2.amazonaws.com/v1/dash/configuration/endpoint` Grundlage des Schlüsselworts *.mpd in der Anfrage an den MediaTailor Pfad weitergeleitet.

Beispiel:

```
https://777788889999.mediatailor.us-east-1.amazonaws.com/v1/dash/  
a1bc06b59e9a570b3b6b886a763d15814a86f0bb/Demo/0.mpd
```

Bewährte Methoden für das CDN-Routing

Beachten Sie bei der Konfiguration des CDN-Routing-Verhaltens die folgenden bewährten Methoden, um eine optimale Leistung und Zuverlässigkeit zu gewährleisten:

Verwenden Sie spezifische Pfadmuster

Erstellen Sie spezifische Pfadmuster, die genau zu Ihrer Inhaltsstruktur passen, um eine korrekte Weiterleitung sicherzustellen.

Priorisieren Sie die Reihenfolge der Verhaltensweisen

In den meisten CDNs Fällen werden Verhaltensweisen der Reihe nach bewertet. Stellen Sie spezifischere Verhaltensweisen vor allgemeinere.

Testen Sie Verhaltensmuster

Stellen Sie sicher, dass Ihre Pfadmuster den erwarteten Anforderungen korrekt entsprechen, bevor Sie sie in der Produktion einsetzen.

Dokumentieren Sie Ihre Konfiguration

Dokumentieren Sie Ihr CDN-Routing-Verhalten, um die Fehlerbehebung und future Updates zu erleichtern.

Nächste Schritte

Nachdem Sie Ihr CDN-Routing-Verhalten konfiguriert haben, besteht der nächste Schritt darin, das CDN-Mapping in zu konfigurieren. MediaTailor Detaillierte Anweisungen finden Sie unter [Richten Sie das CDN-Mapping ein in MediaTailor](#).

Richten Sie das CDN-Mapping ein in MediaTailor

In diesem Abschnitt wird erklärt, wie Sie AWS Elemental MediaTailor die Verwendung Ihrer CDN-Domännennamen (Content Delivery Network) konfigurieren. Nachdem Sie Ihr CDN-Routing-Verhalten eingerichtet haben, müssen Sie Ihre MediaTailor Konfiguration aktualisieren, um sicherzustellen, dass Manifeste auf Ihre CDN-Domain verweisen, anstatt direkt auf Ursprungsserver zu verweisen.

Durch die Konfiguration des CDN-Mappings MediaTailor wird sichergestellt, dass alle Inhalte und Anzeigensegmente URLs in Ihren Manifesten auf Ihr CDN verweisen und nicht direkt auf die Ursprungsserver. Dieser Schritt ist wichtig, um eine vollständige CDN-Lieferkette zu erstellen und die Vorteile Ihrer CDN-Integration zu maximieren.

Konfiguration der CDN-Zuordnung in MediaTailor

Nachdem Sie Ihr CDN-Routing-Verhalten eingerichtet haben, konfigurieren Sie es MediaTailor für die Verwendung Ihrer CDN-Domainnamen:

1. Öffnen Sie die [MediaTailor-Konsole](#).
2. Wählen Sie die Konfiguration aus, die Sie aktualisieren möchten.
3. Geben Sie im Abschnitt Erweiterte Einstellungen der CDN-Konfiguration Ihren CDN-Domainnamen in das Feld CDN-Inhaltssegment-Präfix ein.
4. Wenn Sie eine separate CDN-Domain für Anzeigensegmente verwenden, geben Sie diese in das Feld CDN-Anzeigensegmentpräfix ein.
5. Speichern Sie Ihre Änderungen.

Diese Konfiguration stellt sicher, dass Manifeste mit diesem URLs Verweis auf Ihr CDN MediaTailor generiert werden und nicht direkt auf den Ursprungs- oder Anzeigensegmentspeicher.

Grundlegendes zum Verhalten der Basis-URL

MediaTailor bestimmt die Basis URLs in den Manifesten auf der Grundlage Ihrer CDN-Präfixkonfiguration:

- CDN-Anzeigensegmentpräfix konfiguriert: Anzeigensegmente verwenden das CDN-Präfix als Basis-URL.
- Das CDN-Anzeigensegmentpräfix ist nicht konfiguriert: Anzeigensegmente verwenden den direkten MediaTailor Standort als Basis-URL.
- CDN-Inhaltssegmentpräfix konfiguriert: Inhaltssegmente verwenden das CDN-Präfix als Basis-URL.
- Das CDN-Inhaltssegmentpräfix ist nicht konfiguriert: Inhaltssegmente verweisen auf den ursprünglichen Inhaltsursprung.

Behandlung von DASH BaseUrl

MediaTailor Verwaltet die BaseURL Einstellungen für DASH-Manifeste unterschiedlich für Inhalts- und Anzeigensegmente:

Inhaltssegmente:

- Mit CDN-Inhaltssegmentpräfix: MediaTailor Legt genau eines BaseURL auf der MPD Ebene fest, wobei das von Ihnen angegebene Präfix verwendet wird.
- Ohne CDN-Inhaltssegmentpräfix: MediaTailor Behält die vorhandenen BaseURL Einstellungen aus dem Ursprungsmanifest bei oder fügt eine auf der MPD Quell-URL basierende hinzu, falls keine vorhanden sind.

Anzeigensegmente:

- Mit CDN-Anzeigensegmentpräfix: Jeder Anzeigenzeitraum erhält BaseURL mit dem konfigurierten Präfix genau eins.
- Ohne CDN-Anzeigensegmentpräfix: In jedem Anzeigenzeitraum wird genau einer BaseURL auf den MediaTailor Anzeigeninhaltsserver verwiesen.

Example Beispiel für eine CDN-Zuordnung

Wenn Ihr Inhaltsursprung `http://origin.com/contentpath/` und Ihr CDN-Inhaltssegmentpräfix `https://cdn.example.com/`, dann `http://origin.com/contentpath/subdir/content.ts` erscheint ein Inhaltssegment, auf das normalerweise so verwiesen würde, im Manifest als `https://cdn.example.com/subdir/content.ts`

Wichtige Überlegungen

Beachten Sie bei der Konfiguration des CDN-Mappings in MediaTailor die folgenden wichtigen Überlegungen:

Verwenden Sie HTTPS für CDN-Präfixe

Verwenden Sie immer HTTPS URLs für Ihre CDN-Präfixe, um eine sichere Inhaltsbereitstellung zu gewährleisten.

Passen Sie die CDN-Verhaltenspfade an

Stellen Sie sicher, dass die CDN-Präfixe, die Sie konfigurieren, den Pfadmustern MediaTailor entsprechen, die Sie in Ihren CDN-Verhaltensweisen konfiguriert haben.

Denken Sie an regional CDNs

Wenn Sie unterschiedliche CDN-Domains für verschiedene Regionen verwenden, müssen Sie für jede Region separate MediaTailor Konfigurationen erstellen.

Überprüfen des Domäneneigentümers

Stellen Sie sicher, dass Sie die Kontrolle über die CDN-Domänen haben, in denen Sie konfigurieren. MediaTailor

Überprüfen Sie die CDN-Zuordnungskonfiguration

Stellen Sie nach der Konfiguration der CDN-Zuordnung in sicher MediaTailor, dass Ihre Konfiguration ordnungsgemäß funktioniert:

1. Fordern Sie über Ihr CDN ein Manifest an.
2. Untersuchen Sie den Inhalt des Manifests, um sicherzustellen, dass das Segment URLs auf Ihre CDN-Domain verweist.
3. Überprüfen Sie, ob das Inhaltssegment URLs im Manifest auf Ihre CDN-Domain verweist.

4. Überprüfen Sie, ob das Anzeigensegment URLs im Manifest auf Ihre CDN-Domain verweist.

Umfassende Test- und Validierungsverfahren finden Sie unter [Testen und Validieren für CDN und Integrationen MediaTailor](#).

Nächste Schritte

Nach der Konfiguration der CDN-Zuordnung besteht der nächste Schritt darin MediaTailor, bewährte Sicherheitsmethoden für Ihre CDN-Integration zu implementieren. Detaillierte Anweisungen finden Sie unter [Bewährte Sicherheitsmethoden für die CDN-Integration für MediaTailor](#).

Bewährte Sicherheitsmethoden für die CDN-Integration für MediaTailor

In diesem Abschnitt wird erklärt, wie Sie bewährte Sicherheitsmethoden für Ihre CDN-Integration (AWS Elemental MediaTailor Content Delivery Network) implementieren. Die richtige Sicherheitskonfiguration trägt dazu bei, Ihre Inhalte und Infrastruktur vor unbefugtem Zugriff und potenziellen Bedrohungen zu schützen.

Die Sicherung Ihrer CDN-Integration ist entscheidend, um Ihre Inhalte zu schützen, unbefugten Zugriff zu verhindern und die Einhaltung der Sicherheitsanforderungen sicherzustellen. Die Implementierung dieser bewährten Methoden trägt dazu bei, ein robustes Sicherheitsniveau für Ihren Streaming-Workflow zu schaffen.

CDN-Sicherheitskonfiguration

Bevor Sie Ihre CDN-Routing- und Cache-Einstellungen konfigurieren, sollten Sie diese bewährten Sicherheitsmethoden implementieren, um Ihre Inhalte und Infrastruktur zu schützen:

Sicherheit beim Transport

MediaTailor verwendet nur HTTPS für die gesamte Kommunikation und erlaubt keine HTTP-Verbindungen. Gehen Sie wie folgt vor, um eine sichere Kommunikation zwischen allen Komponenten in Ihrem Streaming-Workflow zu gewährleisten:

1. Konfigurieren Sie HTTPS für die gesamte Kommunikation zwischen Ihrem CDN und den MediaTailor Endpunkten. MediaTailor benötigt HTTPS und akzeptiert keine HTTP-Verbindungen.
2. Verwenden Sie TLS 1.2 oder höher für alle HTTPS-Verbindungen.
3. Konfigurieren Sie Ihr CDN so, dass nur HTTPS-Verbindungen von Zuschauern erzwungen werden.

Zugriffskontrolle

Implementieren Sie die folgenden Zugriffskontrollen, um Ihre Inhalte und Infrastruktur vor unbefugtem Zugriff zu schützen:

1. Konfigurieren Sie Einstellungen für geografische Beschränkungen, wenn Sie den Zugriff auf Inhalte auf bestimmte Regionen beschränken müssen.
2. Implementieren Sie signierte URLs oder signierte Cookies für Inhalte, für die eine Zuschauerauthentifizierung erforderlich ist.
3. Richten Sie IP-basierte Zulassungslisten für den administrativen Zugriff auf die CDN-Konfiguration ein.

Überwachung der Sicherheit

Implementieren Sie die folgenden Überwachungspraktiken, um Sicherheitsereignisse wirksam zu erkennen und darauf zu reagieren:

1. Aktivieren Sie die Zugriffsprotokollierung für Ihre CDN-Distribution.
2. Richten Sie Benachrichtigungen für ungewöhnliche Verkehrsmuster oder Zugriffsversuche ein.
3. Überprüfen Sie regelmäßig die Sicherheitskonfigurationen und aktualisieren Sie sie bei Bedarf.

Erweiterte Sicherheitsfunktionen

Die folgenden erweiterten Sicherheitsfunktionen bieten verbesserten Schutz für Ihre CDN-Integration:

Webanwendungsfirewall (WAF)

Wenn Ihr CDN WAF-Funktionen unterstützt, konfigurieren Sie es so, dass es vor häufigen Sicherheitslücken und Angriffen im Internet schützt.

DDoS-Schutz

Aktivieren Sie die von Ihrem CDN bereitgestellten DDoS-Schutzfunktionen, um verteilte denial-of-service Angriffe abzuwehren.

Zugriffskontrolle von Origin

Konfiguriere deine Ursprungsserver so, dass sie nur Anfragen von deinem CDN annehmen, um direkte Zugriffsversuche zu verhindern.

Verschlüsselung von Inhalten

Bei hochsensiblen Inhalten sollten Sie erwägen, zusätzliche Verschlüsselungsmechanismen zu implementieren, die über die Transportsicherheit hinausgehen.

Nächste Schritte

Nach der Implementierung bewährter Sicherheitsmethoden besteht der nächste Schritt darin, Ihre CDN-Integration zu testen und Fehler zu beheben. Umfassende Anweisungen [Beheben Sie Probleme mit der CDN-Integration](#) zum Testen und zur Fehlerbehebung finden Sie unter.

Richten Sie SSAI mit einem CDN für personalisierte Videowerbung ein

Dieser Abschnitt enthält umfassende Anleitungen zur Integration von AWS Elemental MediaTailor serverseitiger Anzeigeneinfügung (SSAI) in ein Content Delivery Network (CDN). Folgen Sie diesen Schritten, um Ihre SSAI-CDN-Integration einzurichten, zu konfigurieren und zu optimieren.

Serverseitige Anzeigeneinfügung (SSAI) ist eine Technologie, die personalisierte Werbung nahtlos in Videostreams auf Server- und nicht auf Client-Ebene einfügt. In Kombination mit einem CDN entsteht eine robuste, skalierbare Lösung für die Bereitstellung personalisierter Werbung für globale Zielgruppen mit minimaler Latenz.

In diesem Thema verwenden wir den Begriff Manifeste, um uns zusammenfassend auf multivariante Playlisten, Medien-Playlisten und zu beziehen. MPDs

Was du benötigst

Bevor Sie die MediaTailor Anzeigeneinblendung mit einem CDN einrichten, sollten Sie die folgenden erforderlichen Ressourcen zusammenstellen:

AWS Konto und Berechtigungen

Ein AWS Konto mit entsprechenden Berechtigungen zum Erstellen und Verwalten von MediaTailor Ressourcen

IAM-Berechtigungen für MediaTailor CloudFront (falls verwendet) und verwandte Dienste

Ausführliche Informationen zu den Genehmigungsanforderungen finden Sie unter [Sicherheit in AWS Elemental MediaTailor](#).

Erforderliche Dienste

AWS Elemental MediaTailor konfiguriert und läuft

Ein CDN-Konto (Content Delivery Network) (Amazon CloudFront oder CDN eines Drittanbieters)

Ursprungsserver für Ihre Inhalte (HLS oder DASH)

Ad Decision Server (ADS), der VAST oder VMAP unterstützt

Anforderungen an den Inhalt

Der Inhalt ist ordnungsgemäß codiert und im HLS- oder DASH-Format verpackt

Werbeunterbrechungsmarken in Ihren Inhalten (für VOD) oder SCTE-35-Markierungen (für Live)

Bevor Sie beginnen

Führen Sie die folgenden Einrichtungsaufgaben durch, bevor Sie die MediaTailor Anzeigeneinfügung mit einem CDN implementieren:

1. Konfigurieren Sie die Netzwerkkonnektivität zwischen Ihrem CDN und den MediaTailor Ursprungsservern
2. Richten Sie HTTPS für die sichere Bereitstellung von Inhalten ein
3. Konfigurieren Sie die DNS-Einstellungen für Ihre CDN-Domain
4. Führen Sie für die grundlegende MediaTailor Einrichtung die Schritte unter [Einrichtung](#) und [Erste Schritte mit der MediaTailor Anzeigenschaltung](#) aus.

Voraussetzungen für Kenntnisse

Um diese Lösung erfolgreich zu implementieren, sollten Sie über Folgendes verfügen:

- Verständnis von Streaming-Protokollen (HLS/DASH)
- Grundkenntnisse der CDN-Konfigurationsprinzipien
- Vertrautheit mit Konzepten zur Anzeigeneinfügung

Vorteile der CDN-Integration

Die Integration von SSAI in ein CDN bietet die folgenden Hauptvorteile:

Verbessertes Zuschauererlebnis

Durch eine korrekte Konfiguration werden Pufferung, Startzeiten und Wiedergabefehler bei Anzeigenübergängen CDNs reduziert. Dies führt zu einem höheren Engagement und einer höheren Zufriedenheit der Zuschauer.

Senkung der Kosten

Effiziente Caching-Strategien minimieren die Anzahl der ursprünglichen Anfragen. Dies reduziert die Datenübertragungskosten und die Auslastung des Ursprungsservers, was besonders wichtig für werbefinanzierte Inhalte mit hohem Volumen ist.

Skalierbarkeit

Optimierte CDN-Konfigurationen bewältigen Besucherspitzen bei beliebten Veranstaltungen ohne Leistungseinbußen und stellen so sicher, dass Ihre personalisierten Anzeigen auch zu Spitzenzeiten geschaltet werden.

Weltweite Reichweite

Bei richtiger Konfiguration CDNs können Sie Zuschauern weltweit unabhängig von ihrem Standort Inhalte mit geringer Latenz bereitstellen und so Ihr potenzielles Publikum erweitern.

Reibungslose Anzeigenübergänge

Die optimierte CDN-Konfiguration sorgt für reibungslose Übergänge zwischen Inhalten und Werbung und sorgt so für ein Seherlebnis in Sendequalität.

Die folgenden Themen bieten umfassende Anleitungen zur Konfiguration MediaTailor mit einem CDN für optimale Leistung.

Themen

- [Machen Sie sich mit der Architektur der Anzeigeneinfügung für CDN und Integrationen vertraut MediaTailor](#)
- [Richten Sie grundlegendes MediaTailor SSAI mit einem CDN für eine optimale Anzeigenbereitstellung ein](#)
- [Integrieren Sie MediaTailor SSAI in die Kanalbaugruppe für monetarisierte lineare Kanäle](#)
- [Optimieren Sie die Leistung für CDN und Integrationen MediaTailor](#)
- [Überwachen Sie den Betrieb für CDN und MediaTailor Integrationen](#)
- [Problembehebung bei MediaTailor SSAI mit CDNs für eine unterbrechungsfreie Anzeigenbereitstellung](#)

Machen Sie sich mit der Architektur der Anzeigeneinfügung für CDN und Integrationen vertraut MediaTailor

In diesem Abschnitt werden die Konzepte und die Architektur der serverseitigen Anzeigeneinfügung (SSAI) mit Content Delivery Networks (CDNs) für erläutert. AWS Elemental MediaTailor Sie erfahren, wie dynamische Anzeigeneinfügung und Manipulation von Manifesten zusammenarbeiten, um eine effektive Monetarisierung von Videos zu ermöglichen.

Serverseitige Anzeigeneinfügung (SSAI) ermöglicht Ihnen MediaTailor :

- Fügen Sie personalisierte Werbung an definierten Anzeigenunterbrechungen in Ihre Videostreams ein
- Richten Sie Anzeigen präzise auf der Grundlage von Zuschauerdaten aus
- Eliminieren Sie die Notwendigkeit einer Technologie zur kundenseitigen Anzeigenschaltung

In Kombination mit einem CDN können Sie den Zuschauern diese personalisierten Streams mit verbesserter Leistung und Skalierbarkeit bereitstellen und so Ihre Strategie zur Monetarisierung von Videos verbessern.

Die empfohlene Architektur für die Anzeigeneinblendung mit einem CDN positioniert das CDN zwischen Zuschauern und der Anzeigeneinblendung, sodass bei der Anzeigeneinblendung direkt auf Inhalte von Ihrem Ursprung zugegriffen wird. Diese Architektur bietet die folgenden Vorteile sowohl für die Bereitstellung von Inhalten als auch für die Monetarisierung von Videos:

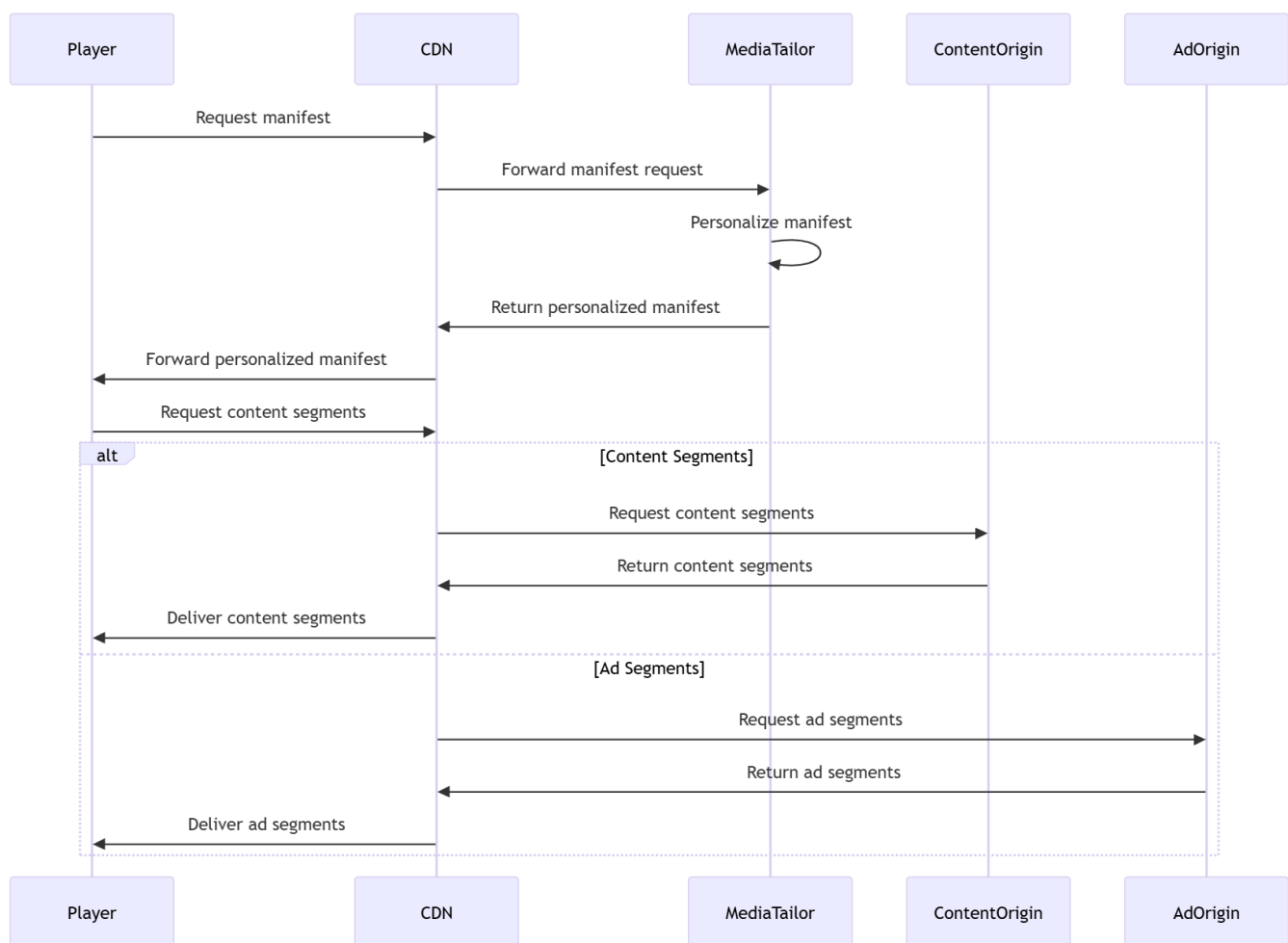
- Effektives Caching von Inhalten und Anzeigensegmenten
- Reduzierte Anforderungslast bei MediaTailor
- Verbesserte Übertragungsgeschwindigkeit für Zuschauer
- Vereinfachtes URL-Management
- Konsistente Bereitstellung personalisierter Werbung auf allen Geräten

In dieser empfohlenen Architektur:

1. Zuschauer fordern Manifeste vom CDN an
2. CDN leitet Anfragen zur Anzeigeneinfügung weiter
3. Anfragen zum Einfügen von Anzeigen enthalten Inhalte, die vom Ursprung her sichtbar sind
4. Beim Einfügen von Anzeigen werden Anzeigen vom Ad Decision Server (ADS) angefordert

5. Bei der Anzeigeneinfügung werden Manifeste personalisiert, indem die Anzeigenmarkierungen (aus dem ursprünglichen Manifest) durch URLs diesen Verweis auf gezielte Anzeigensegmente für den jeweiligen Zuschauer (aus dem ADS) ersetzt werden
6. Durch das Einfügen von Anzeigen werden die personalisierten Manifeste, die das Anzeigensegment enthalten, URLs an das CDN zurückgegeben, das sie an die Zuschauer weiterleitet
7. Zuschauer fordern Segmente über das CDN an
8. CDN leitet Segmentanfragen auf der Grundlage des Segmenttyps weiter:
 - Anfragen nach Inhaltssegmenten werden an den Inhaltsursprung weitergeleitet
 - Anfragen nach Anzeigensegmenten gehen an MediaTailor

Diese Architektur gewährleistet optimale Leistung bei gleichzeitiger Beibehaltung der Sicherheits- und Flexibilitätsvorteile der Verwendung eines CDN.



 Note

Dieser Ablauf variiert geringfügig zwischen VOD- und Live-Inhalten. Bei VOD können Manifeste länger zwischengespeichert werden, wohingegen Live-Inhalte häufigere Manifest-Aktualisierungen erfordern, um die Streamkontinuität aufrechtzuerhalten.

Der Hauptunterschied zwischen VOD und dem Zwischenspeichern von Live-Inhalten:

VOD-Inhalt

Legen Sie längere TTL-Werte (Minuten bis Stunden) für Manifeste fest, da sie sich nicht häufig ändern

Live-Inhalte

Lege kürzere TTL-Werte (Sekunden) für Manifeste fest, um sicherzustellen, dass Zuschauer die aktuellsten Stream-Segmente erhalten

Wir empfehlen nicht, dass Sie ein CDN zwischen Ihrem Inhaltsursprung und platzieren. AWS Elemental MediaTailor Dies kann zu mehreren technischen Herausforderungen führen:

Kollisionen zwischen Cache-Schlüssel

Konfigurieren Sie Ihr CDN so, dass Abfrageparameter ordnungsgemäß verarbeitet werden. Dadurch wird MediaTailor verhindert, dass falsche Manifeste empfangen werden, wenn dasselbe Manifest mit unterschiedlichen Abfrageparametern angefordert wird.

Probleme mit der Gzip-Komprimierung

Wenn bei der Analyse von Manifesten Fehler auftreten, stellen Sie sicher, dass Ihr CDN korrekt formatierte Manifeste an übermittelt. MediaTailor Einige liefern CDNs möglicherweise beschädigte GZip-Payloads, die zu Parsing-Fehlern führen können. In diesem Fall müssen Sie möglicherweise die Komprimierung zwischen Ihrem CDN deaktivieren und MediaTailor gleichzeitig die Komprimierung beibehalten, um an anderen Stellen Ihres Workflows Kosten zu sparen.

Offenkundige Frische

Für Live-Streams konfigurieren Sie Ihr CDN so, dass aktuelle Manifeste an gesendet werden MediaTailor. Dadurch werden Synchronisierungsprobleme zwischen Inhalten und Anzeigen vermieden.

Leistungsoptimierung

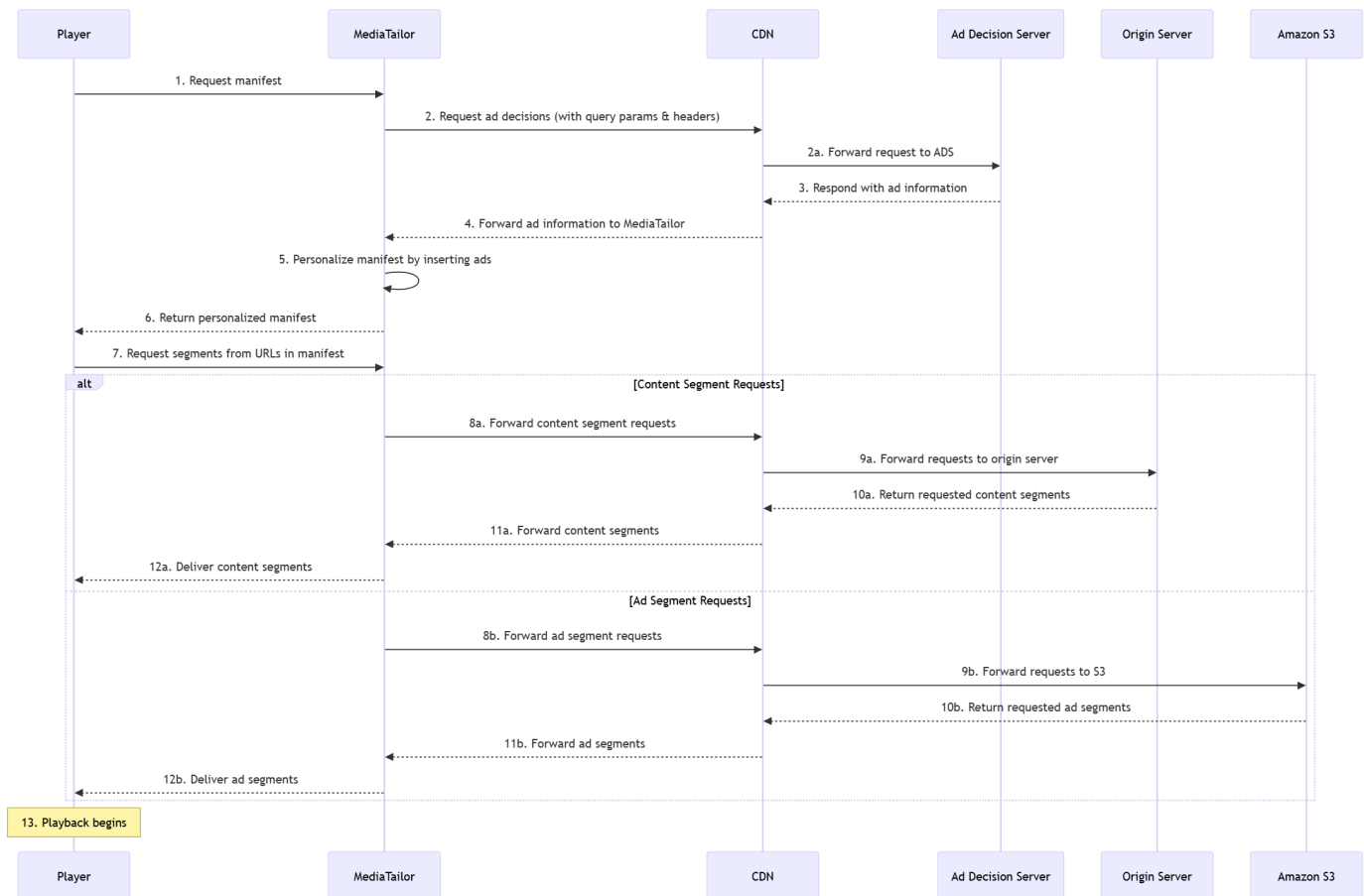
Minimiere Netzwerk-Hops und mögliche Cache-Fehlschläge, um die Startzeiten der Wiedergabe zu reduzieren.

Cache-Verwaltung

Implementieren Sie vereinfachte Strategien zur Cache-Invalidierung, insbesondere für Live-Inhalte, bei denen Manifeste häufig aktualisiert werden.

In dieser suboptimalen Architektur:

1. Zuschauer fordern multivariante Playlisten, Medien-Playlisten oder MPDs direkt von. AWS Elemental MediaTailor
2. MediaTailor fordert Inhaltsverzeichnisse (multivariante Playlisten, Medien-Playlisten oder MPDs) über das CDN an.
3. Das CDN leitet Anfragen an den Ursprungsserver weiter.
4. Der Ursprungsserver gibt multivariante Playlisten, Medien-Playlisten oder MPDs an das CDN zurück.
5. Das CDN leitet multivariante Playlisten, Medien-Playlisten oder an weiter. MPDs MediaTailor
6. MediaTailor fordert Anzeigen vom Ad Decision Server (ADS) an.
7. MediaTailor personalisiert Manifeste, indem Werbeanzeigen in multivariante Playlisten oder Medien-Playlisten eingefügt werden, oder MPDs sie werden direkt an die Zuschauer ausgeliefert.
8. Diese Architektur führt zu zusätzlicher Latenz, potenziellen Caching-Problemen und erschwert die Fehlerbehebung.



Ablauf von Anfragen und Antworten

Wenn Sie die dynamische Anzeigeneinfügung mit einem CDN implementieren, konfigurieren Sie Ihr System so, dass es diesen Anfrage- und Antwortfluss unterstützt:

1. Konfigurieren Sie Ihren Player so, dass er multivariante Playlisten (HLS) oder MPDs (DASH) von Ihrem CDN mit MediaTailor manifestem Ursprung anfordert.
2. Richten Sie Ihr CDN so ein, dass alle multivarianten Playlisten-, Medien-Playlisten- und MPD-Anfragen, einschließlich aller Abfrageparameter und Header, weitergeleitet werden MediaTailor.
3. Stellen Sie sicher, dass Sie mit Ihrem Ad Decision Server (ADS) kommunizieren und Abfrageparameter und Header weitergeben können.
4. Konfigurieren Sie Ihr ADS so, dass es anhand von Abfrageparametern bestimmt, welche Anzeigen eingefügt werden sollen.

5. Richten Sie das CDN-Präfix in der MediaTailor Wiedergabekonfiguration ein, MediaTailor sodass Sie die URL-Präfixe für Inhalte und Anzeigensegmente durch CDN-Domännennamen ersetzen können.
6. Konfigurieren Sie Ihr CDN so, dass personalisierte multivariante Playlisten, Medien-Playlisten und MPDs von an den anfragenden Player weitergeleitet werden. MediaTailor
7. Richten Sie Ihr CDN für die Übersetzung von Segmenten ein URLs, indem Sie Anfragen zu Inhaltssegmenten an den Ursprungsserver und Anzeigenanfragen an den Amazon S3 S3-Bucket weiterleiten, in dem transkodierte Anzeigen MediaTailor gespeichert werden.

CDN-Terminologie für das Einfügen von Anzeigen

Wenn Sie diese Schlüsselbegriffe verstehen, können Sie Ihre CDN-Integration für Anzeigeneinfügung implementieren und Fehler beheben:

Origin-CDN und Edge-CDN

Origin-CDN: Ein CDN, das zwischen MediaTailor und Ihrem Inhaltsursprung positioniert ist. Es speichert Inhaltssegmente im Cache, um die Belastung Ihrer Ursprungsserver zu reduzieren. In einer Multi-CDN-Architektur ist dies die erste CDN-Schicht, die direkt mit dem Ursprung verbunden ist.

Edge-CDN: Ein CDN, das zwischen Zuschauern und positioniert ist. MediaTailor Es bietet Zuschauern personalisierte Manifeste und Inhalte. In einer Multi-CDN-Architektur ist dies die äußerste CDN-Schicht, die direkt mit den Zuschauern verbunden ist.

Bedingungen für die CDN-Konfiguration

Cache-Verhalten: Regeln, die festlegen, wie ein CDN verschiedene Arten von Anfragen behandelt. Zu diesen Regeln gehören:

- Einstellungen für die Dauer des Zwischenspeichers
- Origin-Routing-Konfigurationen
- Parameter für die Bearbeitung von Anfragen

TTL (Time To Live): Die Dauer, für die Inhalte in einem CDN-Cache gültig bleiben, bevor sie vom Ursprung aktualisiert werden müssen.

Cache-Schlüssel: Die eindeutige Kennung, die von einem CDN zum Speichern und Abrufen von zwischengespeicherten Inhalten verwendet wird. Es beinhaltet in der Regel:

- URL-Pfad
- Abfrageparameter
- Ausgewählte Header

Origin Shield: Eine zwischengeschaltete Caching-Ebene zwischen CDN-Edge-Standorten und einem Ursprungsserver. Es reduziert die Anzahl der Anfragen an Ihren Ursprung.

Reduzieren von Anfragen: Eine CDN-Funktion, die mehrere gleichzeitige Anfragen für denselben Inhalt zu einer einzigen ursprünglichen Anfrage zusammenfasst.

MediaTailor-spezifische CDN-Begriffe

CDN-Inhaltssegmentpräfix: Der CDN-Domänenname, der bei der Generierung URLs von Inhaltssegmenten in Manifesten AWS Elemental MediaTailor verwendet wird.

CDN-Anzeigensegmentpräfix: Der CDN-Domainname, der bei der Generierung URLs von Anzeigensegmenten in Manifesten MediaTailor verwendet wird.

Weitere Informationen zur CDN-Konfiguration mit MediaTailor finden Sie unter [Richten Sie die CDN-Integration ein](#)

Note

Diese Begriffe stimmen mit denen überein, die in der Dokumentation zur Kanalmontage verwendet werden. Informationen zur Terminologie der Kanalmontage finden Sie unter [CDN-Terminologie für die Kanalmontage](#).

Richten Sie grundlegendes MediaTailor SSAI mit einem CDN für eine optimale Anzeigenbereitstellung ein

In diesem Abschnitt finden Sie step-by-step Anweisungen zur Konfiguration der AWS Elemental MediaTailor dynamischen Anzeigeneinfügung mit einem Content Delivery Network (CDN), um Ihren Workflow zur Monetarisierung von Videos zu optimieren.

Informationen zur erweiterten Konfiguration von Anzeigenservern mithilfe dynamischer Variablen finden Sie unter [MediaTailor dynamische Anzeigenvariablen für ADS-Anfragen](#). Informationen zur Weitergabe von Parametern CDNs für die Anzeigenausrichtung finden Sie unter [MediaTailor Manifest-Abfrageparameter](#).

Konzeptionelle Informationen zu SSAI with finden Sie CDNs unter [Machen Sie sich mit der Architektur der Anzeigeneinfügung für CDN und Integrationen vertraut MediaTailor](#).

Voraussetzungen

Bevor Sie die Anzeigeneinfügung mit einem CDN einrichten, stellen Sie sicher, dass Sie über Folgendes verfügen:

- Eine aktive Konfiguration AWS Elemental MediaTailor
- Ein Content-Ursprungsserver, der HLS- oder DASH-Inhalte mit entsprechenden Werbemarkierungen bereitstellt

Informationen zu Werbemarkierungen finden Sie unter [Das Verhalten bei der Insertion von Anzeigen verstehen](#).

- Ein Ad Decision Server (ADS), der VAST oder VMAP für die Anzeigenausrichtung unterstützt
- Ein CDN-Konto (wie Amazon CloudFront oder ein anderer CDN-Anbieter)
- Grundkenntnisse über Konzepte zur Manipulation von Manifesten und zur dynamischen Anzeigeneinfügung

Schritt 1: Konfigurieren Sie das CDN-Caching für eine optimale Anzeigenbereitstellung

Die richtige CDN-Caching-Konfiguration ist entscheidend für eine optimale Leistung Ihres Workflows zur Monetarisierung von Videos. Die Caching-Anforderungen unterscheiden sich zwischen serverseitiger Anzeigeneinfügung (SSAI) und serverseitiger Anzeigeneinfügung (SGAI). Verwenden Sie diese empfohlenen Einstellungen, um eine effiziente Bereitstellung von Inhalten und personalisierter Werbung sicherzustellen:

SSAI CDN-Caching-Einstellungen

Für serverseitige Workflows zum Einfügen von Anzeigen ist die richtige Caching-Konfiguration entscheidend für eine optimale Leistung. SSAI erfordert spezielle TTL-Werte und Cache-Schlüsseleinstellungen, um sicherzustellen, dass personalisierte Manifeste nicht zwischengespeichert werden, während Segmente effizient zwischengespeichert werden.

Detaillierte SSAI-Caching-Einstellungen, einschließlich TTL-Werte, Pfadmuster und Cache-Schlüsselkonfigurationen, finden Sie im CDN-Optimierungsleitfaden. [Serverseitiges Zwischenspeichern von Werbeanzeigen \(SSAI\)](#)

Die wichtigsten Caching-Prinzipien für SSAI:

- **Manifeste:** Stellen Sie TTL auf 0 Sekunden ein, um das Zwischenspeichern personalisierter Inhalte zu verhindern
- **Segmente:** Aggressives Zwischenspeichern (mehr als 24 Stunden), um die Belastung durch den Ursprung zu reduzieren
- **Cache-Schlüssel:** Schließt alle Abfrageparameter für Manifeste ein, nur den URL-Pfad für Segmente

SGAI CDN-Caching-Einstellungen

Bei servergesteuerten Workflows zur Anzeigeneinfügung unterscheiden sich die Caching-Anforderungen von denen von SSAI, da SGA-Manifeste für kurze Zeiträume zwischengespeichert werden können und gleichzeitig personalisierte Werbeerlebnisse bieten.

Umfassende SGA-Caching-Einstellungen, einschließlich VOD- und Live-TTL-Werten, finden Sie in den Caching-Tabellen des Optimierungsleitfadens. SGA ermöglicht eine bessere Cache-Effizienz als SSAI und behält gleichzeitig die Funktionen zur Personalisierung von Anzeigen bei.

Die wichtigsten Unterschiede beim SGA-Caching:

- **Manifeste:** Können für kurze Zeiträume zwischengespeichert werden (5-30 Minuten für VOD, 2-10 Sekunden für Live)
- **Segmente:** Aggressiver Cache wie SSAI (mehr als 24 Stunden für die meisten Inhalte)
- **Leistungsvorteil:** Bessere Cache-Trefferquoten als SSAI aufgrund zwischenspeicherbarer Manifeste

Für Amazon CloudFront können Sie diese Einstellungen mithilfe von Cache-Verhalten mit unterschiedlichen TTL-Werten und Cache-Schlüsselrichtlinien implementieren. Weitere CDNs Informationen zur Implementierung ähnlicher Caching-Regeln finden Sie in der jeweiligen Dokumentation.

Schritt 2: Implementieren Sie hybride Ansätze (falls erforderlich)

Wenn Ihre Architektur einen hybriden Ansatz mit einer separaten CDN- oder Caching-Ebene zwischen dem Inhaltsursprung und folgenden Komponenten erfordert: MediaTailor

1. Implementieren Sie in Ihrer CDN-Konfiguration eine klare Trennung der einzelnen Bereiche.
2. Konfigurieren Sie spezifische CDN-Einstellungen, um die im vorherigen Abschnitt beschriebenen technischen Probleme zu vermeiden.

3. Testen Sie Ihre Konfiguration gründlich, um sicherzustellen, dass die Manifest-Personalisierung ordnungsgemäß funktioniert.
4. Überwachen Sie die Leistungskennzahlen, um eine optimale Bereitstellung von multivarianten Playlisten, Medien-Playlisten und Segmenten MPDs sicherzustellen.

Bei der Implementierung eines hybriden Ansatzes sollten Sie die folgenden spezifischen Konfigurationen berücksichtigen:

- Für das CDN zwischen dem Ursprung des Inhalts und MediaTailor:
 - Konfigurieren Sie den Komprimierungs-Passthrough für Manifestdateien, um den ursprünglichen Komprimierungsstatus Ihres Ursprungs beizubehalten
 - Schließt alle Abfrageparameter in den Cache-Schlüssel ein
 - Legen Sie kurze TTL-Werte für Live-Inhaltsmanifeste fest
- Für das CDN zwischen Zuschauern MediaTailor und Zuschauern:
 - Konfigurieren Sie längere Cache-Zeiten für Anzeigensegmente
 - TTLs Für personalisierte Manifeste geeignet einstellen
 - Implementieren Sie das richtige Origin-Routing für Inhalte und Anzeigensegmente

Schritt 3: Vervollständigen Sie Ihr CDN-Setup

Nachdem Sie Ihre Architektur ausgewählt und den Anforderungsablauf verstanden haben, schließen Sie Ihre Einrichtung ab, indem Sie die detaillierten Konfigurationsschritte unter befolgen. [Richten Sie die CDN-Integration ein](#)

Informationen zu bestimmten CDN-Anbietern finden Sie in diesen zusätzlichen Ressourcen:

- Amazon CloudFront: CloudFront Spezifische Konfigurationsschritte finden Sie unter [CloudFront-Integration](#)
- Sonstiges CDNs: Wenden Sie die in diesem Handbuch beschriebenen allgemeinen Prinzipien an und passen Sie sie an die spezifischen Konfigurationsoptionen Ihres CDN an

Schritt 4: Überprüfen Sie Ihre Konfiguration

Stellen Sie nach Abschluss Ihrer CDN-Einrichtung sicher, dass Ihr Workflow zur dynamischen Anzeigeneinfügung ordnungsgemäß funktioniert:

1. Testen Sie die Wiedergabe über Ihr CDN mit einem Beispielsplayer
2. Stellen Sie sicher, dass personalisierte Werbung an den dafür vorgesehenen Anzeigenunterbrechungen korrekt eingefügt wurde
3. Überprüfen Sie die CDN-Protokolle, um die korrekte Weiterleitung von Anfragen zu bestätigen
4. Überwachen Sie die Cache-Trefferraten, um eine optimale Leistung sowohl für Inhalte als auch für Anzeigensegmente sicherzustellen
5. Stellen Sie sicher, dass die Parameter für die Anzeigenausrichtung ordnungsgemäß durch den Workflow geleitet werden

Umfassende Test- und Validierungsverfahren finden Sie unter [Testen und Validieren für CDN und Integrationen MediaTailor](#). Ausführliche Informationen zur Überwachung Ihrer SSAI-Implementierung finden Sie unter [Überwachen Sie den Betrieb für CDN und MediaTailor Integrationen](#). Informationen zur Leistungsoptimierung finden Sie unter [Optimieren Sie die Leistung für CDN und Integrationen MediaTailor](#).

Integrieren Sie MediaTailor SSAI in die Kanalbaugruppe für monetarisierte lineare Kanäle

In diesem Thema wird erklärt, wie die AWS Elemental MediaTailor serverseitige Anzeigeneinfügung mit der Kanalzusammenstellung und der CDN-Integration (Content Delivery Network) kombiniert werden kann. Diese Integration ermöglicht Ihnen:

- Erstellen Sie monetarisierte lineare Kanäle mit personalisierter Werbung
- Stellen Sie zielgerichtete Werbung für verschiedene Zuschauer bereit, die sich dieselben Inhalte ansehen
- Sorgen Sie für ein Fernseherlebnis in Sendequalität

Vorteile der Kombination von SSAI mit Kanalmontage

Die Integration von SSAI in die Kanalmontage bietet mehrere wichtige Vorteile:

Monetarisierung linearer Kanäle

Fügen Sie personalisierte Anzeigen in Ihre linearen Kanäle ein, um Einnahmen aus Ihrer Inhaltsbibliothek zu generieren. Sie können sowohl Live- als auch VOD-Inhalte in einem einzigen linearen Stream monetarisieren.

Personalisierte Werbung

Stellen Sie unterschiedlichen Zuschauern, die sich denselben Kanalinhalt ansehen, unterschiedliche Werbeanzeigen bereit. Dieser gezielte Ansatz erhöht die Relevanz von Anzeigen und erhöht den potenziellen Umsatz im Vergleich zu herkömmlicher Rundfunkwerbung.

Vereinfachtes Management von Werbeunterbrechungen

Definieren Sie Werbeunterbrechungspunkte in Ihren Programmen zur Kanalmontage, ohne den Inhalt mit SCTE-35-Markern konditionieren zu müssen. Dies macht es einfacher, Anzeigen an natürlichen Grenzpunkten in Ihren Inhalten einzufügen.

Erlebnis in Sendequalität

MediaTailor sorgt für ein qualitativ hochwertiges Seherlebnis mit nahtlosen Übergängen zwischen Inhalten und Werbung. Durch das Einfügen von serverseitiger Werbung werden viele häufig auftretende Probleme vermieden, wie z. B.:

- Pufferung bei Anzeigenübergängen
- Werbeblocker, die die Monetarisierung verhindern
- Inkonsistente Wiedergabequalität

Skalierbare Bereitstellung

In Kombination mit einem CDN kann diese Integration auf Millionen von gleichzeitigen Zuschauern skaliert werden, ohne dass die Leistung oder die Personalisierungsmöglichkeiten beeinträchtigt werden.

Übersicht über die Architektur

Die Architektur für die Kombination von SSAI mit Channel Assembly umfasst in der Regel die folgenden Komponenten:

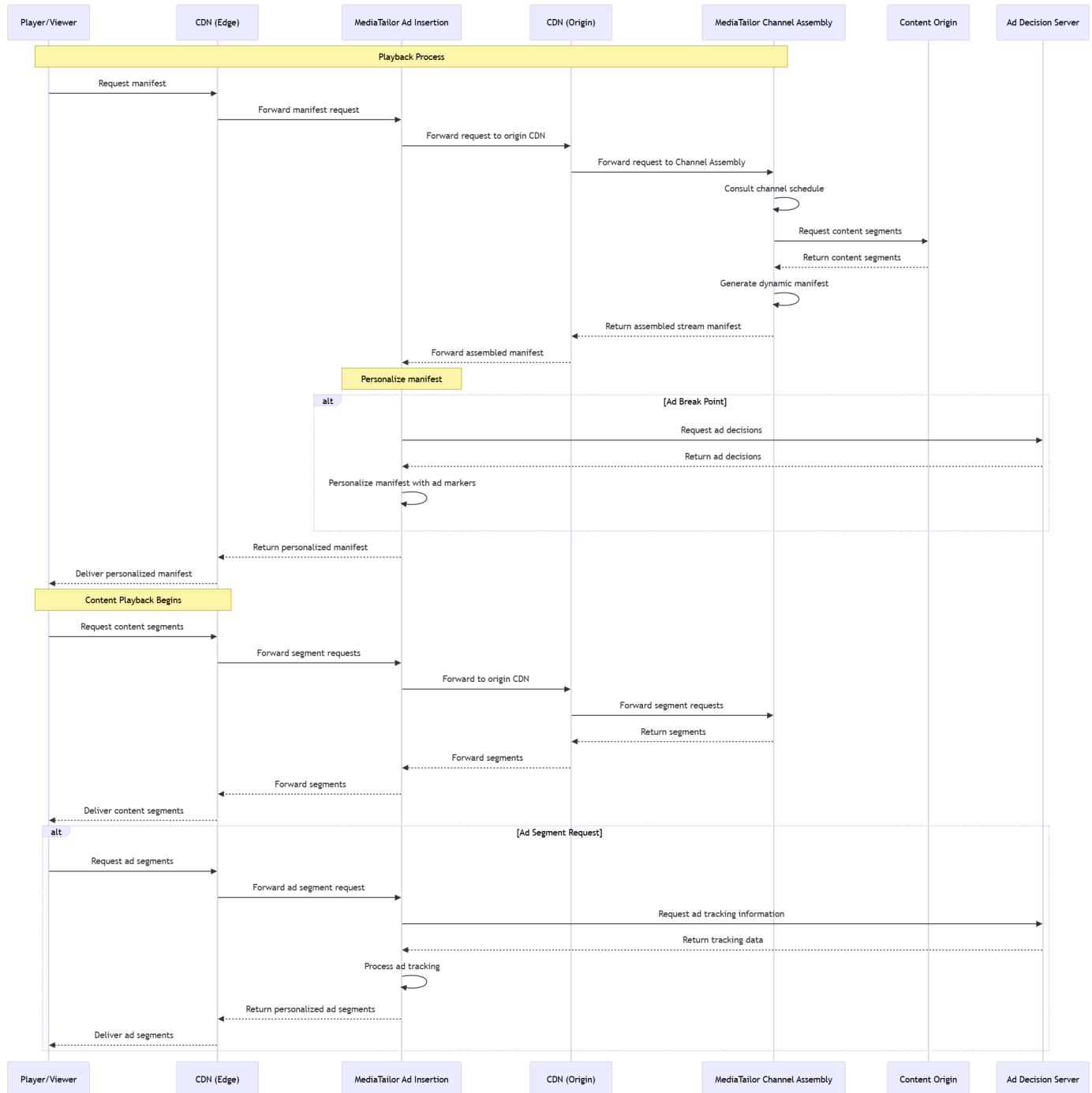
- Kanalzusammenstellung: Erstellt lineare Kanäle aus VOD- und Live-Inhalten und fügt in das generierte Manifest Slate-Inhalte ein, die zu Werbemarkierungen führen
- Anzeigeneinfügung: Erkennt Unterbrechungspunkte und Einfügungen von Anzeigen, die auf personalisierte Anzeigensegmente im Manifest URLs verweisen
- Ad Decision Server (ADS): Legt fest, welche Anzeigen für jeden Zuschauer eingefügt werden
- Content Delivery Network (CDN): Stellt den Zuschauern die zusammengestellten Inhalte und Anzeigensegmente bereit

- Ursprungsserver: Speichert die VOD- und Live-Inhaltssegmente

In dieser Architektur:

1. Channel Assembly erstellt einen linearen Kanal aus Ihren VOD- und Live-Inhalten und fügt Slate-Inhalte ein, die die Werbemarkierungen erstellen, in das generierte Manifest
2. Wenn ein Zuschauer den Kanal anfordert, erkennt die Anzeigeneinfügung die Werbeunterbrechungen, die in den linearen Kanal eingefügt wurden
3. Bei der Anzeigeneinfügung wird das ADS aufgerufen, um die Liste der Anzeigen zu erhalten, sie transkodiert sie und fügt, die auf die transkodierten Anzeigensegmente URLs verweisen, in das personalisierte Manifest ein
4. Das CDN stellt dem Zuschauer den personalisierten Stream zur Verfügung

Das folgende Diagramm veranschaulicht diesen Arbeitsablauf:



Einrichtung der Integration

So richten Sie SSAI mit Kanalmontage ein:

1. Konfigurieren Sie Ihr Edge-CDN so, dass es Manifestanfragen von Zuschauern akzeptiert und sie zur AWS Elemental MediaTailor Anzeigeneinfügung weiterleitet.
2. Richten Sie die MediaTailor Anzeigeneinfügung so ein, dass Anfragen an Ihr Ursprungs-CDN weitergeleitet werden.
3. Konfigurieren Sie Ihr Ursprungs-CDN so, dass Anfragen an die MediaTailor Channel-Assembly weitergeleitet werden.
4. Richten Sie MediaTailor Channel Assembly ein, um dynamische Manifeste auf der Grundlage des aktuellen Zeitplans zu generieren.
5. Konfigurieren Sie Ihr Ursprungs-CDN so, dass die zusammengestellten Manifeste an die MediaTailor Anzeigeneinfügung weitergeleitet werden.
6. Richten Sie die MediaTailor Anzeigeneinfügung so ein, dass Sie an Anzeigenunterbrechungen Anzeigenentscheidungen von Ihrem Anzeigenentscheidungsserver anfordern.
7. Konfigurieren Sie die MediaTailor Anzeigeneinfügung so, dass Manifeste personalisiert werden, indem Sie die Anzeigenmarkierungen (aus der Kanalzusammenstellung) URLs durch Verweise auf gezielte Anzeigensegmente (aus dem ADS) ersetzen.
8. Richten Sie Ihr Edge-CDN ein, um Zuschauern personalisierte Manifeste bereitzustellen.
9. Konfigurieren Sie Ihre CDN-Architektur so, dass sowohl Anfragen nach Inhalten als auch nach Anzeigensegmenten effizient bearbeitet werden.

Definition von Werbeunterbrechungen bei der Kanalzusammenstellung

Beim Erstellen von Programmen in Channel Assembly können Sie Werbeunterbrechungen auf verschiedene Arten definieren:

Programmübergänge

Fügen Sie Werbeanzeigen zwischen Programmen in Ihren Kanalplan ein. Dies ist der einfachste Ansatz und stellt sicher, dass Werbung den Programminhalt nicht unterbricht.

SCTE-35-Markierungen

Wenn Ihr VOD-Inhalt SCTE-35-Markierungen enthält, können diese Markierungen bei der Kanalzusammenstellung beibehalten werden, und beim Einfügen von Anzeigen können sie als Werbeunterbrechungspunkte verwendet werden.

Zeitbasierte Einfügung

Definieren Sie Werbeunterbrechungen zu bestimmten Zeitpunkten innerhalb von Programmen. Auf diese Weise können Sie Anzeigen an natürlichen Unterbrechungspunkten in Ihren Inhalten einfügen.

Ausführliche Informationen zum Erstellen von Programmen mit Werbeunterbrechungen finden Sie unter [Mit Programmen arbeiten](#).

Überlegungen zum CDN-Caching

Für eine optimale Leistung bei der Kombination von Channel-Assembly und SSAI mit einem CDN:

- Konfigurieren Sie das Cache-Verhalten, das zwischen Channel-Assembly- und SSAI-Anfragen unterscheidet
- Legen Sie die entsprechenden TTL-Werte für Manifeste und Segmente fest, wie unter empfohlen [Schritt 1: Konfigurieren Sie das CDN-Caching für eine optimale Anzeigenbereitstellung](#)
- Stellen Sie sicher, dass das Routing zwischen der Kanalmontage, der Anzeigeneinfügung und Ihren CDN-Ursprüngen ordnungsgemäß verläuft
- Überwachen Sie die Leistungskennzahlen sowohl für die Komponenten für die Kanalmontage als auch für die Anzeigeneinfügung

Empfohlene Caching-Einstellungen für die kombinierte Implementierung

Inhaltstyp	TTL	Schlüsselemente zwischenspeichern
Manifeste für die Kanalmontage	0 Sekunden	URL-Pfad + Abfrageparameter
Personalisierte SSAI-Manifeste	0 Sekunden	URL-Pfad + alle Abfrageparameter
Inhaltssegmente	Mehr als 24 Stunden	Nur URL-Pfad
Anzeigensegmente	Mehr als 24 Stunden	Nur URL-Pfad

Überwachung der integrierten Lösung

Um sicherzustellen, dass Ihre integrierte Lösung optimal funktioniert, sollten Sie die folgenden wichtigen Kennzahlen überwachen:

Kennzahlen zur Kanalmontage

Überwachen Sie die Zeit der Manifestgenerierung, die Programmübergänge und alle Fehler bei der Kanalmontage.

Kennzahlen zur Anzeigeneinfügung

Verfolgen Sie die Füllrate von Anzeigen, die Antwortzeiten des Ad Decision Servers und Fehler beim Einfügen von Anzeigen.

CDN-Metriken

Überwachen Sie die Cache-Trefferraten, das Volumen der ursprünglichen Anfragen und die Antwortlatenz sowohl für Inhalte als auch für Anzeigensegmente.

Kennzahlen zum Zuschauererlebnis

Verfolge Ereignisse, Startzeiten und Zuschauerinteraktion, insbesondere bei Anzeigenübergängen.

Ausführliche Informationen zur Überwachung finden Sie unter [Überwachen Sie den Betrieb für CDN und MediaTailor Integrationen](#) und [Überwachen Sie die CDN-Vorgänge bei der MediaTailor Kanalmontage](#)

Behebung häufiger Probleme

Beachten Sie bei der Behebung von Problemen mit der integrierten Lösung die folgenden häufig auftretenden Probleme:

Probleme bei der Synchronisierung von Werbeunterbrechungen

Wenn Anzeigen nicht an den erwarteten Unterbrechungspunkten erscheinen, überprüfen Sie, ob die Definitionen für Werbeunterbrechungen in Ihren Programmen zur Kanalzusammenstellung korrekt konfiguriert sind und ob beim Einfügen von Anzeigen diese Unterbrechungspunkte korrekt identifiziert werden.

Offensichtliche Fehler bei der Zustellung

Wenn Zuschauer Probleme mit der Wiedergabe haben, überprüfen Sie, ob das CDN die Manifestanfragen korrekt zwischen der Kanalzusammenstellung und der Anzeigeneinfügung weiterleitet und ob die Cache-Einstellungen für die Dynamik dieser Manifeste geeignet sind.

Probleme beim Segment-Routing

Wenn Inhalts- oder Anzeigensegmente nicht geladen werden, überprüfen Sie, ob das CDN Segmentanfragen korrekt an die entsprechenden Ursprünge weiterleitet und ob das Segment URLs in den Manifesten korrekt formatiert ist.

Verschlechterung der Leistung

Wenn bei Zuschauern Pufferung oder hohe Latenz auftritt, überprüfen Sie die CDN-Cache-Trefferraten und das Volumen der ursprünglichen Anfragen, um potenzielle Engpässe in der Bereitstellungspipeline zu identifizieren.

Weitere Anleitungen zur Fehlerbehebung finden Sie unter [Problembehebung bei MediaTailor SSAI mit CDNs für eine unterbrechungsfreie Anzeigenbereitstellung](#)

Best Practices

Folgen Sie diesen bewährten Methoden für eine erfolgreiche Integration von SSAI mit Channel Assembly:

- **Gründlich testen:** Testen Sie die integrierte Lösung mit verschiedenen Inhaltstypen, Anzeigenszenarien und Zuschauerbedingungen, bevor Sie sie in der Produktion einsetzen.
- **Kontinuierliche Überwachung:** Richten Sie umfassende Überwachungs- und Warnmeldungen ein, um auftretende Probleme schnell zu erkennen und zu beheben.
- **Optimieren Sie das Caching:** Überprüfen Sie regelmäßig Ihre CDN-Caching-Einstellungen und passen Sie sie auf der Grundlage der tatsächlichen Nutzungsmuster und Leistungskennzahlen an.
- **Maßstabsgetreue Planung:** Entwerfen Sie Ihre Architektur so, dass sie Spitzenbelastungen bewältigen kann, insbesondere bei beliebten Kanälen oder Veranstaltungen.
- **Erwägen Sie Redundanz:** Implementieren Sie Redundanz in kritischen Komponenten, um eine hohe Verfügbarkeit Ihrer linearen Kanäle sicherzustellen.
- **Optimieren Sie Anzeigenübergänge:** Sorgen Sie für reibungslose Übergänge zwischen Inhalten und Anzeigen, indem Sie konsistente Kodierungsprofile und Segmentdauern verwenden.

Ähnliche Informationen

Weitere Informationen zur Integration von SSAI mit Channel Assembly finden Sie unter:

Dokumentation zur Kanalmontage

[Wird verwendet AWS Elemental MediaTailor , um linear zusammengestellte Streams zu erstellen-](#)
Erfahren Sie mehr über Konzepte zur Kanalmontage

[Kanalmontage mit CDN-](#) Richten Sie die Kanalmontage mit einem CDN ein

SSAI-Dokumentation

[Einfügung von Anzeigen mit CDN-](#) Richten Sie die Anzeigeneinfügung mit einem CDN ein

[Machen Sie sich mit der Architektur der Anzeigeneinfügung für CDN und Integrationen vertraut MediaTailor-](#) Verstehen Sie die CDN-Architektur für Anzeigeneinfügung

CDN-Konfiguration

[Richten Sie die CDN-Integration ein-](#) Allgemeine Hinweise zur CDN-Konfiguration

[CloudFront-Integration-](#) CloudFront spezifische Konfiguration

Optimieren Sie die Leistung für CDN und Integrationen MediaTailor

Maximieren Sie die Leistung Ihrer Implementierung von AWS Elemental MediaTailor Anzeigeneinblendungen, indem Sie Ihre CDN-Konfiguration (Content Delivery Network) optimieren. Diese Einstellungen sorgen für eine effiziente Inhaltsbereitstellung und ein optimales Zuschauererlebnis.

Detaillierte Anleitungen zur Caching- und Routing-Optimierung speziell für SSAI-Implementierungen finden Sie unter. [CDN-Leistungsoptimierung](#) Der konsolidierte Optimierungsleitfaden enthält umfassende Caching-Einstellungen, die Konfiguration des Anforderungsroutings und Leistungsbenchmarks, die für alle CDN-Integrationen gelten. MediaTailor

Allgemeine Leistungsherausforderungen

SSAI-Implementierungen mit CDNs können mit verschiedenen Leistungsherausforderungen konfrontiert sein:

Offensichtlicher Manipulationsaufwand

MediaTailor führt Manifestmanipulationen in Echtzeit durch, was zu Latenz führen kann, wenn sie nicht richtig optimiert wird. Folgendes kann zu Latenz führen:

- Verarbeitungszeit für ADS-Anfragen (AD Decision Server)
- Zeit, die benötigt wird, um Manifeste mit Verweisen auf Anzeigensegmente zu ändern
- Zusätzliche Verarbeitung zur Personalisierung

Probleme mit der Cache-Effizienz

Personalisierte Manifeste können die CDN-Cache-Effizienz aus folgenden Gründen verringern:

- Jeder Zuschauer kann ein einzigartiges Manifest erhalten
- Sitzungsparameter können den Cache fragmentieren
- Dynamischer Inhalt erfordert eine sorgfältige Cache-Konfiguration

Auslastungsspitzen bei Origin

Unsachgemäßes Caching kann zu Lastspitzen beim Ursprung führen bei:

- Ereignisse mit hohem Besucheraufkommen
- Der Cache wird aktualisiert
- Änderungen der CDN-Konfiguration

Probleme bei der Wiedergabe im Zusammenhang mit Werbung

Das Einfügen von Werbung kann zu Störungen bei der Wiedergabe führen, wie z. B.:

- Pufferung bei Anzeigenübergängen
- Qualitätsunterschiede zwischen Inhalten und Anzeigen
- Wiedergabefehler, wenn Anzeigen nicht abgerufen werden können

Umfassende Anleitungen zur Leistungsoptimierung, einschließlich Caching-Strategien, Anforderungsweiterleitung, Leistungsbenchmarks und fortgeschrittenen Optimierungstechniken, finden Sie unter [CDN-Leistungsoptimierung](#). Der konsolidierte Optimierungsleitfaden enthält detaillierte Einstellungen und Benchmarks, die für alle MediaTailor CDN-Integrationen, einschließlich SSAI-Implementierungen, gelten.

Zusammenfassung der bewährten Verfahren

Folgen Sie diesen bewährten Methoden, um eine optimale SSAI-Leistung zu gewährleisten, und zwar mit CDNs:

Bewährte Methoden für die Architektur

- Wählen Sie das richtige Architekturmuster für Ihren Umfang und Ihre Anforderungen
- Stellen Sie Dienste in unmittelbarer Nähe bereit, um die Latenz zu minimieren
- Implementieren Sie Redundanz für kritische Komponenten

Bewährte Methoden für das Zwischenspeichern

- Verwenden Sie unterschiedliche Caching-Strategien für verschiedene Inhaltstypen
- Optimieren Sie die Cache-Schlüssel, um Personalisierung und Effizienz in Einklang zu bringen
- Je TTLs nach Inhaltstyp und Aktualisierungshäufigkeit entsprechend einstellen. Ausführliche TTL-Empfehlungen finden Sie unter [Caching-Optimierung für CDN und Integrationen MediaTailor](#).

Bewährte Methoden für die Anzeigenbereitstellung

- Optimieren Sie ADS-Interaktionen mit Timeouts und Fallbacks
- Bereiten Sie Anzeigen so vor, dass sie den Inhaltsspezifikationen entsprechen
- Implementieren Sie eine effiziente Bereitstellung von Anzeigensegmenten

Überwachung bewährter Verfahren

- Überwachen Sie alle Komponenten Ihrer SSAI-Implementierung
- Richten Sie Warnmeldungen für Leistungseinbußen ein
- Überprüfen und optimieren Sie Ihre Konfiguration regelmäßig

Vollständige Anleitung zur Optimierung

Umfassende Anleitungen zur CDN-Optimierung, einschließlich detaillierter Caching-Strategien, Konfiguration des Anforderungsroutings, Leistungsbenchmarks und fortgeschrittener Optimierungstechniken, finden Sie unter [CDN-Leistungsoptimierung](#). Der konsolidierte Optimierungsleitfaden enthält vollständige Einstellungen und Benchmarks, die für alle MediaTailor CDN-Integrationen, einschließlich SSAI-Implementierungen, gelten.

Überwachen Sie den Betrieb für CDN und MediaTailor Integrationen

AWS Elemental MediaTailor bietet robuste Analysefunktionen, die in Kombination mit CDN-Metriken (Content Delivery Network) umfassende Einblicke in Ihre SSAI-Implementierung bieten. In diesem Thema werden folgende Themen behandelt:

Umfassende Anleitungen zur CDN-Überwachung, einschließlich wichtiger Kennzahlen, Einrichtung von Überwachungstools, Konfiguration von Warnmeldungen und Strategien zur Fehlerbehebung,

die für alle MediaTailor Implementierungen gelten, finden Sie unter. [CDN-Überwachung](#) Dieses Thema konzentriert sich auf SSAI-spezifische Überwachungsanforderungen und Analysen zur Anzeigeneinblendung.

- Überwachungsstrategien für die SSAI- und CDN-Integration
- Analysetools und Methoden zur Datenerfassung
- Datengesteuerte Optimierungstechniken

Wichtige Kennzahlen für SSAI mit CDNs

Um Ihre SSAI-Implementierung effektiv zu überwachen CDNs, sollten Sie diese wichtigen Kennzahlen verfolgen:

Kennzahlen zur Anzeigeneinfügung

Füllrate von Anzeigen: Der Prozentsatz der Werbemöglichkeiten, die erfolgreich mit Anzeigen gefüllt wurden.

Fehlerquote für Anzeigen: Der Prozentsatz der Anzeigenanfragen, die zu Fehlern geführt haben.

Antwortzeit für Anzeigen: Wie lange es dauert, bis der Anzeigenentscheidungsserver auf Anzeigenanfragen reagiert.

Genauigkeit der Anzeigendauer: Wie genau die tatsächliche Dauer eingeblendeter Anzeigen mit der erwarteten Dauer übereinstimmt.

Kennzahlen zum Nutzererlebnis

Pufferquote: Der Prozentsatz der Anzeigezeit, die mit dem Puffern verbracht wurde.

Startzeit: Wie lange es dauert, bis die Videowiedergabe beginnt.

Reibungsloser Übergang von Anzeigen: Wie nahtlos der Player zwischen Inhalten und Anzeigen wechselt.

Sitzungsdauer: Wie lange Zuschauer zuschauen, bevor sie den Stream beenden.

Analysetools und Integration

Kombinieren Sie diese Tools, um eine umfassende Analyselösung für Ihre SSAI-Implementierung zu erstellen:

AWS Elemental MediaTailor serverseitige Metriken

MediaTailor bietet über Amazon integrierte Messwerte, mit CloudWatch denen Anzeigenanfragen, -antworten und -fehler nachverfolgt werden können. Diese Metriken können in der CloudWatch Konsole angezeigt oder in benutzerdefinierte Dashboards integriert werden.

Zu den wichtigsten MediaTailor Kennzahlen gehören:

- `AdDecisionServer.Ads`: Die Anzahl der Anzeigen, die vom Ad Decision Server zurückgegeben wurden.
- `AdDecisionServer.Duration`: Die Gesamtdauer der vom Anzeigenentscheidungsserver zurückgegebenen Anzeigen.
- `AdDecisionServer.Errors`: Die Anzahl der vom Anzeigenentscheidungsserver zurückgegebenen Fehler.
- `AdDecisionServer.Latency`: Die Antwortzeit des Ad Decision Servers.

Eine vollständige Liste der MediaTailor Metriken finden Sie unter [Monitoring MediaTailor with Amazon CloudWatch](#).

CDN-Analytik

CDN-Anbieter bieten detaillierte Analysen zur Leistung der Inhaltsbereitstellung. Verwenden Sie für Amazon CloudFront CloudWatch Metriken und CloudFront Amazon-Zugriffsprotokolle, um Liefermuster zu analysieren.

Wichtige CDN-Metriken zur Überwachung:

- Anzahl der Anfragen nach Inhaltstyp (Manifeste vs. Segmente)
- Cache-Trefferquote für verschiedene Inhaltstypen
- Geografische Verteilung der Zuschauer
- Fehlerquoten nach Fehlercode

Kundenseitiges Tracking

Implementieren Sie clientseitiges Tracking, um Kennzahlen zum Zuschauererlebnis zu sammeln, die serverseitig nicht verfügbar sind:

- Spielerereignisse (Abspielen, Pausieren, Suchen, Puffern)
- Abschlussquoten von Anzeigenaufrufen
- Kennzahlen zur Servicequalität (Änderungen der Auflösung, Bitrate)
- Muster der Zuschauerbindung

Erwägen Sie, diese MediaTailor Kennzahlen mithilfe von clientseitigem Tracking zu sammeln und zu melden.

Integrierte Dashboards

Erstellen Sie umfassende Dashboards, die Kennzahlen aus verschiedenen Quellen kombinieren:

- Verwenden Sie CloudWatch Dashboards, um Metriken zu kombinieren MediaTailor CloudFront
- Ziehen Sie Analyseplattformen von Drittanbietern für eine erweiterte Visualisierung in Betracht
- Richten Sie eine serviceübergreifende Korrelation ein, um Beziehungen zwischen Kennzahlen zu identifizieren

Implementierung einer Überwachungsstrategie

Gehen Sie wie folgt vor, um eine umfassende Überwachungsstrategie für Ihre SSAI mit CDN-Implementierung zu implementieren:

1. Richten Sie die grundlegende Überwachung ein
 - Aktivieren Sie CloudWatch Metriken für MediaTailor
 - Konfigurieren Sie die CDN-Protokollierung und die Erfassung von Metriken
 - Implementieren Sie clientseitiges Tracking in Ihrem Videoplayer
2. Erstellen von benutzerdefinierten Dashboards
 - Erstellen Sie ein CloudWatch Dashboard, das wichtige Kennzahlen kombiniert
 - Fügen Sie Visualisierungen für die Füllrate von Anzeigen, die CDN-Leistung und das Zuschauererlebnis hinzu
 - Fügen Sie Anmerkungen zu wichtigen Ereignissen hinzu (Konfigurationsänderungen, wichtige Übertragungen)
3. Benachrichtigungen konfigurieren
 - Richten Sie CloudWatch Alarme für kritische Messwerte ein
 - Erstellen Sie zusammengesetzte Alarme, die bei mehreren verwandten Bedingungen ausgelöst werden
 - Benachrichtigungskanäle konfigurieren (E-Mail, SMS, Amazon SNS)
4. Implementieren Sie automatisierte Antworten
 - Verwenden Sie CloudWatch Ereignisse, um automatisierte Antworten auf häufig auftretende Probleme auszulösen

- Erstellen Sie Runbooks für manuelle Eingriffe bei Bedarf
- Dokumentieren Sie die Verfahren zur Fehlerbehebung für verschiedene Alarmszenarien

Example Erstellung eines umfassenden SSAI-Monitoring-Dashboards

Dieses Beispiel zeigt, wie Sie ein CloudWatch Dashboard erstellen, das folgende CloudFront Kennzahlen kombiniert MediaTailor :

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

Datengesteuerte Optimierung

Verwenden Sie die Analysedaten, die Sie sammeln, um Ihre SSAI-Implementierung zu optimieren:

CDN-Cache-Optimierung

Analysieren Sie die Cache-Trefferquoten, um Verbesserungsmöglichkeiten zu identifizieren:

- Passen Sie die TTL-Einstellungen je nach Inhaltstyp und Aktualisierungshäufigkeit an.
Ausführliche TTL-Empfehlungen finden Sie unter. [Schritt 1: Konfigurieren Sie das CDN-Caching für eine optimale Anzeigenbereitstellung](#)
- Optimieren Sie die Cache-Schlüsseleinstellungen, um die Cache-Effizienz zu verbessern
- Erwägen Sie die Implementierung von Origin Shield für mehrschichtiges Caching

Optimierung der Anzeigenbereitstellung

Verwenden Sie Kennzahlen zur Anzeigenleistung, um die Anzeigenbereitstellung zu verbessern:

- Identifizieren und beheben Sie häufig auftretende Fehler beim Einfügen von Anzeigen
- Optimieren Sie die Antwortzeiten des Ad Decision Servers
- Passen Sie die Parameter für die Anzeigenausrichtung auf der Grundlage der Analyse der Füllrate an

Optimierung des Zuschauererlebnisses

Verbessern Sie das Zuschauererlebnis auf der Grundlage von kundenseitigen Kennzahlen:

- Analysieren Sie die Abbruchmuster während Werbeunterbrechungen
- Optimieren Sie die Übergangspunkte für Werbeanzeigen für eine reibungslosere Wiedergabe
- Passen Sie die Häufigkeit und Dauer der Anzeigen auf der Grundlage der Daten zur Zuschauerbindung an

Kostenoptimierung

Sorgen Sie für ein ausgewogenes Verhältnis zwischen Leistung und Kosten:

- Analysieren Sie die Muster der Bandbreitennutzung, um die CDN-Kosten zu optimieren
- Erwägen Sie Preisklassenanpassungen für Distributionen CloudFront
- Bewerten Sie das Kosten-Nutzen-Verhältnis verschiedener Caching-Strategien

Best Practices

Folgen Sie diesen bewährten Methoden für eine effektive SSAI-Überwachung und -Analyse:

- **Basiswerte festlegen:** Erfassen Sie während des normalen Betriebs Messwerte, um Leistungsbasiswerte festzulegen, die bei der Fehlerbehebung als Vergleich verwendet werden können.
- **Implementieren Sie eine mehrstufige Überwachung:** Überwachen Sie verschiedene Ebenen Ihrer Architektur (Herkunft, CDN, Player), um sich ein vollständiges Bild von der Leistung zu machen.
- **Korrelieren Sie Kennzahlen zwischen verschiedenen Diensten:** Suchen Sie nach Beziehungen zwischen Metriken verschiedener Dienste, um die Hauptursachen von Problemen zu ermitteln.
- **Verwenden Sie die Anomalieerkennung:** Implementieren Sie die CloudWatch Anomalieerkennung, um ungewöhnliche Muster in Ihren Metriken automatisch zu identifizieren.
- **Regelmäßige Überprüfung und Optimierung:** Planen Sie regelmäßige Überprüfungen Ihrer Überwachungsstrategie und passen Sie sie an sich ändernde Anforderungen und neue Erkenntnisse an.
- **Dokumentieren Sie Ergebnisse und Maßnahmen:** Führen Sie Aufzeichnungen über Optimierungsbemühungen und deren Ergebnisse, um institutionelles Wissen aufzubauen.

Ähnliche Informationen

Weitere Informationen zur Überwachung und Analyse von SSAI mit finden Sie CDNs unter:

- [Optimieren Sie die Leistung für CDN und Integrationen MediaTailor](#) für Techniken zur Leistungsoptimierung
- [Problembehebung bei MediaTailor SSAI mit CDNs für eine unterbrechungsfreie Anzeigenbereitstellung](#) zur Behebung häufig auftretender Probleme
- [Überwachung MediaTailor mit Amazon CloudWatch](#) für detaillierte Informationen zu MediaTailor Metriken
- [Metriken anzeigen CloudFront und Edge-Funktionen](#) für Informationen über CloudFront Metriken

Problembehebung bei MediaTailor SSAI mit CDNs für eine unterbrechungsfreie Anzeigenbereitstellung

In diesem Abschnitt finden Sie Lösungen für häufig auftretende Probleme bei der Verwendung von AWS Elemental MediaTailor dynamischer Anzeigeneinfügung mit einem Content Delivery Network (CDN). Diese Lösungen helfen Ihnen bei der Behebung von Problemen bei der Monetarisierung von Videos durch personalisierte Werbung.

Umfassende Anleitungen zur CDN-Fehlerbehebung, einschließlich Cache-Leistungsproblemen, HTTP-Fehlerbehebung, Testverfahren und Diagnosetechniken, die für alle MediaTailor Implementierungen gelten, finden Sie unter. [Beheben Sie Probleme mit der CDN-Integration](#) Dieser Abschnitt konzentriert sich auf SSAI-spezifische Anforderungen zur Fehlerbehebung und auf Probleme beim Einfügen von Anzeigen.

Wenn Sie Probleme mit Ihrem CDN- und SSAI-Setup haben, überprüfen Sie diese häufig auftretenden Probleme:

Personalisierte Werbung erscheint nicht im Stream

Stellen Sie sicher, dass Ihr ADS korrekt reagiert und mit ihm AWS Elemental MediaTailor kommuniziert. Prüfen Sie diese potenziellen Probleme:

- Die Abfrageparameter für die Ausrichtung von Anzeigen wurden nicht ordnungsgemäß über Ihr CDN weitergeleitet
- Die Breakpoints für Werbeanzeigen sind in Ihren Inhalten falsch definiert
- ADS-Konnektivitäts- oder Antwortprobleme

Wiedergabefehler an Anzeigenunterbrechungen

Stellen Sie sicher, dass die Anzeigensegmente ordnungsgemäß transkodiert sind, damit sie den Bitraten und Auflösungen Ihrer Inhalte entsprechen. Prüfen Sie die folgenden häufig auftretenden Probleme:

- CDN leitet Anfragen für Anzeigensegmente falsch weiter
- Offensichtliche Manipulationsfehler an Übergangspunkten
- Nicht übereinstimmende Kodierungsprofile zwischen Inhalten und Anzeigen

Veraltete Manifeste

Stellen Sie bei Live-Inhalten sicher, dass Ihre TTL-Einstellungen für den CDN-Cache angemessen sind. Verwenden Sie für personalisierte Manifeste eine TTL von 0 Sekunden. Erwägen Sie die Implementierung einer Cache-Invalidierung für sich schnell ändernde Manifeste. Umfassende TTL-Anleitungen finden Sie unter. [Caching-Optimierung für CDN und Integrationen MediaTailor](#)

Hohe Latenz

Überprüfen Sie Ihre CDN-Konfiguration für optimales Routing. Stellen Sie sicher, dass Ihr CDN über Edge-Standorte in der Nähe Ihrer Zuschauer verfügt, um eine optimale Leistung zu erzielen.

Fortgeschrittene Problembehebung

Probieren Sie bei komplexeren Problemen diese fortgeschrittenen Techniken zur Problembehebung aus:

Inkonsistentes Verhalten bei der Ausrichtung von Anzeigen

Prüfen Sie, ob die Abfrageparameter zwischen Spieleranfragen und ADS-Anfragen inkonsistent sind. Stellen Sie sicher, dass alle erforderlichen Targeting-Parameter ordnungsgemäß über das CDN weitergeleitet werden.

Inkonsistenzen im CDN-Cache

Überprüfen Sie die Cache-Schlüsselkonfigurationen, um eine korrekte Inhaltsdifferenzierung sicherzustellen. Erwägen Sie die Implementierung der Cache-Bereinigung für wichtige Manifest-Updates.

Fehler bei der Anzeigenverfolgung

Vergewissern Sie sich, dass URLs die Beacons ordnungsgemäß weitergeleitet und nicht vom CDN blockiert wurden. Stellen Sie sicher, dass Client-Spieler die Tracking-Endpunkte erreichen können.

Leistungsoptimierung

So optimieren Sie die Leistung Ihres Workflows zur dynamischen Anzeigeneinfügung und zur Monetarisierung von Videos:

- Passen Sie die TTL-Einstellungen basierend auf dem Inhaltstyp und den Zuschauermustern an. Ausführliche TTL-Empfehlungen finden Sie unter [Caching-Optimierung für CDN und Integrationen MediaTailor](#)
- Implementieren Sie geografisches Routing, um die Latenz für globale Zielgruppen zu minimieren
- Erwägen Sie, mehrere zu verwenden, CDNs um Redundanz und optimale Leistung zu erzielen
- Überwachen Sie die Cache-Trefferquoten und passen Sie die Konfigurationen entsprechend an
- Optimieren Sie die Manifest-Manipulationsprozesse, um die Verarbeitungszeit an Anzeigenunterbrechungen zu reduzieren
- Transkodieren Sie Anzeigen vorab, sodass sie mit gängigen Inhaltsprofilen übereinstimmen, um eine nahtlose personalisierte Werbeeinblendung zu ermöglichen

Leistungsmaßstäbe

Achten Sie bei der Optimierung der CDN-Integration Ihrer AWS Elemental MediaTailor Anzeigeneinblendung auf die folgenden Leistungsmaßstäbe:

Ziele für die Cache-Trefferquote

Inhaltssegmente: Cache-Trefferquote von mehr als 95%

Anzeigensegmente: Cache-Trefferquote von mehr als 90%

Manifeste: Nicht zutreffend (sollte nicht zwischengespeichert werden, um personalisierte Anzeigen einzufügen)

Benchmarks für die Latenz

Latenz bei manifesten Anfragen: weniger als 100 ms (P95)

Bereitstellung von Inhaltssegmenten: weniger als 50 ms (P95)

Bereitstellung von Anzeigensegmenten: weniger als 75 ms (P95)

End-to-end Startzeit: weniger als 2 Sekunden

Origin-Lade-Metriken

Origin-Anfragen pro Zuschauer: weniger als 0,1 Anfragen pro Minute pro Zuschauer

Ursprungsbandbreite pro Zuschauer: weniger als 5% der gesamten Zuschauerbandbreite

Ziele für die Fehlerquote

Offensichtliche Fehler: weniger als 0,1%

Segmentfehler: weniger als 0,01%

Von Spielern gemeldete Zurückweisungen: weniger als 1%

Benchmarks für Skalierbarkeit

Support für das Zehnfache des normalen Datenverkehrs bei Spitzenereignissen ohne Beeinträchtigung

Fähigkeit, mehr als 1000 Anfragen pro Sekunde und Kanal zu verarbeiten

Verwenden Sie CloudWatch Amazon-Metriken, um diese Leistungsindikatoren zu verfolgen. Detaillierte Anweisungen zur Überwachung finden Sie unter [Überwachung AWS Elemental MediaTailor mit CloudWatch Amazon-Metriken](#).

Ähnliche Informationen

Weitere Informationen zum Einfügen von Anzeigen mit CDNs finden Sie unter:

Dokumentation zum Einfügen von Anzeigen

[Erste Schritte mit der MediaTailor Anzeigenschaltung](#)- Erfahren Sie mehr über Konzepte zur Anzeigenschaltung

[Einrichtung](#)- Beginnen Sie mit der Anzeigenschaltung

CDN-Integration

[Richten Sie die CDN-Integration ein](#)- Allgemeine Hinweise zur CDN-Konfiguration

[CloudFront-Integration](#)- CloudFront spezifische Konfiguration

Integration der Kanalbaugruppe

[Kanalmontage mit CDN](#)- Erfahren Sie mehr über die Kanalmontage mit CDNs

[Implementieren Sie das Einfügen von Anzeigen](#)- Implementieren Sie das Einfügen von Anzeigen bei der Kanalmontage

Überwachung und Optimierung

[Überwachen Sie den Betrieb für CDN und MediaTailor Integrationen](#)- Umfassende Überwachung und Analyse

[Optimieren Sie die Leistung für CDN und Integrationen MediaTailor](#)- Leitfaden zur Leistungsoptimierung

[Überwachung AWS Elemental MediaTailor mit CloudWatch Amazon-Metriken](#)- CloudWatch Metriken für MediaTailor

Erstellen Sie MediaTailor lineare Kanäle mit Kanalmontage und CDN

Dieser Abschnitt enthält umfassende Anleitungen zur Integration von AWS Elemental MediaTailor Channel Assembly in ein Content Delivery Network (CDN). Folgen Sie diesen Schritten, um Ihre Channel Assembly CDN-Integration einzurichten, zu konfigurieren und zu optimieren.

Sie können die Kanalzusammenstellung auch mit serverseitiger Anzeigeneinfügung (SSAI) kombinieren, um monetarisierte lineare Kanäle mit personalisierter Werbung zu erstellen. Diese leistungsstarke Integration ermöglicht es Ihnen, zielgerichtete Werbung für verschiedene Zuschauer zu schalten, die sich denselben Kanalinhalt ansehen. So erhöhen Sie Ihre Umsatzchancen und sorgen gleichzeitig für ein Fernseherlebnis in Sendequalität. Informationen zu SSAI mit finden Sie unter. CDNs [Einfügung von Anzeigen mit CDN](#)

In diesem Thema verwenden wir den Begriff Manifeste, um gemeinsam auf multivariante Playlisten, Medien-Playlisten und zu verweisen. MPDs

Weitere Informationen zur MediaTailor Kanalmontage finden Sie unter. [Wird verwendet AWS Elemental MediaTailor , um linear zusammengestellte Streams zu erstellen](#)

Was Sie benötigen

Bevor Sie die MediaTailor Channel-Assembly mit einem CDN einrichten, sollten Sie die folgenden erforderlichen Ressourcen zusammenstellen:

AWS Konto und Berechtigungen

- Ein AWS Konto mit entsprechenden Berechtigungen zum Erstellen und Verwalten von MediaTailor Ressourcen

- IAM-Berechtigungen für MediaTailor CloudFront (falls verwendet) und verwandte Dienste

- Ausführliche Informationen zu den Genehmigungsanforderungen finden Sie unter [Sicherheit in AWS Elemental MediaTailor](#).

Erforderliche Dienste

- Ein laufender MediaTailor Channel Assembly (nicht nur eine SSAI-Konfiguration)

- Ein CDN-Konto (Content Delivery Network) (Amazon CloudFront oder CDN eines Drittanbieters)

Origin-Speicher für Ihre VOD-Inhalte (Amazon S3 oder ein anderer Ursprungsserver)
MediaPackage

Anforderungen an den Inhalt

VOD-Quellen, die ordnungsgemäß codiert und im HLS- oder DASH-Format verpackt sind.
Informationen zum Arbeiten mit Quellverzeichnissen und VOD-Quellen finden Sie unter [Mit Quellstandorten arbeiten](#)

Inhalt mit einheitlicher Segmentdauer (empfohlener Mindestwert: 1 Sekunde)

Fügen Sie Inhalte für Werbeunterbrechungen hinzu (falls das Einfügen von Werbeanzeigen implementiert ist). Informationen zur Konfiguration von Slate finden Sie unter [MediaTailor Schieber und Einfügung](#).

Bevor Sie beginnen

Important

Für diesen Arbeitsablauf ist ein laufender MediaTailor Kanal zur Kanalmontage erforderlich. Für diese Integration reicht es nicht aus, nur eine MediaTailor SSAI-Konfiguration zu haben. Sie müssen einen aktiven Channel-Assembly-Kanal konfiguriert und betriebsbereit haben, bevor Sie mit der CDN-Integration fortfahren können.

Führen Sie diese Einrichtungsaufgaben aus, bevor Sie die MediaTailor Kanalmontage mit einem CDN implementieren:

1. Konfigurieren Sie die Netzwerkkonnektivität zwischen Ihrem CDN und den MediaTailor Originalservern
2. Richten Sie HTTPS für die sichere Bereitstellung von Inhalten ein
3. Konfigurieren Sie die DNS-Einstellungen für Ihre CDN-Domain

Voraussetzungen für Kenntnisse

Um diese Lösung erfolgreich zu implementieren, sollten Sie über Folgendes verfügen:

- Verständnis von Streaming-Protokollen (HLS/DASH)

- Grundkenntnisse der CDN-Konfigurationsprinzipien
- Vertrautheit mit Konzepten der MediaTailor Kanalmontage

Informationen zur grundlegenden MediaTailor Einrichtung finden Sie unter [Einrichtung](#) und [Erste Schritte mit der MediaTailor Kanalmontage](#). Informationen zum Arbeiten mit Quellverzeichnissen und VOD-Quellen finden Sie unter [Mit Quellstandorten arbeiten](#). Informationen zur Konfiguration von Slate finden Sie unter [MediaTailor Schiefer und Einfügung](#).

Vorteile der CDN-Integration

Die Integration von Channel Assembly in ein CDN bietet diese wichtigen Vorteile.

Verbessertes Zuschauererlebnis

Bei richtiger Konfiguration werden Pufferung, Startzeiten und Wiedergabefehler bei linearen Kanälen CDNs reduziert. Dies führt zu einem höheren Engagement und einer höheren Zufriedenheit der Zuschauer.

Senkung der Kosten

Effiziente Caching-Strategien minimieren die Anzahl der ursprünglichen Anfragen. Dies reduziert die Datenübertragungskosten und die Auslastung des Ursprungsservers, was besonders wichtig für lineare Kanäle mit hohem Volumen ist.

Skalierbarkeit

Optimierte CDN-Konfigurationen bewältigen Verkehrsspitzen bei beliebten Veranstaltungen ohne Leistungseinbußen und stellen so sicher, dass Ihre linearen Kanäle auch zu Spitzenzeiten verfügbar bleiben.

Globale Reichweite

Bei richtiger Konfiguration CDNs können Sie Zuschauern weltweit unabhängig von ihrem Standort Inhalte mit geringer Latenz bereitstellen und so Ihr potenzielles Publikum erweitern.

Reibungslose Programmübergänge

Die optimierte CDN-Konfiguration sorgt für reibungslose Übergänge zwischen Programmen in Ihrem linearen Kanal und sorgt so für ein Seherlebnis in Sendequalität.

Themen

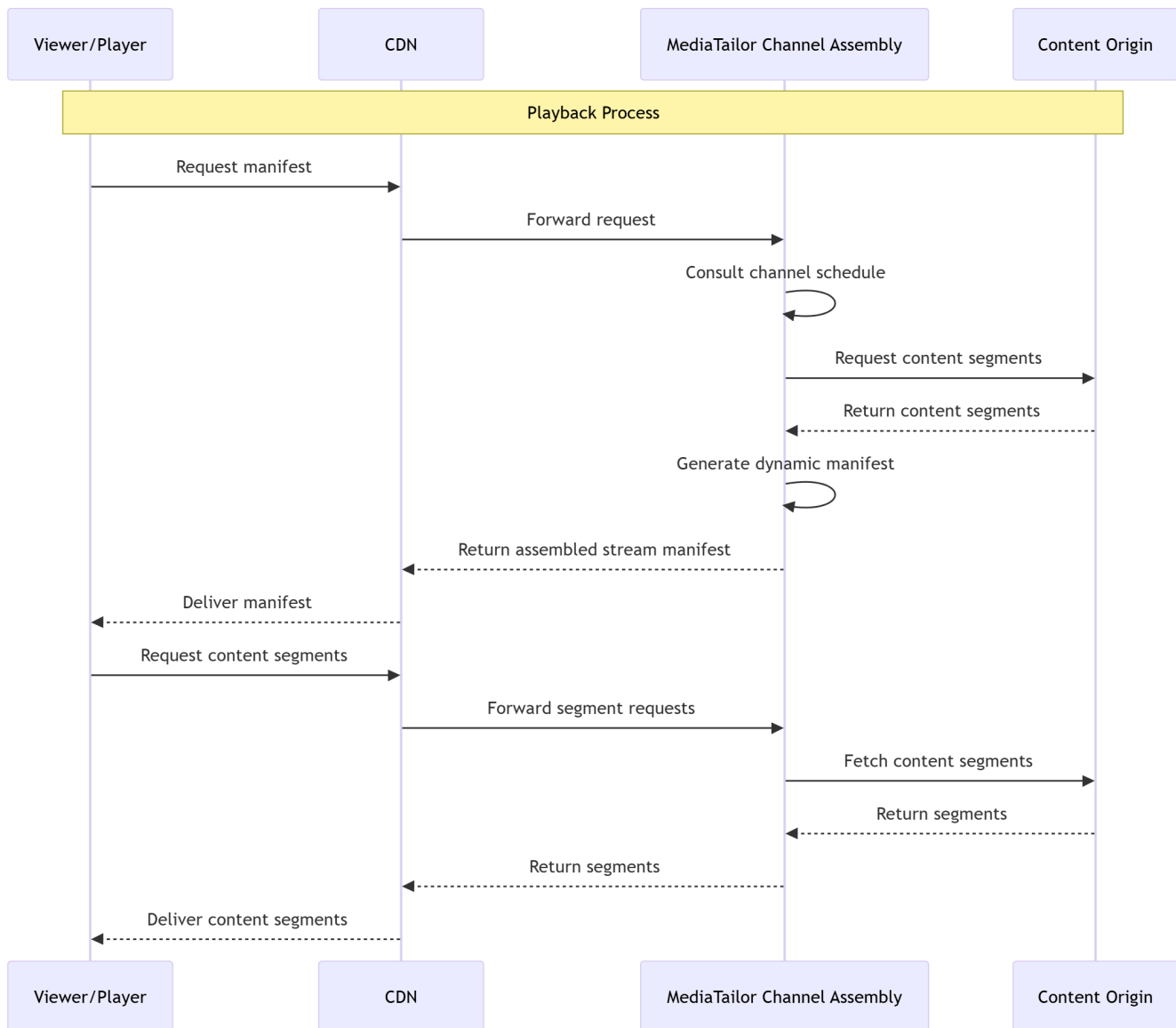
- [Verstehen Sie die CDN-Architektur MediaTailor von Channel Assembly](#)
- [Richten Sie die grundlegende MediaTailor Kanalmontage mit einem CDN ein](#)
- [Konfigurieren Sie die Basis URLs für das MediaTailor CDN zur Kanalmontage](#)
- [Implementieren Sie das Einfügen von Anzeigen mithilfe der Kanalmontage](#)
- [Konfigurieren Sie die zeitversetzte Anzeige für die Kanalmontage MediaTailor](#)
- [Überwachen Sie die CDN-Vorgänge bei der MediaTailor Kanalmontage](#)
- [Schließen Sie die CDN-Optimierung der Channel-Assembly ab](#)

Verstehen Sie die CDN-Architektur MediaTailor von Channel Assembly

AWS Elemental MediaTailor Channel Assembly lässt sich in Content Delivery Networks (CDNs) integrieren, um lineare Streaming-Kanäle mit verbesserter Leistung und globaler Reichweite bereitzustellen. Bei der empfohlenen Architektur wird das CDN zwischen Zuschauern und Channel Assembly positioniert, sodass Channel Assembly auf Inhalte direkt von Ihrer Quelle aus zugreift. In diesem Thema werden die wichtigsten Architekturkomponenten und deren Zusammenspiel bei der Bereitstellung Ihrer Inhalte erläutert.

1. Die Anforderungsmanifeste der Zuschauer aus dem CDN
2. CDN leitet Anfragen an die Channel-Assembly weiter
3. Channel Assembly stellt die Manifeste aus VOD-Quellen zusammen
4. Channel Assembly gibt die Manifeste an das CDN zurück, das sie an die Zuschauer weiterleitet
5. Zuschauer fordern Segmente über das CDN an
6. CDN leitet Segmentanfragen an den entsprechenden Ursprung weiter

Diese Architektur gewährleistet eine optimale Leistung bei gleichzeitiger Beibehaltung der Sicherheits- und Flexibilitätsvorteile der Verwendung eines CDN.



CDN-Terminologie für die Kanalmontage

Wenn Sie diese Schlüsselbegriffe verstehen, können Sie Ihre CDN-Integration für die Kanalmontage implementieren und Fehler beheben:

Origin-CDN und Edge-CDN

Origin-CDN: Ein CDN, das zwischen MediaTailor und Ihrem Inhaltsursprung positioniert ist. Es speichert Inhaltssegmente im Cache, um die Belastung Ihrer Ursprungsserver zu reduzieren. In einer Multi-CDN-Architektur ist dies die erste CDN-Schicht, die direkt mit dem Ursprung verbunden ist.

Edge-CDN: Ein CDN, das zwischen Zuschauern und positioniert ist. MediaTailor Es bietet Zuschauern personalisierte Manifeste und Inhalte. In einer Multi-CDN-Architektur ist dies die äußerste CDN-Schicht, die direkt mit den Zuschauern verbunden ist.

Bedingungen für die CDN-Konfiguration

Cache-Verhalten: Regeln, die festlegen, wie ein CDN verschiedene Arten von Anfragen verarbeitet, einschließlich Cashedauer und Origin-Routing.

TTL (Time To Live): Die Dauer, für die Inhalte in einem CDN-Cache gültig bleiben, bevor sie vom Ursprung aktualisiert werden müssen. Ausführliche TTL-Empfehlungen finden Sie unter. [Caching-Optimierung für CDN und Integrationen MediaTailor](#)

Cache-Schlüssel: Der eindeutige Bezeichner, der von einem CDN zum Speichern und Abrufen zwischengespeicherter Inhalte verwendet wird. Dazu gehören häufig der URL-Pfad, Abfrageparameter und Header.

Origin Shield: Eine zwischengeschaltete Caching-Ebene zwischen CDN-Edge-Standorten und deinem Ursprungsserver, die die Anzahl der Anfragen an deinen Ursprung reduziert.

Zusammenklappen von Anfragen: Eine CDN-Funktion, die mehrere gleichzeitige Anfragen für denselben Inhalt zu einer einzigen ursprünglichen Anfrage kombiniert.

MediaTailor-spezifische CDN-Begriffe

CDN-Inhaltssegmentpräfix: Der CDN-Domänenname, der bei der Generierung URLs von Inhaltssegmenten in Manifesten MediaTailor verwendet wird.

CDN-Anzeigensegmentpräfix: Der CDN-Domainname, der bei der Generierung URLs von Anzeigensegmenten in Manifesten MediaTailor verwendet wird.

Weitere Informationen zur CDN-Konfiguration mit MediaTailor finden Sie unter. [Richten Sie die CDN-Integration ein](#)

Richten Sie die grundlegende MediaTailor Kanalmontage mit einem CDN ein

AWS Elemental MediaTailor Mit Channel Assembly können Sie eine grundlegende Integration mit Ihrem Content Delivery Network (CDN) konfigurieren, um Ihren Zuschauern eine effiziente Bereitstellung linearer Streaming-Kanäle zu ermöglichen. Gehen Sie wie folgt vor, um die Integration zwischen Channel Assembly und Ihrem CDN einzurichten.

1. Konfigurieren Sie Ihr CDN so, dass es Manifeste von Zuschauern akzeptiert und sie an MediaTailor Channel Assembly weiterleitet.
2. Richten Sie MediaTailor Channel Assembly ein, um auf Ihren Kanalplan zuzugreifen und das aktuelle Programm zu ermitteln.
3. Konfigurieren Sie MediaTailor Channel Assembly so, dass Inhaltssegmente auf der Grundlage des Zeitplans von Ihrem Ursprungsserver angefordert werden.
4. Stellen Sie sicher, dass Ihr Inhaltsursprung die angeforderten Segmente an die MediaTailor Channel-Assembly liefern kann.
5. Richten Sie MediaTailor Channel Assembly ein, um dynamische Manifeste auf der Grundlage des aktuellen Zeitplans zu generieren.
6. Konfigurieren Sie Ihr CDN so, dass die zusammengestellten multivarianten Playlisten und MPDs Medien-Playlisten den Zuschauern zur Verfügung gestellt werden.
7. Richten Sie Ihr CDN mit den entsprechenden Cache-Einstellungen so ein, dass es Segmentanfragen von Zuschauern verarbeitet.
8. Konfigurieren Sie Ihr CDN so, dass Cache-Fehlschläge an die MediaTailor Channel-Assembly weitergeleitet werden.
9. Richten Sie die MediaTailor Kanalzusammenstellung ein, um die angeforderten Segmente aus Ihrem Inhaltsursprung abzurufen.
10. Konfigurieren Sie Ihr CDN so, dass Inhaltssegmente den Zuschauern zur Wiedergabe bereitgestellt werden.

Konfigurieren Sie die Basis URLs für das MediaTailor CDN zur Kanalmontage

AWS Elemental MediaTailor Für die Kanalzusammenstellung ist eine korrekte Basis-URL-Konfiguration erforderlich, um sicherzustellen, dass das Routing von Inhalten über Ihr Content Delivery Network (CDN) ordnungsgemäß funktioniert. Konfigurieren Sie die Basis-URL-Einstellungen in der Kanalzusammenstellung, um eine erfolgreiche Inhaltsbereitstellung für Zuschauer zu ermöglichen.

Konfiguration der URL für Inhaltssegmente

Stellen Sie in der Kanalkonfiguration für die Kanalmontage die Basis-URL auf Ihre CDN-Domain ein. Dadurch wird sichergestellt, dass alle Segmente URLs im zusammengestellten Manifest auf Ihr CDN verweisen und nicht direkt auf Ihren Ursprungsserver.

Wenn sich Ihr Originalinhalt beispielsweise auf `http://origin.example.com/content/` und Ihre CDN-Domain befindet `https://cdn.example.com/`, setzen Sie die Basis-URL auf `https://cdn.example.com/content/`

Konfiguration der Zugriffsbeschränkung

Um die Sicherheit zu erhöhen, konfigurieren Sie Ihr CDN so, dass der direkte Zugriff auf Ihren Ursprungsserver eingeschränkt wird:

1. Richten Sie die Origin-Zugriffskontrollen in Ihrem CDN ein.
2. Konfigurieren Sie Ihren Ursprungsserver so, dass er nur Anfragen von Ihrem CDN akzeptiert.
3. Verwenden Sie bei Bedarf signierte URLs oder Cookies für die Zuschauerauthentifizierung.

Für Amazon CloudFront kannst du Origin Access Control (OAC) verwenden, um den Zugriff auf deinen Ursprung zu sichern. Weitere Informationen zur Sicherung deiner CloudFront Integration findest du unter [CloudFront-Integration](#).

Implementieren Sie das Einfügen von Anzeigen mithilfe der Kanalmontage

Channel Assembly AWS Elemental MediaTailor lässt sich nahtlos in serverseitige Anzeigeneinfügung (SSAI) und Content Delivery Networks (CDNs) integrieren, um monetarisierte lineare Kanäle mit personalisierter Werbung zu schaffen.

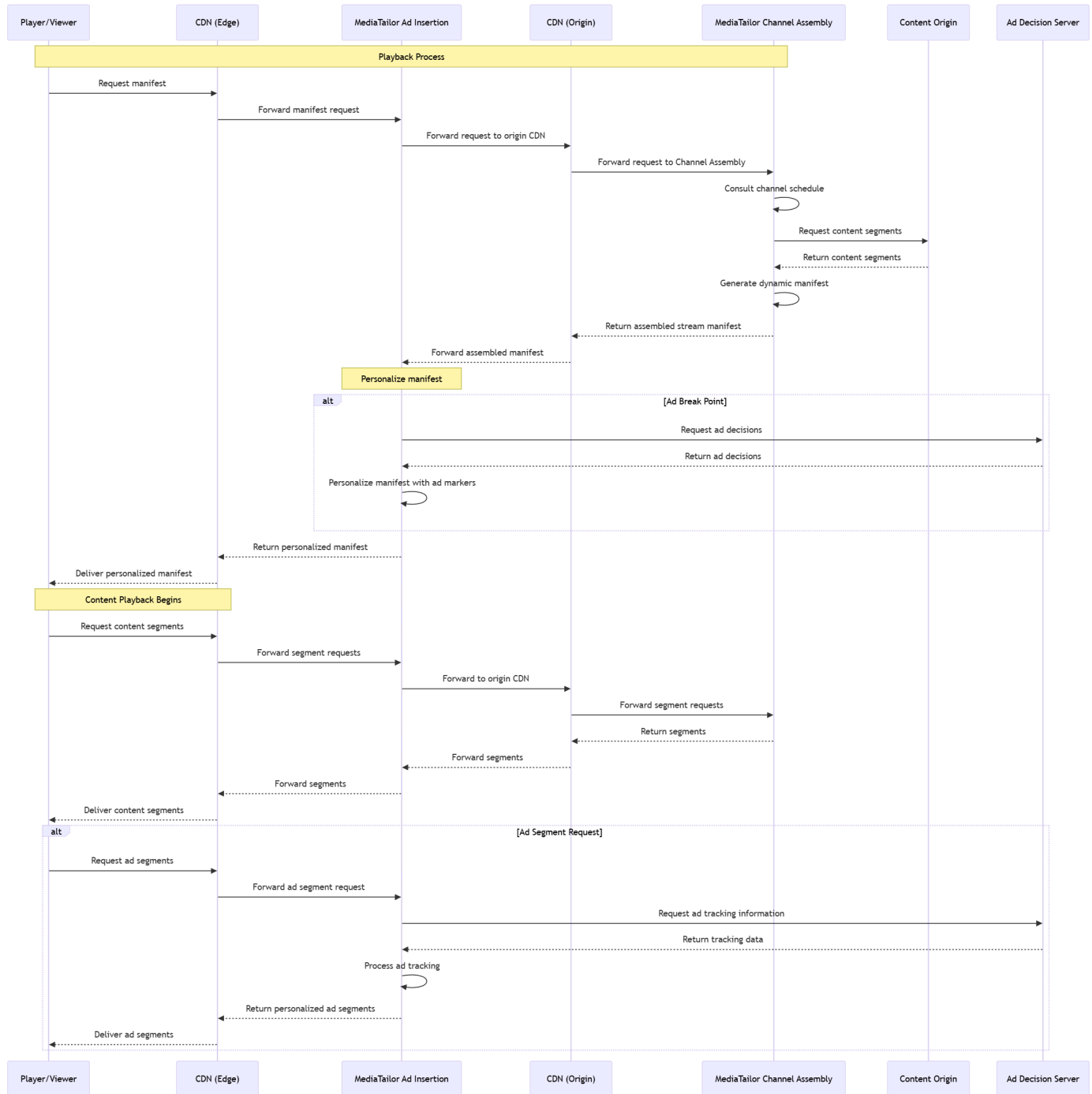
Wenn Sie Channel Assembly mit SSAI kombinieren, können Sie lineare Kanäle aufbauen, die den Zuschauern personalisierte Werbung bieten und gleichzeitig ein qualitativ hochwertiges Erlebnis bieten. Diese Integration ermöglicht Ihnen Folgendes:

- Inhalte monetarisieren — Generieren Sie Einnahmen durch gezielte Werbung in Ihren linearen Kanälen
- Personalize Erlebnisse — Stellen Nutzern, die denselben Kanal ansehen, je nach Profil unterschiedliche Werbeanzeigen bereit
- Behalten Sie die Qualität bei — Sorgen Sie für reibungslose Übergänge zwischen Inhalten und Werbeanzeigen, damit die Inhalte in Sendequalität wiedergegeben werden
- Effizient skalieren — Support Sie Millionen von gleichzeitigen Zuschauern durch CDN-Bereitstellung

Ausführliche Informationen zu SSAI with CDNs finden Sie unter. [Einfügung von Anzeigen mit CDN](#)

1. Konfigurieren Sie Ihr Edge-CDN so, dass es Manifestanfragen von Zuschauern akzeptiert und sie zur MediaTailor Anzeigeneinfügung weiterleitet.
2. Richten Sie die MediaTailor Anzeigeneinfügung so ein, dass Anfragen an Ihr Ursprungs-CDN weitergeleitet werden.
3. Konfigurieren Sie Ihr Ursprungs-CDN so, dass Anfragen an die MediaTailor Channel-Assembly weitergeleitet werden.
4. Richten Sie MediaTailor Channel Assembly ein, um dynamische Manifeste auf der Grundlage des aktuellen Zeitplans zu generieren.
5. Konfigurieren Sie Ihr Ursprungs-CDN so, dass die zusammengestellten Manifeste an die MediaTailor Anzeigeneinfügung weitergeleitet werden.
6. Richten Sie die MediaTailor Anzeigeneinfügung so ein, dass Sie an Anzeigenunterbrechungen Anzeigenentscheidungen von Ihrem Anzeigenentscheidungsserver anfordern.
7. Konfigurieren Sie die MediaTailor Anzeigeneinfügung, um Manifeste mit Anzeigenmarkierungen zu personalisieren.
8. Richten Sie Ihr Edge-CDN ein, um Zuschauern personalisierte Manifeste bereitzustellen.
9. Konfigurieren Sie Ihre CDN-Architektur so, dass sowohl Anfragen nach Inhalten als auch nach Anzeigensegmenten effizient bearbeitet werden.

Das folgende Diagramm veranschaulicht diesen kombinierten Arbeitsablauf:



Für eine optimale Leistung bei der Kombination von Channel-Assembly und SSAI:

- Konfigurieren Sie das Cache-Verhalten, das zwischen Channel-Assembly- und SSAI-Anfragen unterscheidet

- Legen Sie die entsprechenden TTL-Werte für Manifeste und Segmente fest, wie unter empfohlenen [Caching-Optimierung für CDN und Integrationen MediaTailor](#)
- Stellen Sie sicher, dass das Routing zwischen der Kanalmontage, der Anzeigeneinfügung und Ihren CDN-Ursprüngen ordnungsgemäß verläuft
- Überwachen Sie die Leistungskennzahlen sowohl für die Komponenten für die Kanalmontage als auch für die Anzeigeneinfügung

Ausführliche Informationen zur Konfiguration von SSAI mit finden Sie CDNs unter:

- [Machen Sie sich mit der Architektur der Anzeigeneinfügung für CDN und Integrationen vertraut MediaTailor](#)- Erfahren Sie mehr über die Architektur und Konzepte von SSAI
- [Richten Sie grundlegendes MediaTailor SSAI mit einem CDN für eine optimale Anzeigenbereitstellung ein](#)- Anweisungen zur Step-by-step SSAI-Konfiguration
- [Problembehebung bei MediaTailor SSAI mit CDNs für eine unterbrechungsfreie Anzeigenbereitstellung](#)- Beheben Sie häufig auftretende SSAI-Integrationsprobleme

Konfigurieren Sie die zeitversetzte Anzeige für die Kanalmontage MediaTailor

AWS Elemental MediaTailor Channel Assembly unterstützt Funktionen zur zeitversetzten Anzeige, die DVR-ähnliche Funktionen wie Pause, Zurückspulen und Neustarten Ihrer linearen Kanäle ermöglichen. Aktivieren Sie diese Funktionen, indem Sie Ihr Content Delivery Network (CDN) so konfigurieren, dass es zeitversetztes Fernsehen unterstützt, sodass Zuschauer ihr Seherlebnis selbst steuern können.

Grundlegendes zum zeitversetzten Betrachten

Die zeitversetzte Anzeige ermöglicht DVR-ähnliche Funktionen für lineare Kanäle, darunter:

- Neustart: Zuschauer können von Anfang an mit der Wiedergabe einer Sendung beginnen, auch wenn sie erst beitreten, wenn sie gestartet wurde
- Pausieren und fortsetzen: Zuschauer können Inhalte pausieren und die Wiedergabe zu einem späteren Zeitpunkt fortsetzen
- Rücklauf und Schnellvorlauf: Zuschauer können durch die verfügbaren Inhalte rückwärts und vorwärts navigieren

- Verzögerte Wiedergabe: Zuschauer können sich Inhalte ansehen, die zu einem früheren Zeitpunkt im Programm des Kanals ausgestrahlt wurden

Zeitversetztes Ansehen funktioniert, indem der Wiedergabe-URL des Kanals ein `start` Parameter hinzugefügt wird. Der Parameter gibt einen Offset in Sekunden relativ zur aktuellen Zeit an:

- Negative Werte geben eine Zeit in der Vergangenheit an (`start=-3600` bedeutet z. B. „vor 1 Stunde angefangen“)
- Positive Werte geben eine Zeit in der future an (`start=3600` bedeutet z. B. „ab einer Stunde in der future beginnen“)

Beispiel-URL mit Time-Shift-Parameter:

```
https://example-cdn.com/out/v1/channel-name/index.m3u8?start=-3600
```

Auflösung der Zeitverzögerung

So unterstützen Sie die zeitversetzte Anzeige mit einem CDN:

1. Konfigurieren Sie Ihr CDN so, dass der `start` Abfrageparameter an die Channel-Assembly weitergeleitet wird.
2. Richten Sie Cache-Verhalten ein, das den `start` Parameter im Cache-Schlüssel beinhaltet.
3. Verwenden Sie für Manifeste mit Time-Shift-Parametern eine kurze TTL oder kein Caching.

Dadurch wird sichergestellt, dass jeder Betrachter das richtige Manifest für die von ihm angeforderte Zeitposition erhält.

CDN-Anforderungen für Timeshifting

Ihr CDN muss die folgenden Anforderungen erfüllen, um die zeitversetzte Anzeige mit Kanalmontage zu unterstützen:

- Leitet alle Abfrageparameter an die Kanalmontage weiter.
- Nehmen Sie den `start` Parameter in den Cache-Schlüssel auf.
- Support korrekte Cache-Invalidierung für zeitversetzte Manifeste.
- Behandeln Sie unterschiedliche Manifestantworten auf der Grundlage von Abfrageparametern.

Überwachen Sie die CDN-Vorgänge bei der MediaTailor Kanalmontage

AWS Elemental MediaTailor Die Kanalzusammenstellung erfordert eine effektive Überwachung, wenn sie in ein Content Delivery Network (CDN) integriert ist, um eine zuverlässige Inhaltsbereitstellung zu gewährleisten. Implementieren Sie Überwachungsstrategien für Ihre Kanalzusammenstellung und CDN-Integration, um eine zuverlässige Bereitstellung von Inhalten und eine schnelle Problemlösung zu gewährleisten.

Umfassende Anleitungen zur CDN-Überwachung, einschließlich wichtiger Kennzahlen, Einrichtung von Überwachungstools, Konfiguration von Warnmeldungen und Strategien zur Fehlerbehebung, die für alle MediaTailor Implementierungen gelten, finden Sie unter. [CDN-Überwachung](#) Dieser Abschnitt konzentriert sich auf die spezifischen Überwachungsanforderungen für Kanalbaugruppen.

Implementieren Sie eine spezifische Überwachung für Ihre Kanalmontage und CDN-Integration:

- Verfolgen Sie die Metriken zur Manifestgenerierung bei der Kanalzusammenstellung.
- Überwachen Sie zeitversetzte Anzeige Anfragen und deren Auswirkungen auf die CDN-Cache-Trefferraten.
- Konfigurieren Sie Warnmeldungen für ungewöhnliche Muster in Manifestanfragen.
- Implementieren Sie die Nachverfolgung für die Verfügbarkeit von Segmenten in Ihren Inhaltsquellen.

Erstellen Sie für Amazon ein Dashboard CloudFront, das CDN-Metriken mit MediaTailor Metriken integriert, um Ihre gesamte Lieferpipeline mithilfe von [CloudWatchDashboards](#) zu visualisieren.

Wenn Sie SSAI auch für Ihre Channel-Assembly verwenden, finden Sie weitere Empfehlungen zur Überwachung speziell [Überwachen Sie den CDN-Betrieb](#) für die Anzeigenschaltung.

Schließen Sie die CDN-Optimierung der Channel-Assembly ab

Ausführliche Hinweise zur CDN-Optimierung speziell für Channel-Assembly-Implementierungen, einschließlich Caching-Strategien, Konfiguration des Anforderungsroutings, Leistungsbenchmarks und fortgeschrittener Optimierungstechniken, finden Sie unter. [CDN-Leistungsoptimierung](#)

Der konsolidierte Optimierungsleitfaden enthält:

- Kanalassemblyspezifische Caching-Einstellungen mit empfohlenen TTL-Werten
- Fordern Sie eine Routing-Optimierung für die Manifest- und Segmentzustellung an

- Leistungsbenchmarks und -ziele für alle Implementierungen MediaTailor
- Fortgeschrittene Optimierungstechniken wie Origin Shield und Komprimierung
- Kombinierte Workflow-Anleitung für die Kanalmontage mit SSAI

Integrieren MediaTailor mit MediaPackage und CDN

AWS Elemental MediaTailor lässt sich integrieren AWS Elemental MediaPackage , um personalisierte Videoanzeigen über ein Content Delivery Network (CDN) bereitzustellen. MediaPackage ist ein Service für die Verpackung und Erstellung von just-in-time Videos, der Ihre Videoinhalte für die Bereitstellung über das Internet vorbereitet und schützt. Es nimmt Ihre Live- oder On-Demand-Videoinhalte auf und verpackt sie in Streaming-Formate wie HLS und DASH, sodass sie für Zuschauer auf verschiedenen Geräten bereit sind.

Durch die Kombination MediaPackage mit MediaTailor und einem CDN entsteht ein vollständiger Streaming-Workflow, der personalisierte Werbung in großem Umfang bereitstellt. Das CDN verteilt Ihre Inhalte weltweit, reduziert die Latenz und verbessert das Zuschauererlebnis und MediaTailor fügt gleichzeitig gezielte Werbung in Ihre Streams ein.

Dieses Thema konzentriert sich auf die wesentlichen Integrationsschritte MediaTailor, damit MediaPackage, und Ihr CDN zusammenarbeiten. Erweiterte Konfigurationsoptionen, Hinweise zur Fehlerbehebung und zur Überwachung finden Sie unter [Nächste Schritte](#).

Grundlegendes zum MediaPackage und CDN-Workflow

Bevor Sie Ihre Integration konfigurieren, ist es wichtig zu verstehen MediaPackage, wie MediaTailor, und Ihr CDN zusammenarbeiten:

1. **Inhaltsvorbereitung:** MediaPackage empfängt Ihre Live- oder On-Demand-Videoinhalte und verpackt sie in Streaming-Formate (HLS- oder DASH-Manifeste und -Segmente).
2. **Anzeigeneinfügung:** MediaTailor Anfragen werden von Zuschauern MediaPackage versendet, personalisierte Anzeigen eingefügt und die modifizierten Manifeste werden den Zuschauern bereitgestellt.
3. **Weltweite Verbreitung:** Ihr CDN speichert sowohl die Inhaltssegmente (von MediaPackage) als auch die Anzeigensegmente (von) und verteilt sie an Zuschauer auf der MediaTailor ganzen Welt.
4. **Wiedergabe durch Zuschauer:** Videoplayer fordern Manifeste über das CDN an, das Anfragen entsprechend zwischen MediaTailor (für Manifeste) und MediaPackage (für Inhaltssegmente) weiterleitet.

Diese Architektur bietet mehrere Vorteile:

- Skalierbarkeit: Das CDN bewältigt hohe Zuschauerlasten, ohne Ihre Originalserver zu beeinträchtigen
- Leistung: Inhalte werden von Randstandorten bereitgestellt, die den Zuschauern am nächsten sind
- Kosteneffizienz: Reduzierte Bandbreitenkosten durch Caching
- Zuverlässigkeit: Mehrere Edge-Standorte sorgen für Redundanz

Voraussetzungen

Bevor Sie beginnen, stellen Sie sicher, dass Sie die folgenden Komponenten konfiguriert haben:

1. MediaPackage Endpunkt: Ein konfigurierter MediaPackage Endpunkt, der Ihre Videoinhalte empfängt und verpackt. Anweisungen zur Einrichtung finden Sie MediaPackage im MediaPackage Benutzerhandbuch unter [Erste Schritte mit](#).
2. MediaTailor Konfiguration: Eine MediaTailor Konfiguration, die Ihren MediaPackage Endpunkt als Inhaltsquelle verwendet. Anweisungen zur Einrichtung finden Sie unter [Integration einer Inhaltsquelle für die MediaTailor Anzeigeneinfügung](#).
3. CDN-Verteilung: Eine CDN-Distribution (wie CloudFront), die für die Verwendung mit Streaming-Medien konfiguriert ist. Anweisungen zur Einrichtung finden Sie unter [Erstellen einer Distribution](#) im CloudFront Entwicklerhandbuch.
4. Anzeigenentscheidungsserver: Ein konfigurierter Anzeigenentscheidungsserver, der VAST- oder VMAP-Antworten für die Anzeigeneinfügung zurückgibt.

Schritt 1: Konfigurieren Sie die wichtigsten CDN-Einstellungen

Die richtige CDN-Konfiguration ist entscheidend für eine erfolgreiche MediaPackage Integration. Falsche Einstellungen können zu Wiedergabefehlern, schlechter Cache-Leistung und erhöhten Kosten führen. Ohne die richtigen Cache-Richtlinien und die Weiterleitung von Abfrageparametern übermittelt Ihr CDN Manifeste möglicherweise nicht korrekt oder umgeht das Caching vollständig, was zu einer hohen Auslastung des Ursprungsservers und einer verminderten Zuschauererfahrung führt.

Konfigurieren Sie grundlegende Cache-Einstellungen

Die Konfiguration des grundlegenden Cachings ist wichtig, da zur Optimierung der MediaPackage Inhaltsbereitstellung spezielle Cache-Control-Header verwendet werden. Ohne die richtigen Cache-Einstellungen ignoriert Ihr CDN diese Header möglicherweise, was zu unnötigen Ursprungsanfragen und erhöhter Latenz führt. Gehen Sie wie folgt vor, um ein optimales Caching-Verhalten sicherzustellen:

So konfigurieren Sie das grundlegende Caching, das funktioniert mit: MediaPackage

1. Öffnen Sie Ihre CloudFront Distributionseinstellungen in der CloudFront Konsole.
2. Wähle oder erstelle eine Cache-Richtlinie für deinen MediaPackage Origin.
3. Aktiviere die Option „Origin Cache-Control Headers“.
4. Erlaubt es MediaPackage , das Caching-Verhalten über seine Cache-Control-Header zu steuern.

Diese Grundkonfiguration ermöglicht es MediaPackage , die passende Cache-Dauer für verschiedene Inhaltstypen automatisch festzulegen. Um eine erweiterte Cache-Optimierung mit spezifischen TTL-Werten und Leistungsoptimierung zu implementieren, schließen Sie zuerst diese Grundkonfiguration ab und fahren Sie dann fort. [Optimieren Sie das CDN-Caching für MediaTailor und MediaPackage die Bereitstellung von Inhalten](#)

Konfigurieren Sie wichtige Abfrageparameter

Die Konfiguration der Abfrageparameter ist entscheidend für die MediaPackage Funktionalität. Ihr CDN muss bestimmte Abfrageparameter weiterleiten, um Funktionen wie zeitversetzte Anzeige und Streaming mit niedriger Latenz zu ermöglichen. Falsche Einstellungen für Abfrageparameter können verhindern, dass diese Funktionen funktionieren, und ich befürchte, dass die Cache-Effizienz beeinträchtigt wird. Gehen Sie wie folgt vor, um die Weiterleitung von Abfrageparametern zu konfigurieren:

Um sicherzustellen, dass Ihr CDN die erforderlichen Abfrageparameter weiterleitet an: MediaPackage

1. Wählen oder erstellen Sie in Ihren CloudFront Verteilungseinstellungen eine Cache-Richtlinie für Manifestanfragen.
2. Wählen Sie unter „Cache-Schlüsseleinstellungen“ die Option „Spezifizierte Abfragezeichenfolgen einbeziehen“ aus.
3. Fügen Sie die folgenden wichtigen Abfrageparameter hinzu:

- `start` und `end` — Für zeitversetzte Anzeigefunktionen. Diese Parameter werden weitergegeben, MediaPackage um spezifische Inhaltsfenster für die Anzeige von Inhalten für den Start und die Nachholansicht zu definieren.
 - `_HLS_msn` und `_HLS_part` — Für die Unterstützung von LL-HLS-Wiedergabeanfragen
 - `m-` Zum Erfassen der Änderungen am Endpunkt `ti/compame`. MediaPackage Antworten enthalten immer das `?m=###` Tag, um die geänderte Uhrzeit des Endpunkts zu erfassen. Wenn Inhalte bereits mit einem anderen Wert für dieses Tag zwischengespeichert sind, CloudFront wird ein neues Manifest angefordert, anstatt die zwischengespeicherte Version bereitzustellen
 - `aws.manifestfilter`- Für Funktionen zum Filtern von Manifesten. Wenn Sie die Manifestfilterung verwenden, muss dieser Parameter angegeben werden, um die Verteilung so zu konfigurieren, dass die `aws.manifestfilter` Abfragezeichenfolge an den MediaPackage Ursprung weitergeleitet wird. Dies ist erforderlich, damit die Manifestfilterfunktion funktioniert
4. Schließen Sie nur die Abfragezeichenfolgen ein, die MediaPackage verwendet. Das Einbeziehen unnötiger Abfragezeichenfolgen reduziert die Cache-Effizienz, da mehrere Cachevariationen für denselben Inhalt erstellt werden.

Diese Parameter ermöglichen grundlegende MediaPackage Funktionen Ihres CDN. Wenn Sie die Inhaltsfilterung für verschiedene Abonnementstufen oder Gerätetypen implementieren müssen, führen Sie zunächst die Einrichtung dieser grundlegenden Abfrageparameter durch und fahren Sie dann mit fort [Richten Sie die Manifestfilterung mit MediaTailor MediaPackage, und CDN ein](#).

Informationen zur Weitergabe von MediaTailor Abfrageparametern wie `start` und `end` MediaPackage zur zeitversetzten Anzeige finden Sie [MediaTailor Behandlung von Abfrageparametern für Ursprünge](#) unter [MediaTailor Manifest-Abfrageparameter](#).

Konfigurieren Sie das Antwort-Timeout für LL-HLS

Die Timeout-Konfiguration ist für HLS mit niedriger Latenz von entscheidender Bedeutung, da LL-HLS einen Mechanismus zum Blockieren von Anfragen verwendet, bei dem das CDN auf neue Inhaltssegmente wartet. Wenn Ihr Timeout zu kurz ist, schlagen Anfragen fehl, bevor sie mit neuen Segmenten antworten MediaPackage können, was zu Unterbrechungen der Wiedergabe und einem schlechten Zuschauererlebnis führt. Konfigurieren Sie geeignete Timeouts, um eine reibungslose LL-HLS-Wiedergabe zu gewährleisten:

Wenn Sie HLS mit niedriger Latenz verwenden, konfigurieren Sie Ihre CDN-Timeout-Einstellungen:

1. Suchen Sie in Ihren CDN-Einstellungen nach der ursprünglichen Timeout-Konfiguration.

2. Stellen Sie den Wert für das Antwort-Timeout auf mindestens das Dreifache Ihrer Ersatzteildauer ein.
3. Wenn Ihre Ersatzteildauer beispielsweise 0,3 Sekunden beträgt, legen Sie das Timeout auf mindestens 0,9 Sekunden fest.

Dadurch wird sichergestellt, dass das CDN lange genug wartet, bis es reagiert MediaPackage , wenn der Mechanismus zum Blockieren von Anfragen verwendet wird.

Schritt 2: Überprüfen Sie Ihre Integration

Das Testen Ihrer Integration ist unerlässlich, um sicherzustellen, dass alle Komponenten korrekt zusammenarbeiten, bevor bei Ihren Zuschauern Probleme auftreten. Eine fehlgeschlagene Integration kann zu einer fehlerhaften Wiedergabe, fehlenden Anzeigen oder schlechter Leistung führen. Dieser Überprüfungsprozess hilft Ihnen, Probleme in einer kontrollierten Umgebung zu identifizieren und zu lösen.

Stellen Sie nach der Konfiguration Ihrer CDN-Einstellungen sicher, dass Ihre Integration ordnungsgemäß funktioniert, indem Sie den gesamten Workflow von der Inhaltsanfrage bis zur Anzeigeneinblendung testen.

Schritt 2.1: Testen Sie die grundlegende Wiedergabe

Durch grundlegende Wiedergabetests wird überprüft, ob Ihr CDN Manifestanfragen korrekt verarbeitet und an weiterleitet. MediaTailor Dieser Test hilft dabei, Konfigurationsprobleme mit Cache-Richtlinien, der Weiterleitung von Abfrageparametern und der Handhabung von Manifesten zu identifizieren. Gehen Sie wie folgt vor, um die grundlegende Manifestzustellung zu testen:

Testen Sie, ob Ihre Basisintegration funktioniert, indem Sie ein Manifest über Ihr CDN anfordern:

1. Verwenden Sie einen Webbrowser oder Curl, um eine Manifest-URL über Ihr CDN anzufordern.
2. Stellen Sie sicher, dass das Manifest erfolgreich geladen wurde und sowohl Inhalts- als auch Anzeigensegmente enthält.
3. Überprüfen Sie, ob das Inhaltssegment URLs im Manifest auf Ihre CDN-Domain verweist.
4. Vergewissern Sie sich, dass das Anzeigensegment URLs auch auf Ihre CDN-Domain verweist.

Wenn das Manifest korrekt geladen wird und das Erwartete enthält URLs, funktioniert Ihre grundlegende Integration. Umfassende Testmethoden und erweiterte Validierungsverfahren finden Sie unter [Testen und Validieren für CDN und Integrationen MediaTailor](#). Informationen zur Einrichtung

einer umfassenden Überwachung der Leistung und des Zustands Ihrer Integration finden Sie unter [Überwachen Sie die MediaPackage Leistung von CDN und Integrationen MediaTailor](#).

Schritt 2.2: Testen Sie die Videowiedergabe

Tests zur Videowiedergabe stellen sicher, dass Ihre vollständige Integration funktioniert end-to-end, einschließlich der Anzeigeneinfügung und der Bereitstellung von Inhalten über Ihr CDN. Mit diesem Test wird überprüft, ob sowohl Inhaltssegmente als auch Anzeigensegmente ordnungsgemäß zwischengespeichert und bereitgestellt werden und ob das Zuschauererlebnis Ihren Qualitätsstandards entspricht. Gehen Sie wie folgt vor, um die vollständige Wiedergabefunktion zu testen:

Testen Sie, ob die Videowiedergabe bei eingeblendeter Werbung korrekt funktioniert:

1. Verwenden Sie einen Videoplayer (wie Video.js oder HLS.js), um Ihre Inhalte über das CDN abzuspielen.
2. Stellen Sie sicher, dass das Video reibungslos und ohne Pufferprobleme abgespielt wird.
3. Vergewissern Sie sich, dass Werbeanzeigen während der Wiedergabe zu den erwarteten Zeiten eingeblendet werden.
4. Vergewissern Sie sich, dass sowohl Inhalte als auch Anzeigensegmente von Ihrem CDN geladen werden (nicht direkt von Origins).

Wenn die Wiedergabe mit Anzeigen reibungslos funktioniert, funktioniert Ihre Integration ordnungsgemäß. Umfassende Testmethoden und erweiterte Validierungsverfahren finden Sie unter [Testen und Validieren für CDN und Integrationen MediaTailor](#). Falls Sie Probleme mit der Wiedergabe, der Pufferung oder der Anzeigeneinfügung haben, finden Sie weitere Informationen unter [Fehlerbehebung MediaPackage, CDN und Integrationen MediaTailor](#)

Nächste Schritte

Nach Abschluss der Basisintegration können Sie erweiterte Funktionen und Optimierungen implementieren:

Erweiterte CDN-Optimierung

Ausführliche Informationen zur Cache-Optimierung, TTL-Konfiguration und Leistungsoptimierung finden Sie unter [Optimieren Sie das CDN-Caching für MediaTailor und MediaPackage die Bereitstellung von Inhalten](#)

Manifestfilterung

Informationen zur Implementierung von Inhaltsfiltern für mehrstufige Dienste, Geräteoptimierung oder Zugriffskontrolle finden Sie unter [Richten Sie die Manifestfilterung mit MediaTailor MediaPackage, und CDN ein](#).

Fehlerbehebung

Wenn Sie Probleme mit Ihrer Integration haben, finden Sie weitere Informationen unter [Fehlerbehebung MediaPackage, CDN und Integrationen MediaTailor](#).

Überwachung der Leistung

Informationen zur Einrichtung einer umfassenden Überwachung und zum Verständnis der wichtigsten Leistungskennzahlen finden Sie unter [Überwachen Sie die MediaPackage Leistung von CDN und Integrationen MediaTailor](#).

Richten Sie die Manifestfilterung mit MediaTailor MediaPackage, und CDN ein

AWS Elemental MediaTailor verwendet die Manifestfilterung mit AWS Elemental MediaPackage , um festzulegen, welche Audio- und Videostreams in Manifesten enthalten sind, die über ein Content Delivery Network (CDN) an verschiedene Zuschauer gesendet werden. Dies ist besonders nützlich für die Implementierung von mehrstufigen Serviceangeboten, gerätespezifischen Optimierungen oder Inhaltzugriffskontrollen.

Dieses Thema konzentriert sich speziell auf die Implementierung von Funktionen zur Manifestfilterung. Bevor Sie die Manifestfilterung implementieren, müssen Sie die grundlegende Einrichtung der Content Delivery Network-Integration abschließen. Wenn Sie Ihre grundlegende Integration MediaPackage und die Integration des Content Delivery Network noch nicht eingerichtet haben, beginnen Sie mit [Integrieren MediaTailor mit MediaPackage und CDN](#).

Funktionen zum Filtern von Manifesten

Bevor Sie die Manifestfilterung implementieren, sollten Sie sich darüber im Klaren sein, was Sie mit dieser Funktion erreichen können:

Die wichtigsten Filterfunktionen

Die Manifestfilterung bietet mehrere wichtige Funktionen, mit denen Sie die Inhaltsbereitstellung kontrollieren können:

- Beschränken Sie den Zuschauerzugriff auf Premium-Inhalte (wie 4K HEVC)
- Richten Sie sich mit entsprechenden Streams auf bestimmte Gerätetypen
- Filtern Sie Inhalte auf der Grundlage von Audio-Sampleraten, Sprachen oder Videocodecs
- Stellen Sie unterschiedlichen Abonnenten unterschiedliche Qualitätsstufen zur Verfügung

Häufige Anwendungsfälle

Diese Anwendungsfälle zeigen, wie die Manifestfilterung bestimmten Geschäftsanforderungen gerecht werden kann:

Abonnementstufen

Bieten Sie Basisabonnenten Streams mit niedrigerer Auflösung an und gewähren Sie Premium-Abonnenten gleichzeitig Zugriff auf 4K-Inhalte

Beispiel: Der Basis-Tarif ist auf 720p begrenzt, der Premium-Tarif auf bis zu 4K

Geräteoptimierung

Automatische Bereitstellung geeigneter Streams auf der Grundlage der Gerätefunktionen

Beispiel: Mobilgeräte haben niedrigere Bitraten, intelligente Geräte TVs erhalten eine höhere Qualität

Verwaltung der Bandbreite

Beschränken Sie die Stream-Qualität zu Spitzenzeiten, um die Netzwerkkosten zu senken

Beispiel: Reduzieren Sie die maximale Bitrate bei Ereignissen mit hohem Datenaufkommen

Regionaler Inhalt

Bieten Sie je nach Standort des Betrachters unterschiedliche Audiosprachen oder Inhaltsvarianten an

Beispiel: Automatisches Filtern nach Audiotracks in der Landessprache

Weitere Informationen zu Konzepten der Manifestfilterung finden Sie im AWS Elemental MediaPackage Benutzerhandbuch unter [Manifestfilterung](#).

Konfigurieren Sie Ihr CDN für die Manifestfilterung

Die CDN-Konfiguration für die Manifestfilterung ist wichtig, da Ihr CDN den `aws.manifestfilter` Abfrageparameter weiterleiten muss, damit die Filterung MediaPackage funktioniert. Ohne die korrekte Weiterleitung der Abfragezeichenfolge werden die Filterparameter vom CDN entfernt, und alle Zuschauer erhalten ungefilterte Manifeste, unabhängig von ihrer Abonnementstufe oder ihren Gerätefunktionen. Diese Konfiguration stellt sicher, dass Ihre Filterlogik erreicht MediaPackage und wie vorgesehen funktioniert.

Um die Manifestfilterung über Ihr CDN zu aktivieren, müssen Sie die Weiterleitung von Abfragezeichenfolgen konfigurieren:

1. Erstellen oder bearbeiten Sie in Ihrer CloudFront Distribution das Cache-Verhalten für Manifestanfragen.
2. Erstellen Sie für die Cache-Richtlinie eine neue Richtlinie oder bearbeiten Sie eine vorhandene.
3. Wählen Sie unter Cache-Schlüsseinstellungen die Option „Spezifizierte Abfragezeichenfolgen einbeziehen“.
4. `aws.manifestfilter` Zur Liste der zulässigen Abfragezeichenfolgen hinzufügen.
5. Wenn Sie auch andere MediaPackage Funktionen verwenden, fügen Sie deren Abfrageparameter hinzu:
 - `startund end` - Für zeitversetzte Anzeige
 - `time_delay`- Für die Funktion zur Zeitverzögerung
 - `_HLS_msnund _HLS_part` - Für LL-HLS

Weitere Informationen zum Erstellen von Distributionen finden Sie unter [Eine Distribution erstellen](#) im CloudFront Amazon-Entwicklerhandbuch.

Implementieren Sie eine clientseitige Filterung

Bei der clientseitigen Implementierung definieren Sie, wie Ihre Videoplayer und Anwendungen gefilterte Inhalte anfordern. Diese Konfiguration bestimmt, welche Inhalte jeder Zuschauer auf der Grundlage seiner Abonnementstufe, der Gerätefunktionen oder anderer Kriterien erhält. Durch die richtige Implementierung wird sichergestellt, dass Zuschauer nur die Inhalte erhalten, auf die sie Zugriff haben sollten, und gleichzeitig eine optimale CDN-Cache-Effizienz beibehalten.

So implementieren Sie die Manifestfilterung in Ihren Videoplayern und Anwendungen:

So funktioniert das Filtern

Der Filtervorgang funktioniert wie folgt:

1. Ihr Videoplayer oder Ihre Anwendung fordert eine Manifest-URL an, die Filterparameter enthält
2. Das CDN leitet die Anfrage (einschließlich Abfrageparameter) weiter an MediaTailor
3. MediaTailor übergibt die Filterparameter an, MediaPackage wenn das Ursprungsmanifest angefordert wird
4. MediaPackage wendet die Filter an und gibt ein benutzerdefiniertes Manifest zurück, das nur die Varianten enthält, die Ihren Kriterien entsprechen
5. MediaTailor verarbeitet das gefilterte Manifest für die Anzeigeneinfügung und gibt es an den Player zurück

URL-Format für die Filterung

Das Verständnis des richtigen URL-Formats ist entscheidend für eine erfolgreiche Implementierung der Filterung. Eine falsche URL-Formatierung führt dazu, dass Filterparameter ignoriert werden oder HTTP-Fehler verursacht werden. Die URL-Struktur muss Filterparameter als Abfragezeichenfolgen enthalten, an die Ihr CDN weiterleitet. MediaPackage Gehen Sie wie folgt vor, um die richtige URL-Formatierung zu implementieren:

So implementieren Sie die Manifestfilterung in Ihren Videoplayern:

1. Ändern Sie die Manifestanforderung Ihres Players so URLs , dass sie die entsprechenden Filterparameter enthält.
2. Verwenden Sie das folgende URL-Format mit Abfrageparametern:

```
https://CloudFront-Domain/v1/master/MediaTailor-Config/index.m3u8?  
aws.manifestfilter=video_codec:h264;audio_language:en-US
```

3. Wenn Ihr Player diese URL anfordert, MediaTailor werden diese Parameter an übergeben MediaPackage, was zu einem gefilterten Manifest führt.

Allgemeine Filterszenarien

Verwenden Sie diese Beispiele, um gängige Filterszenarien zu implementieren:

Gerätespezifische Bereitstellung von Inhalten

Um nach Gerätefunktionen zu filtern, fügen Sie Ihrer Manifestanfrage diesen Parameter hinzu:

```
aws.manifestfilter=video_codec:h264;audio_sample_rate:0-44100
```

In diesem Beispiel wird der Inhalt auf H.264-Video und -Audio mit Abtastraten von bis zu 44,1 kHz beschränkt, was für mobile Geräte geeignet ist.

Beschränkung auf Premium-Inhalte

Um den Zugriff auf Inhalte mit hoher Bitrate einzuschränken, fügen Sie Ihrer Manifestanfrage diesen Parameter hinzu:

```
aws.manifestfilter=video_bitrate:0-9000000
```

In diesem Beispiel werden Videobitraten auf 9 Mbit/s oder weniger beschränkt, was für grundlegende Abonnementstufen geeignet ist.

Auswahl der Sprache

Um nach bestimmten Audiosprachen zu filtern, fügen Sie diesen Parameter zu Ihrer Manifestanfrage hinzu:

```
aws.manifestfilter=audio_language:fr,en-US,de
```

Dieses Beispiel umfasst nur französische, US-amerikanische und deutsche Audiotracks.

Zielauflösung

Um nach bestimmten Videoauflösungen zu filtern, fügen Sie Ihrer Manifestanfrage diesen Parameter hinzu:

```
aws.manifestfilter=video_height:240-360,720-1080
```

Dieses Beispiel umfasst Videostreams mit Höhen zwischen 240 und 360 Pixeln und 720 bis 1080 Pixeln, ausgenommen Auflösungen im mittleren Bereich.

Codec-basierte Filterung

Um nach bestimmten Videocodecs zu filtern, fügen Sie Ihrer Manifestanforderung diesen Parameter hinzu:

```
aws.manifestfilter=video_codec:h264,h265
```

Dieses Beispiel umfasst nur H.264- und H.265-Videostreams, ausgenommen andere Codecs.

Besondere Überlegungen und Einschränkungen

So vermeiden Sie häufig auftretende Probleme bei der Implementierung der Manifestfilterung:

Technische Einschränkungen

- Verwenden Sie für TS-Manifeste Audiowiedergabegruppen, um zu vermeiden, dass Videostreams entfernt werden, die mit herausgefilterten Audiostreams gemultiplext wurden
- In TS- und CMAF-Manifesten sind die Audio-Samplerate und die Videobitrate im Manifest zur Überprüfung nicht ohne weiteres sichtbar
- Anforderungsparameter, die an Medien-Playlisten oder -segmente angehängt werden, führen zu einem HTTP 400-Fehler

Fehlerbedingungen

- Wenn das Filtern zu einem leeren Manifest führt (keine Streams erfüllen die Filterkriterien), MediaPackage wird ein HTTP 400-Fehler zurückgegeben
- Widersprüchliche Filterkonfigurationen (Endpunktfiler + Abfrageparameter) führen zu HTTP 404-Fehlern
- Ungültige Filtersyntax oder nicht unterstützte Filtertypen führen zu HTTP 400-Fehlern

Leistungsaspekte

- Jede eindeutige Filterkombination erstellt einen separaten Cache-Eintrag, wodurch die Cache-Effizienz möglicherweise beeinträchtigt wird
- Komplexe Filter mit vielen Kriterien können die Leistung der Manifestgenerierung beeinträchtigen
- Erwägen Sie die Verwendung von Filtern auf Endpunktebene für statische Filterszenarien, um die Cache-Leistung zu verbessern

Testen Sie Ihre Filterimplementierung

Das Testen Ihrer Implementierung der Manifestfilterung ist entscheidend, um sicherzustellen, dass Zuschauer die richtigen Inhalte erhalten, die auf ihrer Zugriffsebene und ihren Gerätefunktionen basieren. Fehlgeschlagene Filterung kann dazu führen, dass Zuschauer falsche Qualitätsstufen, nicht unterstützte Formate oder Inhalte erhalten, auf die sie keinen Zugriff haben sollten. Umfassende Tests helfen dabei, diese Probleme zu identifizieren und zu lösen, bevor sie sich auf Ihre Zuschauer auswirken.

Gehen Sie wie folgt vor, um zu überprüfen, ob Ihre Manifestfilterung ordnungsgemäß funktioniert:

1. Fordern Sie Manifeste mit unterschiedlichen Filterparametern an und überprüfen Sie die Ergebnisse
2. Stellen Sie sicher, dass gefilterte Manifeste nur die erwarteten Streams enthalten
3. Testen Sie Randfälle (leere Ergebnisse, ungültige Filter), um eine korrekte Fehlerbehandlung sicherzustellen
4. Stellen Sie sicher, dass Ihr CDN die Filterparameter ordnungsgemäß weiterleitet
5. Testen Sie mit verschiedenen Geräten und Playern, um die Kompatibilität sicherzustellen

Informationen zur Behebung von Filterproblemen finden Sie unter [Beheben von Problemen mit der MediaPackage CDN-Integration](#).

Wenn Sie auf HTTP 400-Fehler, leere Manifeste oder Filterparameter stoßen, die nicht wie erwartet funktionieren, finden Sie [Fehlerbehebung MediaPackage, CDN und Integrationen MediaTailor](#) spezifische Anleitungen zur Fehlerbehebung bei der Manifestfilterung.

Optimieren Sie das CDN-Caching für MediaTailor und MediaPackage die Bereitstellung von Inhalten

AWS Elemental MediaTailor erfordert eine korrekte Caching-Konfiguration für das Content Delivery Network (CDN), um eine optimale Leistung zu erzielen, wenn Sie es AWS Elemental MediaPackage als Inhaltsquelle verwenden. MediaPackage bietet spezielle Cache-Control-Header, die Ihrem Content Delivery Network mitteilen, wie lange verschiedene Inhaltstypen zwischengespeichert werden sollen. Die Einhaltung dieser Empfehlungen gewährleistet eine reibungslose Wiedergabe und effiziente Bereitstellung von Inhalten.

Dieses Thema konzentriert sich speziell auf die Optimierung des Caching-Verhaltens, um die Leistung zu maximieren und die Kosten zu minimieren. Stellen Sie vor der Implementierung der

erweiterten Caching-Optimierung sicher, dass Sie die grundlegende Einrichtung der Content Delivery Network-Integration abgeschlossen haben. Wenn Sie Ihre Basisintegration noch nicht eingerichtet haben, beginnen Sie mit [Integrieren MediaTailor mit MediaPackage und CDN](#).

MediaPackage Header zur Cache-Steuerung

MediaPackage legt spezifische TTL-Werte für verschiedene Inhaltstypen fest, um das Caching-Verhalten zu optimieren:

Multivariante Playlisten (HLS und LL-HLS)

TTL: Die Hälfte der Dauer der Mediensegmente

Grund: Diese Playlisten ändern sich, wenn neue Segmente verfügbar werden. Sie müssen daher häufig aktualisiert werden

Medien-Playlisten (reguläres HLS)

TTL: Die Hälfte der Dauer der Mediensegmente

Grund: Ähnlich wie multivariante Playlisten werden diese mit fortschreitendem Inhalt aktualisiert

Medien-Playlisten (LL-HLS)

TTL: 1 Sekunde

Grund: Streaming mit niedriger Latenz erfordert sehr häufige Updates

TS-Mediensegmente und Init-Segmente

TTL: 1209600 Sekunden (14 Tage)

Grund: Mediensegmente ändern sich nicht, nachdem sie einmal erstellt wurden, sodass sie für längere Zeiträume zwischengespeichert werden können

CMAF-Mediensegmente und Initialisierungssegmente

TTL: 1209600 Sekunden (14 Tage)

Grund: Wie TS-Segmente sind auch diese unveränderlich, sobald sie erstellt wurden

Umfassende TTL-Empfehlungen für alle MediaTailor Workflows und zusätzliche Strategien zur Caching-Optimierung finden Sie unter [Caching-Optimierung für CDN und Integrationen MediaTailor](#)

Konfigurieren Sie CDN-Cache-Richtlinien

Die richtige Konfiguration der Cache-Richtlinien ist für optimale Leistung und Kosteneffizienz unerlässlich. Verschiedene Inhaltstypen (Manifeste, Segmente, Initialisierungsdateien) haben unterschiedliche Caching-Anforderungen. Durch die Verwendung separater Cache-Verhaltensweisen können Sie das Caching für jeden Inhaltstyp optimieren, wodurch die Cache-Trefferquoten verbessert und die Ausgangslast reduziert werden. Ohne die richtigen Cache-Richtlinien kann es zu unnötigen Anfragen an die Herkunft, erhöhten Kosten und einer schlechten Wiedergabeleistung kommen.

Gehen Sie wie folgt vor, um MediaPackage Cache-Control-Header korrekt zu berücksichtigen und das Caching zu optimieren:

1. Öffnen Sie Ihre CloudFront Distributionseinstellungen in der Konsole. CloudFront
2. Erstellen Sie separate Cache-Verhaltensweisen für verschiedene Inhaltstypen:
 - Manifeste Anfragen (*.m3u8, *.mpd)
 - Mediensegmente (*.ts, *.mp4, *.m4s)
 - Initialisierungssegmente
3. Erstellen oder wählen Sie für jedes Cache-Verhalten eine Cache-Richtlinie mit den folgenden Einstellungen aus:
 - Aktivieren Sie die Option „Origin Cache-Control Headers“
 - Stellen Sie die „Origin-Anforderungsrichtlinie“ ein, um die erforderlichen Header weiterzuleiten
 - Konfigurieren Sie die Weiterleitung von Abfragezeichenfolgen basierend auf dem Inhaltstyp

Manifestiertes Cache-Verhalten

Für Manifestanfragen (*.m3u8, *.mpd):

- Pfadmuster: *.m3u8 und *.mpd
- Cache-Richtlinie: Beachten Sie die ursprünglichen Cache-Control-Header
- Abfragezeichenfolgen: Spezifische Parameter weiterleiten (siehe [Optimieren Sie die Weiterleitung von Abfragestrings](#))
- Header: Alle Header weiterleiten (Mindestanforderungen finden Sie unter) [Erforderliche Header für MediaTailor die CDN-Integration](#)

Verhalten des Mediensegment-Cache

Für Mediensegmente (*.ts, *.mp4, *.m4s):

- Pfadmuster: *.ts, *.mp4, *.m4s
- Cache-Richtlinie: Beachten Sie die ursprünglichen Cache-Control-Header (14-Tage-TTL)
- Abfragezeichenfolgen: Keine (Segmente verwenden keine Abfrageparameter)
- Komprimierung: Aktivieren Sie diese Option, um die Lieferleistung zu verbessern

Optimieren Sie die Weiterleitung von Abfragestrings

Die Optimierung von Abfragezeichenfolgen ist für die Cache-Effizienz von entscheidender Bedeutung, da unnötige Abfrageparameter mehrere Cachevariationen für denselben Inhalt erzeugen. Jede eindeutige Kombination von Abfrageparametern erstellt einen separaten Cache-Eintrag, wodurch die Cache-Trefferquoten reduziert und die Anzahl der ursprünglichen Anfragen erhöht wird. Indem Sie nur die Abfragezeichenfolgen weiterleiten, die MediaPackage tatsächlich verwendet werden, maximieren Sie die Cache-Effizienz und behalten gleichzeitig die volle Funktionalität bei.

Konfigurieren Sie Ihr CDN so, dass nur die MediaPackage verwendeten Abfragezeichenfolgen weitergeleitet werden, was die Cache-Effizienz verbessert:

Wesentliche Abfragezeichenfolgen

`startund end` - Für zeitversetzte Sichtfenster

`time_delay` - Zum Anwenden von Zeitverzögerungen bei Manifestinhalten

`_HLS_msn_HLS_m`, und `_HLS_part` - Für LL-HLS-Wiedergabeanfragen

Funktionsspezifische Abfragezeichenfolgen

`aws.manifestfilter` - [Für die Manifestfilterung](#)

Important

Nehmen Sie keine anderen Abfragezeichenfolgen in Ihren Cache-Schlüssel auf. MediaPackage ignoriert unbekannte Parameter und deren Einbeziehung verringert die Cache-Effizienz, da unnötige Cachevariationen erzeugt werden.

Techniken zur Leistungsoptimierung

Diese Optimierungen werden auf Ihrem CDN (z. B. CloudFront) konfiguriert, nicht in MediaPackage oder MediaTailor. Implementieren Sie diese zusätzlichen Optimierungen, um die Cache-Leistung zu maximieren:

Origin-Schild

Origin Shield bietet eine zusätzliche Caching-Ebene zwischen deinen CDN-Edge-Standorten und MediaPackage Endpunkten. Dadurch wird die Anzahl der Anfragen, die Ihre MediaPackage Endgeräte erreichen, reduziert, was die Leistung verbessern und die Kosten senken kann, insbesondere bei Verkehrsspitzen oder wenn die Cache-Trefferquoten unter dem Optimum liegen. Origin Shield ist besonders vorteilhaft für Live-Streaming, bei dem mehrere Edge-Standorte möglicherweise gleichzeitig dieselben Inhalte anfordern.

Aktiviere Origin Shield, um die Belastung deiner MediaPackage Endgeräte zu reduzieren:

1. Aktiviere in deiner CloudFront Distribution Origin Shield für deinen MediaPackage Ursprung.
2. Wähle eine Origin Shield-Region in der Nähe deines MediaPackage Endpunkts aus.
3. Dadurch entsteht eine zusätzliche Caching-Ebene, auf die Anfragen reduziert werden MediaPackage.

Konfiguration der Komprimierung

Komprimierung für textbasierte Antworten aktivieren:

- Aktivieren Sie die Komprimierung für Manifestdateien (*.m3u8, *.mpd)
- Komprimieren Sie keine Mediensegmente (die bereits komprimiert sind)
- Stellen Sie sicher, dass alle Header weitergeleitet werden MediaPackage (Mindestanforderungen finden Sie unter [Erforderliche Header für MediaTailor die CDN-Integration](#))

Überwachen Sie die Cache-Leistung

Verfolgen Sie diese wichtigen Kennzahlen, um eine optimale Cache-Leistung sicherzustellen:

Trefferquote im Cache

Ziel: 90% oder mehr für Mediensegmente, 70% oder mehr für Manifeste

Niedrige Werte können auf falsche TTL-Einstellungen oder unnötige Abfrageparameter hinweisen
Volumen der ursprünglichen Anfrage

Überwachen Sie Anfragen, die MediaPackage Endpunkte erreichen

Hohe Volumen können auf Caching-Probleme hinweisen

Schlüsselvariationen zwischenspeichern

Überprüfen Sie die Cache-Schlüsselmuster, um unnötige Variationen zu identifizieren

Zu viele Variationen verringern die Cache-Effizienz

Richten Sie nach der Implementierung dieser Cache-Optimierungen eine Überwachung ein, um deren Effektivität zu verfolgen. Hinweise zur Überwachung der Cache-Trefferquoten, der ursprünglichen Anforderungsmuster und anderer wichtiger Leistungskennzahlen finden Sie unter [Überwachen Sie die MediaPackage Leistung von CDN und Integrationen MediaTailor](#). Wenn Sie eine schlechte Cache-Leistung oder unerwartete Anfragen bei der Herkunft feststellen, finden Sie unter Schritte [Fehlerbehebung MediaPackage, CDN und Integrationen MediaTailor](#) zur Fehlerbehebung weitere Informationen.

Überwachen Sie die MediaPackage Leistung von CDN und Integrationen MediaTailor

AWS Elemental MediaTailor erfordert eine effektive Überwachung, um die optimale Leistung Ihrer Integration AWS Elemental MediaPackage und der CDN-Integration (Content Delivery Network) aufrechtzuerhalten. In diesem Thema finden Sie Anleitungen zu den wichtigsten Kennzahlen, die Sie verfolgen sollten, zu verwendenden Überwachungstools und zur Einrichtung von Warnmeldungen für die proaktive Problemerkennung.

Stellen Sie vor der Einrichtung der Überwachung sicher, dass Ihre Basisintegration ordnungsgemäß funktioniert. Wenn Sie die grundlegende Einrichtung der Content Delivery Network-Integration noch nicht abgeschlossen haben, beginnen Sie mit [Integrieren MediaTailor mit MediaPackage und CDN](#). Informationen zur Behebung von Problemen, die im Rahmen der Überwachung festgestellt wurden, finden Sie unter [Fehlerbehebung bei der CDN-Integration](#).

Wichtige Leistungskennzahlen

Überwachen Sie diese wichtigen Kennzahlen, um eine optimale Leistung Ihrer MediaPackage und der CDN-Integration sicherzustellen:

CDN-Leistungskennzahlen

Umfassende CDN-Leistungskennzahlen, einschließlich der Zielvorgaben für die Cache-Trefferquote, Überwachung des Volumens der ursprünglichen Anfragen und Benchmarks für die Antwortzeit, finden Sie [Leistungsbenchmarks für CDN und Integrationen MediaTailor](#) im CDN-Optimierungsleitfaden.

Wichtige EMP-spezifische Überlegungen zu CDN-Metriken:

EMP-Header zur Cache-Steuerung

Was Sie überprüfen sollten: Stellen Sie sicher, dass Ihr CDN die Cache-Control-Header von EMP berücksichtigt, um ein optimales TTL-Verhalten zu gewährleisten

Erwartetes Verhalten: Verschiedene Inhaltstypen sollten je nach EMP-Headern unterschiedliche Cashedauern haben

Eine ausführliche Anleitung zur EMP-Cache-Optimierung finden Sie unter. [Optimieren Sie das CDN-Caching für MediaTailor und MediaPackage die Bereitstellung von Inhalten](#)

Auswirkung von Abfrageparametern

Was überwacht werden sollte: Verfolgen Sie, wie sich EMP-spezifische Abfrageparameter auf die Cache-Effizienz auswirken

Optimierungsziel: Stellen Sie sicher, dass nur die erforderlichen EMP-Abfrageparameter in den Cache-Schlüsseln enthalten sind

CDN-Antwortzeiten

Was Sie überwachen sollten: Verfolgen Sie die Antwortzeiten für verschiedene Inhaltstypen (Manifeste und Segmente).

Zielwerte:

- Inhalt im Cache: weniger als 100 ms
- Ursprüngliche Anfragen: weniger als 500 ms

MediaPackage Leistungskennzahlen

Fehlerquoten

Was Sie überwachen sollten: Überwachen Sie die HTTP-Fehlerraten sowohl von Ihrem CDN als auch von Ihren MediaPackage Endpunkten aus. Achten Sie besonders auf 4xx-Fehler, die auf Konfigurationsprobleme hinweisen können.

Die wichtigsten Fehlercodes:

- 400-Fehler: Oft im Zusammenhang mit offensichtlichen Filterproblemen
- 404-Fehler: Könnte auf Routing- oder Konfigurationsprobleme hinweisen
- 504-Fehler: Timeout-Probleme, insbesondere bei LL-HLS

Volumen und Muster anfragen

Was überwacht werden sollte: Verfolgen Sie die Anforderungsmuster bis zu den MediaPackage Endpunkten, um Nutzungstrends und Kapazitätsanforderungen zu ermitteln.

Zu beobachtende Muster:

- Spitzennutzungszeiten
- Geografische Verteilung der Anfragen
- Verteilung der Inhaltstypen (live oder auf Abruf)

Latenzmetriken

End-to-end Latenz

Was überwacht werden sollte: Überwachen Sie bei LL-HLS-Implementierungen die end-to-end Latenz von der Inhaltsaufnahme bis zur Wiedergabe durch den Betrachter. Eine hohe Latenz kann auf Probleme mit der CDN-Konfiguration hinweisen.

Zielwerte:

- LL-HLS: Latenz von weniger als 3 Sekunden glass-to-glass
- Reguläres HLS: weniger als 30 Sekunden

Zeit der Manifestgenerierung

Was überwacht werden muss: Zeit, die für die Generierung von MediaPackage Manifesten benötigt wird, insbesondere bei angewendeter Filterung.

Zielwerte: weniger als 200 ms für die Manifestgenerierung

Überwachungstools und Einrichtung

Die Einrichtung umfassender Überwachungstools ist unerlässlich, um eine optimale Leistung aufrechtzuerhalten und Probleme schnell zu erkennen, bevor sie sich auf die Zuschauer auswirken.

Ohne eine angemessene Überwachung bleiben Leistungseinbußen, Cache-Ineffizienzen oder Integrationsprobleme möglicherweise unbemerkt, bis die Zuschauer eine schlechte Wiedergabequalität feststellen. Das richtige Monitoring-Setup bietet Einblick in alle Aspekte Ihrer MediaPackage und der CDN-Integration.

Verwenden Sie diese AWS-Services und -Tools, um Ihre MediaPackage und die CDN-Integration zu überwachen:

Amazon CloudWatch

Amazon CloudWatch bietet die Grundlage für die Überwachung Ihrer MediaPackage und der CDN-Integration, indem es Metriken von beiden Diensten sammelt und speichert. Die richtige CloudWatch Konfiguration stellt sicher, dass Sie über die Daten verfügen, die Sie benötigen, um Leistungstrends zu identifizieren, Probleme zu beheben und Ihre Integration zu optimieren. Ohne CloudWatch Metriken fehlt Ihnen der Einblick in die Systemleistung und Sie können Probleme möglicherweise erst erkennen, wenn sie kritisch werden.

Richten Sie die CloudWatch Überwachung für eine umfassende Erfassung von Kennzahlen ein:

1. MediaPackage Metriken: Aktivieren Sie CloudWatch Metriken für Ihre MediaPackage Endgeräte, um das Anforderungsvolumen, die Fehlerquoten und die Antwortzeiten zu verfolgen.
2. CDN-Metriken: Konfigurieren Sie CloudWatch die Konfiguration so, dass CloudFront Metriken wie Cache-Trefferquoten, Anzahl der ursprünglichen Anfragen und Fehlerraten erfasst werden.
3. Benutzerdefinierte Metriken: Erstellen Sie benutzerdefinierte Metriken für unternehmensspezifische Messwerte KPIs wie Zuschauerinteraktion oder Beliebtheit von Inhalten.

CloudWatch Dashboards

Erstellen Sie umfassende Dashboards zur Visualisierung Ihrer Kennzahlen:

1. Übersichts-Dashboard: Allgemeine Kennzahlen, die den allgemeinen Systemzustand anzeigen
2. CDN-Leistungs-Dashboard: Detaillierte CDN-Metriken, einschließlich Cache-Leistung und geografischer Verteilung
3. MediaPackage Leistungs-Dashboard: MediaPackage -spezifische Metriken, einschließlich Anforderungsmustern und Fehlerraten
4. Latenz-Dashboard: End-to-end Latenzkennzahlen für verschiedene Inhaltstypen und Regionen

Protokollanalyse

Richten Sie die Protokollanalyse für eine detaillierte Fehlerbehebung ein:

1. CDN-Zugriffsprotokolle: Aktivieren und analysieren Sie CDN-Zugriffsprotokolle, um Anforderungsmuster zu verstehen und Probleme zu identifizieren
2. MediaPackage CloudWatch Logs: Überwachen Sie MediaPackage Logs auf Fehler und Leistungsprobleme
3. Protokollaggregation: Verwenden Sie Amazon CloudWatch Logs Insights oder Tools von Drittanbietern, um Protokollmuster zu analysieren

Richten Sie Warnmeldungen und Benachrichtigungen ein

Die Konfiguration von Warnmeldungen ist entscheidend für die proaktive Erkennung und Lösung von Problemen. Ohne angemessene Warnmeldungen bleiben Probleme möglicherweise unbemerkt, bis sie das Zuschauererlebnis erheblich beeinträchtigen oder zu Serviceunterbrechungen führen. Gut konfigurierte Benachrichtigungen helfen Ihnen dabei, Probleme zu erkennen und zu beheben, bevor sie sich auf Ihre Zuschauer auswirken, und stellen sicher, dass Ihr Team über kritische Probleme informiert wird, die sofortige Aufmerksamkeit erfordern.

Konfigurieren Sie proaktive Benachrichtigungen, um Probleme zu erkennen, bevor sie sich auf Zuschauer auswirken:

Kritische Warnmeldungen

Richten Sie sofortige Benachrichtigungen für kritische Probleme ein:

- Hohe Fehlerraten: Warnung, wenn die 4xx- oder 5xx-Fehlerraten innerhalb von 5 Minuten 5% überschreiten
- Senkung der Cache-Trefferquote: Warnmeldung, wenn die Cache-Trefferquote bei Manifesten unter 70% oder bei Segmenten unter 85% fällt
- Hohe Latenz: Warnung, wenn die end-to-end Latenz die Zielschwellenwerte überschreitet
- Spitzenwerte bei den ursprünglichen Anfragen: Warnung, wenn die Anzahl der ursprünglichen Anfragen im Vergleich zum Ausgangswert um mehr als 50% steigt

Warnmeldungen

Richten Sie Warnmeldungen für Trends ein, die auf sich entwickelnde Probleme hinweisen könnten:

- Schrittweiser Leistungsabfall: Warnung, wenn sich die Reaktionszeiten innerhalb von 30 Minuten um 20% erhöhen
- Trends bei der Cache-Effizienz: Warnmeldung, wenn die Cache-Trefferquoten im Laufe der Zeit rückläufig sind
- Ungewöhnliche Verkehrsmuster: Warnung vor unerwarteten Änderungen des Anforderungsvolumens oder der geografischen Verteilung

Verwenden Sie Überwachungsdaten zur Optimierung

Nutzen Sie Überwachungsdaten, um die Leistung kontinuierlich zu verbessern:

Regelmäßige Leistungsüberprüfungen

1. Wöchentliche Bewertungen: Analysieren Sie wöchentliche Leistungstrends und identifizieren Sie Optimierungsmöglichkeiten
2. Monatliche Kapazitätsplanung: Verwenden Sie Verkehrsmuster, um den Kapazitätsbedarf und die CDN-Optimierung zu planen
3. Vierteljährliche Architekturprüfungen: Bewerten Sie die Gesamteffizienz der Architektur und identifizieren Sie Bereiche mit Verbesserungspotenzial

Allgemeine Optimierungsmaßnahmen

Ziehen Sie auf der Grundlage von Überwachungsdaten die folgenden Optimierungsmaßnahmen in Betracht:

- Anpassungen der Cache-Richtlinien: Ändern Sie die TTL-Werte auf der Grundlage der tatsächlichen Muster für die Aktualisierung von Inhalten. Eine ausführliche Anleitung zur TTL-Optimierung finden Sie unter [Caching-Optimierung für CDN und Integrationen MediaTailor](#)
- Geografische Optimierung: Fügen Sie CDN-Edge-Standorte in Regionen mit hohem Verkehrsaufkommen hinzu
- Optimierung der Abfrageparameter: Entfernen Sie unnötige Abfrageparameter, die den Cache fragmentieren
- Origin Shield-Konfiguration: Implementieren Sie Origin Shield in Regionen mit hohem Origin-Anfragevolumen

Eine ausführliche Anleitung zur Überwachung speziell für finden Sie unter [Überwachung MediaPackage](#) im MediaPackage Benutzerhandbuch. MediaPackage

Fehlerbehebung MediaPackage, CDN und Integrationen MediaTailor

AWS Elemental MediaTailor Bei der Integration mit einem AWS Elemental MediaPackage Content Delivery Network (CDN) können häufig auftretende Probleme auftreten, die sich auf die Wiedergabe, das Caching oder andere Integrationsfunktionen auswirken. Verwenden Sie dieses Handbuch, wenn Sie auf Wiedergabeprobleme, Probleme beim Zwischenspeichern oder andere Fehler im Zusammenhang mit der Integration stoßen.

Umfassende Anleitungen zur CDN-Fehlerbehebung, einschließlich universeller Cache-Leistungsprobleme, HTTP-Fehlerbehebung, Testverfahren und Diagnosetechniken, die für alle MediaTailor Implementierungen gelten, finden Sie unter. [Beheben Sie Probleme mit der CDN-Integration](#) Dieser Abschnitt konzentriert sich auf MediaPackage spezifische Anforderungen zur Problembehandlung.

Stellen Sie vor der Fehlerbehebung sicher, dass Sie die grundlegende Integrationseinrichtung korrekt abgeschlossen haben. Wenn Sie Ihre Integration noch nicht eingerichtet haben oder die Einrichtungsschritte überprüfen müssen, finden Sie weitere Informationen unter [Integrieren MediaTailor mit MediaPackage und CDN](#). Hinweise zur Optimierung der Cache-Leistung nach der Behebung von Problemen finden Sie unter [CDN-Caching](#).

Offensichtliche Filterfehler

Probleme mit der Funktion zur MediaPackage Manifestfilterung, basierend auf dokumentierten Fehlerbedingungen:

HTTP 400-Fehler bei der Manifestfilterung

Symptome: Anfragen mit `aws.manifestfilter` Parametern geben HTTP 400 Bad Request zurück

Bestätigte Ursachen (aus der AWS Dokumentation):

- Filterkriterien führen zu einem leeren Manifest (keine Streams entsprechen den Filterbedingungen)
- Ungültige Namen oder Werte für Filterparameter
- Fehlerhafte Syntax der Abfragezeichenfolge

- Doppelte oder wiederholte Filterparameter
- Die Zeichenfolge für den Filterparameter umfasst mehr als 1024 Zeichen
- Auf Medien-Playlisten oder -segmente angewendete Abfrageparameter (nicht unterstützt)

Lösungen:

1. Überprüfen Sie Ihre Filterparameter, um sicherzustellen, dass sie mit den verfügbaren Inhaltsströmen übereinstimmen. Wenn das Filtern zu keinen passenden Streams führt, wird HTTP 400 MediaPackage zurückgegeben.
2. Überprüft die Filtersyntax anhand der unterstützten Parameternamen und Wertformate.
3. Suchen Sie nach doppelten Parametern in Ihrer Abfragezeichenfolge.
4. Stellen Sie sicher, dass Filterparameter nur auf multivariante Playlisten angewendet werden, nicht auf Medien-Playlisten oder -segmente.
5. Stellen Sie sicher, dass Ihre gesamte Parameterzeichenfolge weniger als 1024 Zeichen enthält.

Referenz: [Bedingungen für AWS Elemental MediaPackage manifeste Filterfehler](#)

Die Manifestfilterung funktioniert nicht (HTTP 200, aber es wurde keine Filterung angewendet)

Symptome: Anfragen geben HTTP 200 zurück, aber das Manifest enthält alle Streams statt der gefilterten Teilmenge

Mögliche Ursachen:

- CDN leitet den `aws.manifestfilter` Abfrageparameter nicht weiter an MediaPackage
- Der Filterparameter wurde in den verfügbaren Streams nicht gefunden (gibt ein ungefiltertes Manifest mit HTTP 200 zurück)

Lösungen:

1. Stellen Sie sicher, dass Ihre CDN-Cache-Richtlinie `aws.manifestfilter` in der Liste der weitergeleiteten Abfragezeichenfolgen aufgeführt ist.
2. Testen Sie die Filterparameter direkt an MediaPackage Endpunkten (unter Umgehung des CDN), um sicherzustellen, dass sie erwartungsgemäß funktionieren.
3. Vergewissern Sie sich, dass die Filterwerte den tatsächlichen Eigenschaften Ihrer Inhaltsstreams entsprechen.

Referenz: [Bedingungen für AWS Elemental MediaPackage offensichtliche Filterfehler](#)

Diagnoseverfahren

Systematische Diagnoseverfahren helfen Ihnen, die Ursache von Integrationsproblemen schnell und effizient zu identifizieren. Ein strukturierter Ansatz verhindert Zeitverschwendung aufgrund falscher Annahmen und stellt sicher, dass Sie das eigentliche Problem und nicht die Symptome angehen. Diese evidenzbasierten Diagnoseschritte dienen dazu, Probleme zu isolieren und Sie zur geeigneten Lösung zu führen.

Folgen Sie diesen evidenzbasierten Diagnoseschritten, um Probleme zu identifizieren:

Analysieren Sie die Cache-Leistung

Die Cache-Leistungsanalyse ist für EMP-Integrationen von entscheidender Bedeutung, da eine schlechte Cache-Effizienz zu einer erhöhten Ausgangslast, höheren Kosten und potenziellen Wiedergabeproblemen führt.

Eine umfassende Problembehandlung bei der Cache-Leistung, einschließlich Analyse der Cache-Trefferquote, Optimierung der Cache-Schlüssel und systematische Diagnoseschritte, finden Sie [Leistungsprobleme beim CDN-Cache](#) im Hauptleitfaden zur CDN-Fehlerbehebung.

EMP-spezifische Überlegungen zum Cache:

- EMP-Cache-Control-Header: Stellen Sie sicher, dass Ihr CDN die Cache-Control-Header von EMP berücksichtigt, anstatt sie zu überschreiben
- EMP-Abfrageparameter: Stellen Sie sicher, dass nur die erforderlichen EMP-Abfrageparameter in den Cache-Schlüsseln enthalten sind
- EMP-TTL-Verhalten: Stellen Sie sicher, dass verschiedene EMP-Inhaltstypen über eine angemessene Cashedauer verfügen

Ausführliche Hinweise zur Optimierung von EMP-Cache-Richtlinien und TTL-Einstellungen finden Sie unter. [Optimieren Sie das CDN-Caching für MediaTailor und MediaPackage die Bereitstellung von Inhalten](#)

Überprüfen Sie die Konfiguration der Manifestfilterung

Die Überprüfung der Manifestfilterung ist unerlässlich, da Filterprobleme dazu führen können, dass Zuschauer falsche Inhalte, nicht unterstützte Formate oder Inhalte erhalten, auf die sie keinen Zugriff haben sollten. Mithilfe systematischer Tests kann festgestellt werden, ob Probleme mit der CDN-Konfiguration, der Syntax der Filterparameter oder der Verfügbarkeit von Inhalten zusammenhängen.

Testen Sie die Funktionalität der Manifestfilterung systematisch:

1. Testen Sie die Filterparameter direkt an MediaPackage Endpunkten (unter Umgehung des CDN), um sicherzustellen, dass sie ordnungsgemäß funktionieren.
2. Vergleichen Sie gefilterte und ungefilterte Manifeste, um sicherzustellen, dass erwartete Streams enthalten bzw. ausgeschlossen sind.
3. Stellen Sie sicher, dass Ihre CDN-Cache-Richtlinie den Abfrageparameter weiterleitet.
`aws.manifestfilter`
4. Suchen Sie nach HTTP 400-Fehlern und vergleichen Sie diese mit den dokumentierten Fehlerbedingungen.

Wenn Sie nach der Behebung von Problemen die Manifestfilterung implementieren oder ändern müssen, finden Sie eine vollständige Anleitung [Richten Sie die Manifestfilterung mit MediaTailor MediaPackage, und CDN ein](#) zur Einrichtung unter.

Überprüfen Sie die Konfiguration der Abfrageparameter

Stellen Sie sicher, dass Ihr CDN nur die erforderlichen Abfrageparameter weiterleitet:

1. Überprüfen Sie Ihre CDN-Cache-Richtlinie, um sicherzustellen, dass sie nur AWS empfohlene Parameter enthält:
 - `aws.manifestfilter`- für die Manifestfilterung
 - `aws.manifestsettings`- für zeitversetztes Betrachten
 - `_HLS_msnund_HLS_part` - für LL-HLS-Unterstützung
2. Entfernen Sie alle anderen Abfrageparameter aus Ihrem Cache-Schlüssel, da sie MediaPackage ignoriert werden und sie die Cache-Effizienz verringern.

Referenz: [Arbeiten mit und AWS Elemental MediaPackage CDNs](#)

Fehlercodereferenz

Referenz für dokumentierte Fehlerbedingungen und deren Ursachen:

Ungültige HTTP 400-Anfrage (Manifestfilterung)

Dokumentierte Ursachen:

- Die Anwendung des Filters führt zu einem leeren Manifest

- Ungültige Parameternamen oder -werte
- Fehlerhafte Syntax der Abfragezeichenfolge
- Doppelte Filterparameter
- Die Parameterzeichenfolge umfasst mehr als 1024 Zeichen
- Fragen Sie Parameter für Medien-Playlisten oder -segmente ab

Referenz: [Bedingungen für MediaPackage manifeste Filterfehler](#)

HTTP 200 OK (keine Filterung angewendet)

Dokumentierte Ursachen:

- Der Filterparameter wurde in den verfügbaren Streams nicht gefunden (gibt ein ungefiltertes Manifest zurück)
- Nach dem Filtern sind nur Untertitel-Streams vorhanden (gibt ein ungefiltertes Manifest zurück)

Referenz: Fehlerbedingungen bei der [MediaPackage Manifestfilterung](#)

Weitere Ressourcen zur Fehlerbehebung

Informationen zu Themen, die in diesem Thema nicht behandelt werden, finden Sie in den folgenden offiziellen AWS Ressourcen:

- [Vorschau eines Manifests von AWS Elemental MediaPackage— Verwenden Sie die Manifestvorschau](#), um Probleme beim Paketieren von Inhalten zu beheben
- [Erhöhen Sie die CloudFront Cache-Trefferquote](#) — Umfassender Leitfaden zur CDN-Cache-Optimierung
- [Manifestfilterung](#) — Vollständiger Leitfaden zur MediaPackage Filterfunktion

Integration AWS Elemental MediaTailor mit Amazon CloudFront

AWS Elemental MediaTailor lässt sich in Amazon integrieren CloudFront , um die Leistung und Zuverlässigkeit der Inhaltsbereitstellung zu verbessern. CloudFront ist ein Content Delivery Network (CDN), das Ihre Inhalte über ein weltweites Netzwerk von Rechenzentren verteilt, die als Edge-Standorte bezeichnet werden. Wenn Zuschauer deine Inhalte anfordern MediaTailor, CloudFront leitet es Anfragen an den nächstgelegenen Edge-Standort weiter. Dieser Ansatz reduziert die Latenz und verbessert die Leistung für Ihre Zuschauer.

Die Integration MediaTailor mit CloudFront bietet mehrere wichtige Vorteile:

- Geringere Latenz für Zuschauer, die auf personalisierte Inhalte zugreifen
- Verbesserte Skalierbarkeit für den Umgang mit großen Zielgruppen
- Verbesserte Zuverlässigkeit durch redundante Lieferwege
- Kostenoptimierung durch effiziente Caching-Strategien
- Erweiterte Funktionen wie regionsübergreifendes Failover mit Media Quality-Aware Resiliency (MQAR)

[Umfassende Informationen zu den CloudFront Funktionen finden Sie im Entwicklerhandbuch.](#)
[CloudFront](#) Informationen zur CloudFront Preisgestaltung finden Sie unter [CloudFront Preise](#).

Der Kürze halber verwenden wir manchmal „Manifeste“, um uns zusammenfassend auf multivariante Playlisten, Medien-Playlisten und zu beziehen. MPDs

Themen

- [Richten Sie die grundlegende CloudFront Integration ein mit MediaTailor](#)
- [Optimierung der MediaTailor Leistung mit CloudFront Funktionen](#)
- [Implementieren Sie Resilienz für mehrere Regionen MediaTailor mit MQAR](#)
- [Überwachen Sie Ihre und Ihre MediaTailor Integration CloudFront und beheben Sie Fehler](#)

Richten Sie die grundlegende CloudFront Integration ein mit MediaTailor

AWS Elemental MediaTailor Die Integration mit Amazon CloudFront verbessert die Leistung der Inhaltsbereitstellung für Ihre Zuschauer. Dieses Thema führt Sie durch die Einrichtung einer grundlegenden CloudFront Distribution für MediaTailor. Mit dieser Integration können Ihre Zuschauer über das CloudFront Netzwerk auf personalisierte Inhalte zugreifen. Sie erfahren auch, wie Sie das richtige Caching für verschiedene Inhaltstypen konfigurieren.

Hinweise zur Weitergabe von Abfrageparametern CloudFront für Autorisierung und Routing finden Sie unter [MediaTailor Manifest-Abfrageparameter](#). Informationen zu erweitertem Routing mit dynamischen Variablen finden Sie unter [MediaTailor Domain-Variablen für mehrere Inhaltsquellen](#).

Voraussetzungen

Stellen Sie vor der Konfiguration CloudFront mit sicher MediaTailor, dass Sie über Folgendes verfügen:

- Ein aktives AWS Konto mit Berechtigungen zum Erstellen und Verwalten von CloudFront Distributionen
- Eine konfigurierte MediaTailor Wiedergabekonfiguration (siehe [AWS Elemental MediaTailor Zum Einfügen von Anzeigen verwenden](#))
- Ihr Content-Ursprungsserver wurde ordnungsgemäß eingerichtet und ist zugänglich
- Grundlegendes Verständnis von Videostreaming-Konzepten (HLS/DASH)

Verteilung konfigurieren CloudFront

Gehen Sie wie folgt vor, um eine CloudFront Distribution zu erstellen und zu konfigurieren für MediaTailor:

Um eine CloudFront Distribution zu erstellen für MediaTailor

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die CloudFront Konsole unter <https://console.aws.amazon.com/cloudfront/v3/home>.
2. Wählen Sie Create Distribution (Distribution erstellen).
3. Geben Sie für Origin-Domain Ihre MediaTailor Endpunkt-URL ein (z. B. `a1b2c3d4.mediatailor.us-west-2.amazonaws.com`).
4. Wählen Sie für Protokoll die Option Nur HTTPS aus.
5. Geben Sie unter Name einen Namen ein, anhand dessen Sie diesen Ursprung identifizieren können (z. B. `mediatailor-origin`).
6. Konfigurieren Sie die Standardeinstellungen für das Cache-Verhalten:
 - a. Verwenden Sie für das Pfadmuster den Standardwert (*).
 - b. Wählen Sie für Objekte automatisch komprimieren die Option Ja aus.
 - c. Wählen Sie für Viewer-Protokollrichtlinie die Option HTTP zu HTTPS umleiten aus.
 - d. Wählen Sie für Zulässige HTTP-Methoden die Optionen GET, HEAD aus.
 - e. Wählen Sie für Cache-Richtlinie die Option aus CachingDisabled.
 - f. Wählen Sie für Origin-Anforderungsrichtlinie aus AllViewer, dass alle Header für das Standardverhalten weitergeleitet werden sollen.

 Note

Das Standardverhalten dient AllViewer dazu, Inhalte, die nicht bestimmten Pfadmustern entsprechen, sicher zu handhaben. Spezifische Cache-Verhaltensweisen für Manifeste und Segmente werden separat mit entsprechenden Richtlinien konfiguriert.

7. Konfigurieren Sie die Verteilungseinstellungen:

- a. Wählen Sie unter Preisklasse die Option aus, die am besten zu den Standorten Ihrer Zielgruppe passt.
- b. Wählen Sie für AWS WAF Web-ACL eine bestehende Web-ACL aus oder lassen Sie die Einstellung Sicherheitsvorkehrungen nicht aktivieren stehen.
- c. Lassen Sie das Feld Standard-Stammobjekt leer.
- d. Wählen Sie für Standardprotokollierung die Option On aus, um die Protokollierung zu aktivieren.

8. Wählen Sie Create Distribution (Distribution erstellen).

Cache-Verhalten konfigurieren

Nachdem Sie Ihre Distribution erstellt haben, müssen Sie zusätzliche Cache-Verhaltensweisen konfigurieren, um unterschiedliche Inhaltstypen angemessen behandeln zu können. In diesem Abschnitt wird die grundlegende Einrichtung des Cache-Verhaltens für beschrieben CloudFront.

Eine umfassende Caching-Optimierung, einschließlich erweiterter TTL-Einstellungen, Cache-Schlüsselkonfigurationen und Leistungsoptimierung, finden Sie [Caching-Optimierung für CDN und Integrationen MediaTailor](#) im CDN-Optimierungsleitfaden.

Konfiguration des Manifest-Cache-Verhaltens

Manifeste sollten nicht zwischengespeichert werden, da sie personalisierte Inhalte enthalten. Gehen Sie wie folgt vor, um das Cache-Verhalten zu konfigurieren:

So konfigurieren Sie das Verhalten des Manifest-Cache

1. Wählen Sie in der CloudFront Konsole Ihre Distribution aus.
2. Wählen Sie die Registerkarte Behaviors aus.

3. Wählen Sie Create behavior (Verhalten erstellen) aus.
4. Geben Sie im Feld Pfadmuster die Option `*.m3u8` zum Abgleichen von Multivarianten- und Medien-Playlisten von HLS ein.
5. Wählen Sie für Origin Ihren Ursprung aus. MediaTailor
6. Wählen Sie als Cache-Richtlinie die Option aus CachingDisabled.
7. Wählen Sie für Origin-Anforderungsrichtlinie aus AllViewer, dass alle erforderlichen Header für dynamische Inhalte weitergeleitet werden sollen.
8. Wählen Sie Erstellen aus.
9. Wiederhole diese Schritte für DASH-Manifeste und verwende dabei das passende MPDs Pfadmuster `*.mpd`.

Diese Konfiguration stellt sicher, dass jeder Zuschauer ein personalisiertes Manifest mit seinen spezifischen Anzeigeninhalten erhält. Das CDN speichert diese Manifeste nicht im Cache, sodass jede Anfrage direkt an MediaTailor gesendet wird.

Konfiguration des Segment-Cache-Verhaltens

Konfigurieren Sie separate Cache-Verhaltensweisen für Anzeigensegmente und Inhaltssegmente, um die Leistung zu optimieren und eine korrekte CORS-Handhabung sicherzustellen.

Konfiguration des Cache-Verhaltens von Anzeigensegmenten

Anzeigensegmente, die über das `/tm/*` Pfadmuster bereitgestellt werden, erfordern eine spezielle Konfiguration, um CORS ordnungsgemäß verarbeiten zu können. Dazu gehen Sie wie folgt vor:

Um das Cache-Verhalten von Anzeigensegmenten zu konfigurieren

1. Wählen Sie in der CloudFront Konsole Ihre Distribution aus.
2. Wählen Sie die Registerkarte Behaviors aus.
3. Wählen Sie Create behavior (Verhalten erstellen) aus.
4. Geben Sie im Feld Pfadmuster ein, `/tm/*` um den Anzeigensegmenten zu entsprechen, die von ausgeliefert werden MediaTailor.
5. Wählen Sie unter Origin den Ursprung Ihres MediaTailor Segments aus (unter Verwendung des `segments.mediatailor.region.amazonaws.com` Hostnamens).
6. Wählen Sie als Cache-Richtlinie die Option aus CachingOptimized.

7. Wählen Sie für Origin-Anforderungsrichtlinie die Option Keine aus.
8. Wählen Sie als Richtlinie für Antwort-Header die Option CORS- with-preflight-and — aus, SecurityHeadersPolicy um sicherzustellen, dass die richtigen CORS-Header in den Antworten enthalten sind.
9. Wählen Sie Erstellen aus.

Konfiguration des Cache-Verhaltens von Inhaltssegmenten

Inhaltssegmente können standardmäßige Caching-Richtlinien verwenden, um eine optimale Leistung zu erzielen. Konfigurieren Sie separate Verhaltensweisen für verschiedene Segmentformate:

So konfigurieren Sie das Verhalten des Inhaltssegment-Caches

1. Wählen Sie in der CloudFront Konsole Ihre Distribution aus.
2. Wählen Sie die Registerkarte Behaviors aus.
3. Wählen Sie Create behavior (Verhalten erstellen) aus.
4. Geben Sie im Feld Pfadmuster den *.ts Wert to match HLS-Inhaltssegmente ein.
5. Wählen Sie für Origin Ihren Inhaltsursprung aus.
6. Wählen Sie als Cache-Richtlinie die Option aus CachingOptimized.
7. Wählen Sie für Origin-Anforderungsrichtlinie die Option Keine aus.
8. Wählen Sie als Richtlinie für Antwortheader die Option CORS- with-preflight-and — aus, SecurityHeadersPolicy um eine konsistente CORS-Behandlung für alle Inhaltstypen sicherzustellen.
9. Wählen Sie Erstellen aus.
10. Wiederholen Sie diese Schritte für andere Inhaltssegmentformate mit entsprechenden Pfadmustern:
 - *.mp4 für MP4 Segmente
 - *.m4s für DASH-Segmente
 - *.cmfv und *.cmfa für CMAF-Segmente

Diese Konfiguration stellt sicher, dass Anzeigensegmente und Inhaltssegmente ordnungsgemäß zwischengespeichert werden und die CORS-Verarbeitung ordnungsgemäß erfolgt.

Anzeigensegmente verwenden den Ursprung der MediaTailor Segmente mit CORS-Schutz, während Inhaltssegmente Ihren Inhaltsursprung mit optimierten Caching-Richtlinien verwenden.

Konfiguration wird aktualisiert MediaTailor

Nachdem Sie Ihre CloudFront Distribution eingerichtet haben, aktualisieren Sie Ihre MediaTailor Konfiguration, um die CloudFront Domain zu verwenden:

Um Ihre MediaTailor Konfiguration zu aktualisieren

1. Öffnen Sie die [MediaTailor-Konsole](#).
2. Wählen Sie die Konfiguration aus, die Sie aktualisieren möchten.
3. Geben Sie im Abschnitt CDN-Konfiguration Ihren CloudFront Distributionsdomännennamen (z. B. `d1234abcdef.cloudfront.net`) in das Feld CDN-Inhaltssegment-Präfix ein.
4. Speichern Sie Ihre Änderungen.

Bei dieser Konfiguration werden Manifeste mit URLs diesem Verweis auf Ihre CloudFront Distribution MediaTailor generiert, anstatt direkt auf den Ursprung.

Testen Sie Ihre Integration

Nachdem Sie Ihre CloudFront Distribution konfiguriert und Ihre MediaTailor Konfiguration aktualisiert haben, testen Sie die Integration:

Um Ihre MediaTailor Landintegration CloudFront zu testen

1. Fordern Sie über Ihre CloudFront Distribution ein Manifest an (z. B. `https://d1234abcdef.cloudfront.net/v1/master/12345/my-config/index.m3u8`).
2. Stellen Sie sicher, dass das Manifest Segmente enthält URLs , die auf Ihre CloudFront Domain verweisen.
3. Spielen Sie den Inhalt über einen Videoplayer ab und stellen Sie sicher, dass sowohl Inhalt als auch Werbung korrekt wiedergegeben werden.
4. Überprüfen Sie die CloudFront Protokolle, um sicherzustellen, dass Anfragen korrekt weitergeleitet werden.

Beispielkonfiguration

Hier ist ein Beispiel für eine CloudFront Verteilungskonfiguration MediaTailor mit korrektem Cache-Verhalten:

Example CloudFront Beispiel für eine Verteilungskonfiguration

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

Dieses Beispiel zeigt:

- Standardverhalten: Verwendet alle InhalteAllViewer, die nicht bestimmten Pfadmustern entsprechen, CachingDisabled und um sie sicher zu handhaben
- Manifestes Verhalten (*.m3u8): Verwendet CachingDisabled und für dynamische Inhalte AllViewer

- Segmentverhalten (*.ts): Verwendet und CachingOptimized CORS-with-preflight-and-SecurityHeadersPolicy

Nächste Schritte

Nachdem Sie Ihre CloudFront Basisintegration mit eingerichtet haben MediaTailor, sollten Sie die folgenden nächsten Schritte in Betracht ziehen:

- Optimieren Sie die Leistung mit zusätzlichen CloudFront Funktionen (siehe [Optimierung der MediaTailor Leistung mit CloudFront Funktionen](#))
- Implementieren Sie Resilienz in mehreren Regionen mit MQAR (siehe) [Implementieren Sie Resilienz für mehrere Regionen MediaTailor mit MQAR](#)
- Richten Sie die Überwachung und Fehlerbehebung ein (siehe) [Überwachen Sie Ihre und Ihre MediaTailor Integration CloudFront und beheben Sie Fehler](#)

Optimierung der MediaTailor Leistung mit CloudFront Funktionen

AWS Elemental MediaTailor Die Leistung bei Amazon CloudFront kann durch zusätzliche Funktionen verbessert werden, die über die Grundkonfiguration hinausgehen. Nachdem Sie Ihre CloudFront Basiskonfiguration mit eingerichtet haben MediaTailor, können Sie zusätzliche Funktionen implementieren, um die Leistung, Zuverlässigkeit und Anpassungsoptionen zu verbessern. Diese Optimierungen tragen zu einem besseren Seherlebnis bei. Sie reduzieren auch die Kosten und die Ausgangslast.

Informationen zur Leistungsoptimierung mithilfe von dynamischem Routing und Parameterbehandlung finden Sie unter [MediaTailor dynamische Anzeigenvariablen für ADS-Anfragen](#). Strategien zur Optimierung von Abfrageparametern finden Sie unter [MediaTailor Manifest-Abfrageparameter](#).

Reduzierung der Origin-Auslastung mit Origin Shield

Origin Shield fügt eine Caching-Ebene zwischen CloudFront Edge-Standorten und deinem Ursprungsserver hinzu. Diese Funktion ist nützlich für Live-Streaming und beliebte VOD-Inhalte. Es ist hilfreich, wenn viele Zuschauer gleichzeitig denselben Inhalt anfordern.

Durch die Konsolidierung von Anfragen von mehreren Edge-Standorten reduziert Origin Shield die Belastung MediaTailor und die Herkunft Ihrer Inhalte.

Um Origin Shield für deinen MediaTailor Origin zu aktivieren

1. Öffne die CloudFront Konsole und navigiere zu deiner Distribution.
2. Wählen Sie den Ursprung aus, der auf Ihre MediaTailor Wiedergabekonfiguration verweist.
3. Wähle unter Origin Shield Ja aus.
4. Wähle aus dem Drop-down-Menü die AWS Region aus, die deinem MediaTailor Endpunkt am nächsten liegt.
5. Speichern Sie Ihre Änderungen.

Bei stark frequentierten Ereignissen reduziert Origin Shield die Anforderungslast auf deinem Ursprungsserver erheblich und verbessert die Zuverlässigkeit. Eine ausführliche Anleitung findest du unter [Origin Shield aktivieren](#) im CloudFront Entwicklerhandbuch.

Anpassung der Inhaltsbereitstellung mit Funktionen CloudFront

CloudFront Mit Funktionen können Sie einfachen JavaScript Code am Edge ausführen, um die Anfragen und Antworten von Zuschauern zu ändern. Sie können diese Funktionen für einfache Anpassungen wie URL-Änderungen, Header-Manipulation oder Standardauthentifizierung verwenden. Bei MediaTailor Workflows helfen Funktionen bei Aufgaben, die keine komplexe Verarbeitung erfordern.

CloudFront Funktionen bieten eine einfache Möglichkeit, die Bereitstellung von Inhalten an der Peripherie individuell anzupassen. So implementieren Sie sie für Ihre MediaTailor Integration:

Um CloudFront Funktionen zu implementieren für MediaTailor

1. Navigieren Sie in der CloudFront Konsole zu Funktionen.
2. Erstellen Sie eine neue Funktion und wählen Sie den entsprechenden Zweck aus:
 - URL-Manipulation — Um multivariante Playlisten, Medien-Playlisten und MPD-Anfragen URLs zu ändern, bevor sie ankommen MediaTailor
 - Header-Manipulation — Um Anforderungsheader hinzuzufügen oder zu ändern
 - Einfache Authentifizierung — Um Token zu validieren oder Parameter abzufragen
3. Schreiben Sie Ihren JavaScript Funktionscode.
4. Testen Sie Ihre Funktion mit MediaTailor Beispielanfragen.
5. Veröffentlichen Sie die Funktion und verknüpfen Sie sie mit dem Cache-Verhalten Ihrer Distribution.

Example CloudFront Beispielfunktion für die URL-Normalisierung

```
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;
}
```

Weitere Informationen und Codebeispiele finden Sie unter [Customize at the Edge by using CloudFront Functions](#) im CloudFront Entwicklerhandbuch.

Implementierung erweiterter Anpassungen mit Lambda @Edge

Wenn Sie komplexere Verarbeitungsfunktionen benötigen, als CloudFront Functions bieten kann, verwenden Sie Lambda @Edge. Mit diesem Dienst können Sie Node.js- oder Python-Funktionen an CloudFront Edge-Standorten ausführen.

Lambda @Edge -Funktionen können komplexe Operationen wie komplexe Authentifizierung, größere Antwortänderungen oder API-Integrationen von Drittanbietern ausführen.

Für komplexere Anpassungen verwenden Sie Lambda @Edge -Funktionen mit Ihrer MediaTailor und CloudFront -Integration:

Um Lambda @Edge zu implementieren mit MediaTailor

1. Erstellen Sie eine Lambda-Funktion in der Region USA Ost (Nord-Virginia).
2. Schreiben Sie Ihren Funktionscode für einen dieser Anwendungsfälle:
 - URL-Manipulation — Um multivariante Playlisten, Medien-Playlisten und MPD-Anfragen URLs zu modifizieren, bevor sie ankommen MediaTailor
 - A/B-Tests — Um Nutzer an verschiedene Anzeigenentscheidungsserver weiterzuleiten
 - Authentifizierung anfordern — Um Authentifizierungsheader hinzuzufügen
 - Änderung des Antwort-Headers — Um CORS-Header hinzuzufügen

3. Veröffentlichen Sie eine Version Ihrer Funktion und erstellen Sie einen Funktionsalias.
4. Ordnen Sie die Funktion Ihrer CloudFront Verteilung am entsprechenden Triggerpunkt zu (Zuschaueranfrage oder Zuschauerantwort).

Note

Wenn Sie Lambda @Edge mit verwenden MediaTailor, vermeiden Sie die Verwendung von Triggern, die auf den Ursprung gerichtet sind (ursprüngliche Anfrage und ursprüngliche Antwort), wenn Sie beabsichtigen, Funktionen von Media Quality-Aware Resiliency (MQAR) zu verwenden, da diese derzeit nicht kompatibel sind.

Weitere Informationen und Codebeispiele finden [Sie unter Anpassen am Edge mithilfe von Lambda @Edge](#) im CloudFront Entwicklerhandbuch.

Zusätzliche Tipps zur Leistungsoptimierung

Ziehen Sie diese zusätzlichen Optimierungen in Betracht, um die Leistung weiter zu verbessern:

Optimieren Sie die Cache-Trefferquoten

Überwachen Sie Ihre Cache-Trefferquote anhand von CloudFront Kennzahlen und suchen Sie nach Möglichkeiten, diese zu verbessern:

- Standardisieren Sie URL-Muster, um die Konsistenz der Cache-Schlüssel zu verbessern
- Verwenden Sie Whitelisting für Abfragezeichenfolgen, um nur die erforderlichen Parameter in den Cache-Schlüssel aufzunehmen
- Erwägen Sie die Implementierung der URL-Normalisierung mit Funktionen CloudFront

Latenz reduzieren

Implementieren Sie diese Techniken, um die Latenz zu minimieren:

- Aktivieren Sie die Brotli-Komprimierung für textbasierte Antworten
- Verwenden Sie HTTP/2 oder HTTP/3 für eine verbesserte Verbindungseffizienz
- Erwägen Sie, die IPv6 Unterstützung für moderne Netzwerke zu aktivieren

Kostenoptimierung

Bringen Sie Leistung und Kosteneffizienz in Einklang:

- Verwende Origin Shield, um redundante Herkunftsanfragen zu reduzieren
- Erwägen Sie die Auswahl der Preisklasse auf der Grundlage Ihrer Zielgruppenregion
- Implementieren Sie aggressives Caching für Segmente, um den ursprünglichen Traffic zu reduzieren

Nächste Schritte

Nachdem Sie die Leistung mithilfe von CloudFront Funktionen optimiert haben, sollten Sie die folgenden nächsten Schritte in Betracht ziehen:

- Implementieren Sie Resilienz in mehreren Regionen mit MQAR (siehe) [Implementieren Sie Resilienz für mehrere Regionen MediaTailor mit MQAR](#)
- Richten Sie die Überwachung und Fehlerbehebung ein (siehe) [Überwachen Sie Ihre und Ihre MediaTailor Integration CloudFront und beheben Sie Fehler](#)

Implementieren Sie Resilienz für mehrere Regionen MediaTailor mit MQAR

AWS Elemental MediaTailor Die Widerstandsfähigkeit mehrerer Regionen wird durch Media Quality-Aware Resiliency (MQAR) verbessert, eine fortschrittliche CloudFront Amazon-Funktion, die dazu beiträgt, das bestmögliche Streaming-Erlebnis MediaTailor zu bieten. Es wählt automatisch den Ursprung mit dem höchsten Qualitätsfaktor aus, wenn Sie mehrere Herkunft in verschiedenen Regionen haben. AWS Diese Funktion ist besonders nützlich für Live-Streaming, bei denen Sie einen unterbrechungsfreien Service benötigen.

Wie funktioniert MQAR

MQAR ermöglicht CloudFront es, automatisch den Ursprung mit dem höchsten Qualitätsfaktor auszuwählen. Dadurch wird sichergestellt, dass die Zuschauer das bestmögliche Streaming-Erlebnis erhalten. Bei richtiger Konfiguration bietet MQAR folgende Vorteile:

- Automatische Auswahl der Herkunft von Produkten höchster Qualität
- Reibungsloser Failover zwischen AWS Regionen bei Ausfällen
- Verbessertes Zuschauererlebnis mit minimalen Unterbrechungen
- Qualitätsüberwachung und Anpassung in Echtzeit

Schritt 1: Überprüfen Sie die MQAR-Anforderungen

Stellen Sie vor der Implementierung von MQAR sicher, dass Ihre Infrastruktur diese Anforderungen erfüllt. MQAR vergleicht Qualitätswerte verschiedener Herkunft, sodass Sie identische Inhalte benötigen, die in mehreren AWS Regionen verfügbar sind.

- Encoder senden mithilfe epochengebundener CMAF-Ingest-Streamsets ausgerichtete Ingest-Streams an alle MediaPackage Kanäle
- Zwei identische MediaPackage Kanäle in verschiedenen Regionen mit identischen Ausgangsendpunkten AWS
- CMAF-Ingest für MediaPackage Kanäle (standardmäßig aktiviert)
- CloudFront Verteilung, die für die Unterstützung von MQAR konfiguriert ist
- MediaTailor Konfigurationen für jeden Endpunkt MediaPackage

Schritt 2: Konfigurieren Sie Ihre Encoder für MQAR

Ihre Encoder müssen konsistente, synchronisierte Ausgaben in allen Regionen erzeugen, damit MQAR effektiv funktioniert. Diese Konsistenz ermöglicht CloudFront genaue Qualitätsvergleiche zwischen den Ursprüngen.

Konfigurieren Sie Ihre MediaLive Ausgänge mit diesen Einstellungen:

- Stellen Sie sicher, dass alle Video-Frameraten in der CMAF-Ausgabegruppe konsistent sind (alle gebrochenen oder alle ganzzahligen Frameraten).
- Vermeiden Sie Übergänge zwischen fraktionalen und ganzzahligen Frameraten in Kodierungssitzungen.
- Konfigurieren Sie die Sequenznummern der Ausgabesegmente so, dass sie in Kodierungssitzungen nicht rückwärts gehen.
- Verwenden Sie in allen Regionen identische Encoder-Ausgabenamen.

Weitere Informationen zur Konfiguration MediaLive für MQAR finden Sie im Benutzerhandbuch unter [Arbeiten mit MQCS](#). MediaLive

Schritt 3: Für MQAR konfigurieren MediaPackage

Richten Sie Ihre MediaPackage Kanäle und Endpunkte mit diesen Konfigurationen ein:

Um für MQAR MediaPackage zu konfigurieren

1. Erstellen Sie identische Kanal- und Endpunktkonfigurationen in jeder AWS Region.
2. Verwenden Sie CMAF als Kanaleingangstyp.
3. Aktivieren Sie für den primären MediaPackage Ursprung die Konfiguration von Force Endpoint Error mit den folgenden Einstellungen:
 - Veraltete multivariante Playlisten, Medien-Playlisten oder MPDs
 - Unvollständige multivariante Playlist, Medien-Playlist oder MPD
 - Slate-Eingabe
4. Bei MediaPackage Backup-Ursprüngen sollten Sie diese Fehlerkonfigurationen nicht aktivieren, um die Wahrscheinlichkeit eines erfolgreichen Failovers zu maximieren.

Weitere Informationen zur Konfiguration MediaPackage für MQAR finden Sie unter [Nutzung von Medienqualitätskennzahlen AWS Elemental MediaPackage im Benutzerhandbuch](#). MediaPackage

Schritt 4: Für MQAR konfigurieren CloudFront

In der CloudFront Konfiguration aktivieren Sie MQAR und definieren, wie es zwischen Ihren Ursprüngen auswählt. Erstellen Sie eine Ursprungsgruppe mit aktivierter Option Medienqualitätsfaktor.

Um CloudFront für MQAR zu konfigurieren

1. Erstellen Sie eine CloudFront Distribution, deren Ursprünge auf Ihre MediaTailor Endpunkte verweisen.
2. Erstellen Sie eine Ursprungsgruppe, die diese Ursprünge enthält.
3. Aktivieren Sie in den Einstellungen der Ursprungsgruppe die Option Qualitätsfaktor für Medien.
4. Konfigurieren Sie die Failover-Kriterien so, dass sie die Antwortcodes 404 Not Found enthalten. Sie können optional andere 4xx/5xx-Codes hinzufügen.
5. Erstellen Sie separate Cache-Verhaltensweisen für das Pfadmuster jedes Kanals. Dadurch wird verhindert, dass Scores gemischt werden, wenn dieselbe Ursprungsgruppe mehrere Kanäle bedient.

 Note

MQAR ist nicht verfügbar, wenn Lambda @Edge -Funktionen in Triggern verwendet werden, die auf den Ursprung gerichtet sind (Origin-Anfrage und Origin-Antwort), die mit dem Cache-Verhalten Ihrer Distribution verknüpft sind.

Weitere Informationen zur Konfiguration CloudFront für MQAR finden Sie im Entwicklerhandbuch unter [Medienqualität berücksichtigende Resilienz](#). CloudFront

Schritt 5: Für MQAR konfigurieren MediaTailor

Um Ihr MQAR-Setup abzuschließen, konfigurieren Sie es MediaTailor in jeder Region so, dass es mit Ihrer Multi-Region-Architektur funktioniert. Dies gewährleistet eine konsistente Anzeigeneinblendung, unabhängig davon, welcher Origin auswählt CloudFront .

Zur Konfiguration MediaTailor für MQAR

1. Erstellen Sie in jeder Region identische MediaTailor Konfigurationen, die auf die entsprechenden MediaPackage Endpunkte verweisen.
2. Konfigurieren Sie das CDN-Inhaltssegmentpräfix so, dass es Ihre CloudFront Vertriebsdomäne verwendet.
3. Stellen Sie sicher, dass die Ad Decision Server-Konfigurationen in allen Regionen identisch sind.

Diese Konfiguration stellt sicher, dass unabhängig davon, welche Herkunft auf der Grundlage von Qualitätskennzahlen CloudFront auswählt, Anzeigen weiterhin konsistent personalisiert werden MediaTailor können.

Schritt 6: Testen Sie Ihre MQAR-Konfiguration

Nachdem Sie MQAR eingerichtet haben, testen Sie Ihre Konfiguration, um sicherzustellen, dass sie wie erwartet funktioniert:

Um Ihre MQAR-Konfiguration zu testen

1. Fordern Sie Inhalte über Ihre CloudFront Distribution an.
2. Überwachen Sie Echtzeitprotokolle, um sicherzustellen, CloudFront dass die Herkunft auf Grundlage von Qualitätskennzahlen ausgewählt wird.

3. Simulieren Sie einen Ausfall in Ihrer primären AWS Region, um das Failover-Verhalten zu testen.
4. Stellen Sie sicher, dass das Einfügen von Anzeigen während des Failovers weiterhin ordnungsgemäß funktioniert.

Nächste Schritte

Beachten Sie nach der Implementierung von MQAR die folgenden nächsten Schritte:

- Richten Sie die Überwachung und Fehlerbehebung für Ihre MQAR-Konfiguration ein (siehe [Überwachen Sie Ihre und Ihre MediaTailor Integration CloudFront und beheben Sie Fehler](#))
- Implementieren Sie die automatisierte Bereitstellung mithilfe von AWS CloudFormation (siehe [Automatisieren MediaTailor und CDN mit CloudFormation](#))

Überwachen Sie Ihre und Ihre MediaTailor Integration CloudFront und beheben Sie Fehler

AWS Elemental MediaTailor Die Integration mit Amazon CloudFront erfordert eine kontinuierliche Überwachung und Fehlerbehebung, um eine optimale Leistung aufrechtzuerhalten. Überwachen Sie nach der Implementierung Ihrer CloudFront und der MediaTailor Integration die Leistung und bereiten Sie sich auf die Behebung von Problemen vor. Das Content Delivery Network (CDN) bietet Tools, die Ihnen helfen, die Leistung Ihrer Distribution zu verstehen und potenzielle Probleme zu identifizieren.

Richten Sie die Überwachung für Ihre Integration ein

Implementieren Sie diese Überwachungsstrategien, um die Leistung Ihrer CloudFront und der MediaTailor Integration zu verfolgen:

Um die Überwachung für Ihre CloudFront und die MediaTailor Integration einzurichten

1. CloudFront Standardprotokollierung aktivieren:
 - a. Wählen Sie in der CloudFront Konsole Ihre Distribution aus.
 - b. Wählen Sie die Registerkarte Protokolle aus.
 - c. Wählen Sie unter Standardprotokolle die Option Bearbeiten aus.
 - d. Wählen Sie Ein und konfigurieren Sie einen Amazon S3 S3-Bucket für die Protokollspeicherung.
 - e. Wählen Sie Änderungen speichern aus.

2. Richten Sie CloudFront Echtzeit-Protokolle ein:

- a. Wählen Sie in der CloudFront Konsole Ihre Distribution aus.
- b. Wählen Sie die Registerkarte Protokolle aus.
- c. Wählen Sie unter Echtzeitprotokolle die Option Bearbeiten aus.
- d. Wählen Sie Ein und konfigurieren Sie einen Amazon Kinesis Data Streams- oder Amazon Data Firehose-Lieferstream.
- e. Nehmen Sie diese Felder in Ihre Protokollkonfiguration auf:
 - `time-to-first-byte`- Reaktionszeit
 - `sc-status`- HTTP-Statuscode
 - `c-ip`- IP-Adresse des Betrachters
 - `cs-uri-stem`- URI-Pfad anfordern
 - `cs-user-agent`- Benutzeragent
 - `x-edge-result-type`- Art des Ergebnisses (Treffer, Fehlschlag, Fehler)
- f. Wählen Sie Änderungen speichern aus.

3. CloudWatch Dashboards erstellen:

- a. Wählen Sie in der CloudWatch Konsole Dashboards aus.
- b. Klicken Sie auf Dashboard erstellen.
- c. Fügen Sie Widgets für diese CloudFront Metriken hinzu:
 - Anforderungen
 - BytesDownloaded
 - 4 xxErrorRate
 - 5xxErrorRate
 - TotalErrorRate
 - CacheHitRate

4. CloudWatch Alarme einrichten:

- a. Wählen Sie in der CloudWatch Konsole Alarme aus.
- b. Wählen Sie Alarm erstellen aus.
- c. Erstellen Sie Alarme für diese Bedingungen:

- 5 `xxErrorRate` > 1% für 5 Minuten
- 4 `xxErrorRate` mehr als 5% für 5 Minuten
- `CacheHitRate` weniger als 80% für 30 Minuten

Überwachung der MAR-Leistung

Wenn Sie MQAR verwenden, hilft Ihnen die Überwachung dabei zu verstehen, wie CloudFront Sie zwischen Ihren Herkunftsorten auswählen und ob die Qualitätswerte den Erwartungen entsprechen. Echtzeitprotokolle zeigen diese Entscheidungen, sobald sie getroffen werden.

Um die Leistung von MQAR zu überwachen

1. Aktivieren Sie Echtzeit-Protokolle für Ihre CloudFront Distribution.
2. Nehmen Sie diese Felder in Ihre Protokollkonfiguration auf:
 - `r-host`- Der Hostname des ausgewählten Ursprungs
 - `sr-reason`- Der Grund für die Auswahl des Ursprungs
 - `x-edge-mqcs`- Der Wert für das Vertrauen in die Medienqualität
3. Richten Sie ein Protokollziel in Amazon Kinesis Data Streams oder Amazon Data Firehose ein.
4. Erstellen Sie Dashboards oder Warnmeldungen auf der Grundlage dieser Kennzahlen, um Qualitätswerte und Entscheidungen zur Herkunftsauswahl zu überwachen.

Example CloudWatch Beispiel-Dashboard für die MQAR-Überwachung

Erstellen Sie ein CloudWatch Dashboard mit diesen Metriken:

- Die Auswahl der Herkunft zählt nach Region
- Durchschnittliche Qualitätswerte im Laufe der Zeit
- Failover-Ereignisse
- 4xx- und 5xx-Fehlerraten nach Herkunft

Weitere Informationen zum Einrichten von Echtzeitprotokollen finden Sie unter [Echtzeitprotokolle](#) im CloudFront Entwicklerhandbuch.

Behebung häufiger Probleme

Selbst bei sorgfältiger Planung können Probleme mit Ihnen CloudFront und der MediaTailor Integration auftreten. Wenn Sie häufig auftretende Probleme und Lösungen verstehen, können Sie Probleme schnell lösen und die Auswirkungen auf die Zuschauer minimieren.

Wenn Sie Probleme mit Ihrer CloudFront und der MediaTailor Integration haben, schauen Sie sich diese häufigen Probleme und Lösungen an:

Offensichtliche Probleme beim Zwischenspeichern

Symptom: Veraltete Manifeste oder Anzeigen werden nicht aktualisiert

Lösung: Stellen Sie sicher, dass Sie die `CachingDisabled` Cache-Richtlinie für multivariante Playlisten, Medien-Playlisten und MPD-Pfade verwenden. Vergewissern Sie sich, dass Ihre Konfiguration Abfrageparameter korrekt weiterleitet.

CORS-Fehler

Symptom: Die Browserkonsole zeigt beim Zugriff auf Inhalte CORS-Fehler

Lösung: Konfigurieren Sie eine Antwort-Header-Richtlinie mit entsprechenden CORS-Headern und verknüpfen Sie sie mit Ihrem Cache-Verhalten.

MQAR funktioniert nicht

Symptom: Bei der Auswahl des Ursprungs werden Qualitätswerte ignoriert

Lösung: Vergewissere dich, dass du die Option „Qualitätsfaktor für Medien“ in den Einstellungen deiner Ursprungsgruppe aktiviert hast. Stellen Sie sicher, dass Sie keine originalen Lambda `@Edge` -Trigger verwenden.

Fehler beim Einfügen von Anzeigen

Symptom: Anzeigen werden nicht richtig eingefügt

Lösung: Stellen Sie sicher, dass Ihre MediaTailor Konfiguration auf das Präfix für das Segment CloudFront Distribution for Content verweist. Vergewissern Sie sich, dass Ihr Setup alle erforderlichen Header weiterleitet.

Bei komplexeren Problemen können Sie die folgenden Lösungsansätze verwenden:

1. Überprüfen Sie die CloudFront Verteilungsprotokolle auf Fehlermuster
2. Verwenden Sie Browser-Entwicklertools, um Netzwerkanfragen zu überprüfen
3. Vergleichen Sie manifeste Inhalte von MediaTailor direkt mit Inhalten, die über bereitgestellt wurden CloudFront
4. Testen Sie mit einem einfachen Player, der detaillierte Protokollierung unterstützt

Weitere Unterstützung bei der Fehlerbehebung finden Sie im Abschnitt [Fehlerbehebung](#) im MediaTailor Benutzerhandbuch.

Arbeitsablauf zur Fehlerbehebung

Folgen Sie diesem systematischen Ansatz, um Probleme mit Ihrer CloudFront und Ihrer MediaTailor Integration zu beheben:

Zur Behebung von Problemen CloudFront und MediaTailor Integrationsproblemen

1. Identifizieren Sie das spezifische Problem:
 - a. Stellen Sie fest, ob das Problem alle Zuschauer oder nur einige betrifft
 - b. Identifizieren Sie, welche Inhaltstypen betroffen sind (Manifeste, Segmente oder beides)
 - c. Notieren Sie sich alle Fehlermeldungen oder Symptome
2. Überprüfen Sie die CloudFront Protokolle:
 - a. Suchen Sie nach Fehlerstatuscodes (4xx oder 5xx)
 - b. Überprüfen Sie die Cache-Muster hit/miss
 - c. Stellen Sie sicher, dass Anfragen an den richtigen Ursprung weitergeleitet werden
3. Konfiguration überprüfen:
 - a. Überprüfen Sie das Verhalten des Caches auf korrekte Pfadmuster
 - b. Stellen Sie sicher, dass die Cache-Richtlinien korrekt angewendet werden
 - c. Vergewissern Sie sich, dass die Richtlinien für ursprüngliche Anfragen die erforderlichen Header weiterleiten
4. Testen Sie den direkten Zugriff:
 - a. Versuchen Sie, direkt von MediaTailor (Umgehung CloudFront) auf Inhalte zuzugreifen
 - b. Vergleichen Sie die Antworten zwischen Direktzugriff und CloudFront Zugriff

5. Lösung implementieren:

- a. Wenden Sie auf der Grundlage Ihrer Ergebnisse die entsprechende Lösung an
- b. Testen Sie, ob das Problem behoben ist
- c. Dokumentieren Sie das Problem und die Lösung zum future Nachschlagen

Nächste Schritte

Nachdem Sie die Überwachung und Fehlerbehebung für Ihre CloudFront und die MediaTailor Integration eingerichtet haben, sollten Sie die folgenden nächsten Schritte in Betracht ziehen:

- Implementieren Sie die automatisierte Bereitstellung mithilfe von AWS CloudFormation (siehe [Automatisieren MediaTailor und CDN mit CloudFormation](#))
- Erstellen Sie Runbooks für gängige Betriebsszenarien und Verfahren zur Fehlerbehebung
- Richten Sie automatische Problembehebungen für häufig auftretende Probleme ein

Drittanbieter CDNs für die MediaTailor Anzeigenbereitstellung einrichten

Drittanbieter CDNs wie Akamai und Fastly können die Leistung und Skalierbarkeit Ihrer AWS Elemental MediaTailor Anzeigenbereitstellung erheblich verbessern und gleichzeitig die Bandbreitenkosten senken. Die CDN-Konfiguration für personalisierte Werbung erfordert jedoch spezielle Einstellungen, die sich von der standardmäßigen Videobereitstellung unterscheiden.

Hinweise zur Weitergabe von Abfrageparametern über Drittanbieter finden Sie CDNs unter [MediaTailor Manifest-Abfrageparameter](#). Erweiterte Routingkonfigurationen mit dynamischen Variablen finden Sie unter [MediaTailor dynamische Anzeigenvariablen für ADS-Anfragen](#).

Diese Anleitung führt Sie durch den gesamten Prozess der Einrichtung Ihres Drittanbieter-CDN, damit es optimal funktioniert. MediaTailor Sie erfahren, wie Sie zwei grundlegende Verhaltensweisen konfigurieren:

- Umgehung von Manifesten: Stellt sicher, dass jeder Zuschauer personalisierte Anzeigeneinblendungen erhält, indem das Zwischenspeichern von Manifesten verhindert wird
- Segment-Caching: Maximiert die Leistung und senkt die Kosten durch effizientes Zwischenspeichern von Videoinhalten

Der Konfigurationsprozess dauert in der Regel 30 bis 60 Minuten und umfasst die Auswahl, Einrichtung, Überprüfung und Optimierung des Anbieters. Sobald der Vorgang abgeschlossen ist, verfügen Sie über eine CDN-Konfiguration, die personalisierte Werbung effizient bereitstellt und gleichzeitig ein optimales Zuschauererlebnis gewährleistet.

 Note

Dieser Leitfaden konzentriert sich auf die Konfigurationen von Akamai und Fastly. Anweisungen zur CloudFront Einrichtung finden Sie unter. [CloudFront-Integration](#)

Voraussetzungen

Bevor Sie Ihr Drittanbieter-CDN mit einrichten, stellen Sie sicher MediaTailor, dass Sie über Folgendes verfügen:

- Eine aktive MediaTailor Konfiguration, die Ihren Inhaltsursprungs- und Anzeigenentscheidungsserver umfasst
- Zugriff auf die Konfigurationsoberfläche Ihres CDN
- Eine Liste der Dateierweiterungen, die in Ihren Inhalten verwendet werden (.m3u8, .mpd, .ts usw.)
- Die Dokumentation Ihres CDN-Anbieters als Referenz

Anleitungen zur CloudFront Einrichtung anstelle von CDNs Drittanbieteranleitungen finden Sie unter [CloudFront-Integration](#).

Terminologie

Um die CDN-Konfigurationsanforderungen zu verstehen, müssen Sie die folgenden Manifesttypen kennen:

- HLS-Manifestationen:
 - Multivariante Playlist: Das Manifest der obersten Ebene, das Links zu Medien-Playlisten enthält
 - Medien-Playlist: Das Manifest der zweiten Ebene mit Links zu Inhaltssegmenten
- DASH-Manifestationen:
 - MPD (Media Presentation Description): Die Standardbezeichnung für DASH-Manifeste

In diesem Handbuch werden alle Manifestdateien (multivariante Playlisten, Medien-Playlisten und MPDs) zusammenfassend als Manifeste bezeichnet, wenn es um allgemeine Konfigurationsanforderungen geht.

Allgemeine CDN-Konfigurationsprinzipien, die für alle Anbieter gelten, finden Sie unter. [Richten Sie die CDN-Integration ein mit MediaTailor](#)

Anleitungen zur CDN-Optimierung finden Sie unter. [Leitfaden zur Leistungsoptimierung für CDN und Integrationen MediaTailor](#)

CDN-Caching-Regeln konfigurieren

Die CDN-Caching-Konfiguration ist für die MediaTailor Anzeigenbereitstellung von entscheidender Bedeutung, da sie bestimmt, wie Ihre Inhalte die Zuschauer erreichen. Durch die richtige Konfiguration wird sichergestellt, dass die Manifeste für jeden Zuschauer personalisiert bleiben, während Segmente effizient zwischengespeichert werden, um die Belastung durch die Herkunft zu reduzieren und die Leistung zu verbessern.

Diese Konfiguration dauert in der Regel 15 bis 30 Minuten pro CDN-Anbieter und erfordert zwei unterschiedliche Verhaltensweisen:

- **Umgang mit Manifesten:** Verhindert das Zwischenspeichern, um sicherzustellen, dass jeder Zuschauer personalisierte Anzeigeneinblendungen erhält
- **Segment-Caching:** Maximiert die Cache-Effizienz für Videoinhalte, um die Auslieferungsleistung zu verbessern

Gehen Sie wie folgt vor, um die Caching-Regeln Ihres CDN für eine optimale Anzeigenbereitstellung zu konfigurieren.

Wählen Sie auf den folgenden Tabs Ihren CDN-Anbieter aus, um spezifische Anweisungen zu erhalten:

Akamai

Konfigurieren Sie diese beiden Verhaltensweisen in Ihrer Akamai-Property:

- Manifest-Behandlung zur Verhinderung von Caching
- Segment-Caching für optimale Leistung

Konfigurieren Sie die Manifestzustellung

Konfigurieren Sie Ihr Akamai-CDN so, dass Manifeste nicht zwischengespeichert werden, sodass jeder Zuschauer personalisierte Werbung erhält.

Manifestdateien enthalten die personalisierten Einfügepunkte für Anzeigen, die für jeden Zuschauer MediaTailor generiert werden. Das Zwischenspeichern dieser Dateien würde dazu führen, dass alle Zuschauer identische Anzeigen sehen, was dem Zweck personalisierter Werbung zuwiderlaufen würde.

Gehen Sie bei Manifestanfragen (Dateien, die auf .m3u8, .mpd oder .smil enden) wie folgt vor:

1. Erstellen Sie ein Verhalten, das den Manifest-Dateierweiterungen entspricht (.m3u8, .mpd, .smil)
2. Stellen Sie die Caching-Option auf Kein Speicher ein
3. Konfigurieren Sie die Cache-Schlüssel so, dass sie alle Abfrageparameter enthalten
4. Aktivieren Sie Forward Host Header für das richtige Origin-Routing
5. Konfigurieren Sie die Header-Weiterleitung für alle Header. Die Mindestanforderungen finden Sie unter [Erforderliche Header für MediaTailor die CDN-Integration](#).

Konfigurieren Sie die Segmentzustellung

Konfigurieren Sie Ihr Akamai-CDN so, dass Videosegmente zwischengespeichert werden, um die CDN-Effizienz zu maximieren und die Ausgangslast zu reduzieren.

Videosegmente sind die eigentlichen Inhaltsdateien, die sicher zwischengespeichert werden können, da sie keine personalisierten Informationen enthalten. Durch das richtige Zwischenspeichern von Segmenten werden die Bandbreitenkosten gesenkt und die Wiedergabeleistung für die Zuschauer verbessert.

Gehen Sie bei Segmentanfragen (Dateien, die auf .ts, .mp4, .m4s usw. enden) wie folgt vor:

1. Erstellen Sie ein Verhalten, das den Segmentdateierweiterungen (.ts, .mp4, .m4s) entspricht
2. Setze Honor Origin Cache Control auf Ja
3. Konfigurieren Sie Standardeinstellungen time-to-live (TTL) für den Fall, dass Origin-Header fehlen:
 - Standard-TTL: 86400 Sekunden (24 Stunden)
 - Maximale TTL: 604.800 Sekunden (7 Tage)

 Note

Nachdem Sie diese Verhaltensweisen konfiguriert haben, aktivieren Sie Ihre Eigenschaftsänderungen im Akamai Control Center.
Die Änderungen werden nach der Aktivierung wirksam.

Fastly

Erstellen Sie diese beiden Konfigurationen in Ihrem Fastly-Service:

- Manifest-Behandlung zur Verhinderung von Caching
- Segment-Caching für optimale Leistung

Konfigurieren Sie die Manifestzustellung

Konfigurieren Sie Ihr Fastly CDN so, dass das Zwischenspeichern von Manifestdateien umgangen wird, sodass jeder Zuschauer personalisierte Anzeigeninhalte erhält.

Die Manifestdateien müssen MediaTailor für jede Anfrage verfügbar sein, um eine korrekte Anzeigenpersonalisierung zu gewährleisten. Durch das Umgehen des Caches für diese Dateien wird sichergestellt, dass die individuellen Targeting-Parameter jedes Betrachters korrekt verarbeitet werden.

Gehen Sie bei Manifestanfragen wie folgt vor:

1. Erstellen Sie eine Anforderungsbedingung, um Manifestpfade zu identifizieren
2. Stellen Sie die Cache-Bedingung auf Für diese Anfragen nicht zwischenspeichern
3. Konfigurieren Sie die Forward-Einstellungen so, dass sie alle Abfrageparameter enthalten
4. Fügen Sie User-Agent es Ihrer Konfiguration für die Header-Weiterleitung hinzu

Konfigurieren Sie die Segmentzustellung

Konfigurieren Sie Ihr Fastly CDN so, dass Videosegmente zwischengespeichert werden, um die Übertragungsleistung zu verbessern und den ursprünglichen Datenverkehr zu reduzieren.

Das Zwischenspeichern von Segmenten ist für eine kostengünstige Bereitstellung und ein optimales Zuschauererlebnis unerlässlich. Diese Dateien sind für alle Zuschauer identisch und profitieren erheblich vom CDN-Caching.

Gehen Sie bei Segmentanfragen wie folgt vor:

1. Erstellen Sie eine Anforderungsbedingung, um Segmentpfade zu identifizieren
2. Stellen Sie die Cache-Einstellungen so ein, dass sie die ursprünglichen Cache-Header berücksichtigen
3. Konfigurieren Sie die Standardeinstellung time-to-live (TTL) auf 86400 Sekunden (24 Stunden) für den Fall, dass die ursprünglichen Header fehlen

 Note

Nachdem Sie diese Änderungen vorgenommen haben, aktivieren Sie eine neue Version Ihres Fastly-Dienstes.

Die Konfiguration wird nach der Aktivierung wirksam.

Überprüfen Sie Ihre CDN-Konfiguration

Durch die Überprüfung wird sichergestellt, dass Ihre CDN-Konfiguration ordnungsgemäß funktioniert, bevor Sie den Produktionsdatenverkehr über sie weiterleiten. Diese Tests bestätigen, dass die Anzeigenpersonalisierung ordnungsgemäß funktioniert und dass das Caching die erwarteten Leistungsvorteile bietet.

Die vollständige Überprüfung dauert in der Regel 10 bis 15 Minuten und sollte möglichst von mehreren geografischen Standorten aus durchgeführt werden.

Führen Sie nach der Einrichtung Ihres CDN die folgenden Prüfungen durch, um sicherzustellen, dass es ordnungsgemäß funktioniert:

1. Personalisierung von Testmanifesten:
 - a. Fordern Sie dieselbe Inhalts-URL mit unterschiedlichen Anzeigenparametern an
 - b. Stellen Sie sicher, dass bei jeder Anfrage unterschiedliche Anzeigeneinfügungen zurückgegeben werden
2. Testen Sie das Segment-Caching:
 - a. Überprüfen Sie die CDN-Metriken auf die Trefferquote im Segment-Cache (sollte über 90% liegen)

- b. Überwachen Sie den ursprünglichen Datenverkehr, um sicherzustellen, dass er niedriger ist als der Direktzustellung
3. Testen Sie die Wiedergabeleistung:
- a. Spielen Sie Inhalte über Ihr CDN von verschiedenen Orten aus ab
 - b. Stellen Sie sicher, dass die Wiedergabe reibungslos und ohne Pufferprobleme funktioniert

Umfassende Testmethoden und erweiterte Validierungsverfahren finden Sie unter. [Testen und Validieren für CDN und Integrationen MediaTailor](#)

Optimieren Sie die CDN-Leistung

Nachdem Sie Ihre Basiskonfiguration überprüft haben, implementieren Sie diese Optimierungen, um die Leistung zu maximieren und die Kosten zu minimieren:

Überwachen Sie die wichtigsten Leistungskennzahlen

Verfolgen Sie diese Kennzahlen, um eine optimale Leistung sicherzustellen:

Trefferquote im Cache

Ziel: Mehr als 90% für Videosegmente

Wirkung: Höhere Quotienten reduzieren die Auslastung der Ausgangsdaten und verbessern das Zuschauererlebnis

Überwachen: Überprüfe täglich das Analyse-Dashboard deines CDN-Anbieters

Antwortzeit von Origin

Ziel: Weniger als 200 ms für Manifestanfragen

Auswirkung: Durch die schnellere Übermittlung von Manifesten wird die Startzeit für Zuschauer reduziert

Monitor: Richten Sie Warnmeldungen für Antwortzeiten von mehr als 500 ms ein

Fehlerraten

Ziel: Weniger als 0,1% für alle Anfragen

Auswirkung: Hohe Fehlerraten deuten auf Konfigurationsprobleme oder Ursprungsprobleme hin

Überwachen: Richten Sie Warnmeldungen für Fehlerraten von mehr als 1% ein

Passen Sie das Verhalten beim Zwischenspeichern an

Passen Sie diese Einstellungen an Ihre Inhaltsmerkmale und Zuschauermuster an:

TTL-Optimierung für Segmente

Live-Inhalte: Verwenden Sie kürzere TTL (1—4 Stunden), um zeitnahe Updates sicherzustellen

VOD-Inhalte: Verwenden Sie längere TTL (24-48 Stunden), um die Cache-Effizienz zu maximieren

Anzeigensegmente: Ziehen Sie eine kürzere TTL (30 Minuten bis 2 Stunden) für häufig aktualisierte Anzeigeninhalte in Betracht

Umfassende TTL-Empfehlungen und Caching-Strategien für alle MediaTailor Workflows finden Sie unter. [Caching-Optimierung für CDN und Integrationen MediaTailor](#)

Geografische Optimierung

Herkunft mehrerer Regionen: Konfigurieren Sie die Herkunftsauswahl auf der Grundlage des Standorts des Betrachters

Randstandorte: Ermöglichen Sie zusätzliche Randstandorte in Regionen mit hoher Zuschauerkonzentration

Failover: Konfigurieren Sie die Backup-Ursprünge für hohe Verfügbarkeit

Optimieren Sie die Kosten

Implementieren Sie diese Strategien, um die CDN-Kosten zu senken und gleichzeitig die Leistung aufrechtzuerhalten:

- Komprimierung: Aktivieren Sie die Gzip-Komprimierung für Manifestdateien, um die Bandbreitennutzung zu reduzieren
- Strategie zum Löschen: Implementieren Sie selektives Löschen des Cache anstelle von vollständigen Cache-Löschungen
- Verkehrsanalyse: Überprüfen Sie monatlich die Verkehrsmuster, um Optimierungsmöglichkeiten zu identifizieren
- Tierausswahl: Verwenden Sie die geeigneten CDN-Servicestufen, die Ihren Leistungsanforderungen entsprechen

Beheben Sie CDN-Probleme von Drittanbietern

Probleme mit der CDN-Konfiguration äußern sich in der Regel entweder in Problemen mit der Anzeigenpersonalisierung oder in Leistungseinbußen. Verwenden Sie diesen systematischen Ansatz, um die häufigsten Probleme zu identifizieren und zu lösen, die sich auf die MediaTailor Anzeigenbereitstellung auswirken.

Die meisten Problembehebungen können innerhalb von 15 bis 30 Minuten abgeschlossen werden, indem die spezifischen Symptome überprüft und die entsprechenden Lösungen angewendet werden.

Wenn Zuschauer Probleme mit der Anzeigenbereitstellung oder der Wiedergabequalität haben, können Sie anhand dieses Leitfadens häufig auftretende Probleme mit der CDN-Konfiguration identifizieren und lösen:

Akamai: Manifeste im Cache

Symptom: Zuschauer sehen identische Anzeigen, auch wenn Sie unterschiedliche Targeting-Parameter konfigurieren.

Lösung: Stellen Sie sicher, dass Sie die Option No Store Caching auf Manifestpfade angewendet haben.

Stellen Sie außerdem sicher, dass Sie Abfrageparameter in den Cache-Schlüssel aufgenommen haben.

Schnell: Falsche Cache-Schlüssel

Symptom: Zuschauer erleben eine inkonsistente Anzeigenpersonalisierung.

Zuschauer sehen möglicherweise auch Anzeigen, die für andere Zuschauer bestimmt sind.

Lösung: Stellen Sie sicher, dass Sie die Forward-Einstellungen so konfiguriert haben, dass alle Abfrageparameter in den Cache-Schlüssel aufgenommen werden.

Allgemein: Hoher Ausgangsverkehr

Symptom: Auf Ihren Ursprungsservern kommt es zu unerwartet hohem Datenverkehr

Lösung: Überprüfen Sie die Einstellungen und Werte für das Segment-Caching time-to-live (TTL).

Überprüfen Sie die Cache-Trefferquoten in Ihren CDN-Metriken.

Allgemein: Wiedergabefehler

Symptom: Bei Zuschauern treten Puffer- oder Wiedergabefehler auf

Lösung: Überprüfen Sie die CDN-Routing-Regeln und den Zustand des Ursprungs.

Stellen Sie sicher, dass alle erforderlichen Header korrekt weitergeleitet werden.

 Note

Wenn diese Lösungen Ihr Problem nicht lösen, lesen Sie in der Dokumentation Ihres CDN-Anbieters nach. Sie können sich auch an das Support-Team wenden, um weitere Schritte zur Fehlerbehebung zu erhalten.

Allgemeine Hinweise zur CDN-Fehlerbehebung finden Sie unter [Probleme mit MediaTailor der CDN-Integration beheben](#).

Leitfaden zur Leistungsoptimierung für CDN und Integrationen MediaTailor

AWS Elemental MediaTailor Die Leistung kann durch systematische Optimierung des Content Delivery Network (CDN) maximiert werden. Ganz gleich, ob Sie serverseitige Anzeigeneinfügung (SSAI), Channel-Assembly oder kombinierte Workflows implementieren, die Optimierungsprinzipien und Leistungsziele bleiben einheitlich. Dieser Leitfaden bietet umfassende Optimierungstechniken und Benchmarks, die für alle Implementierungen gelten. MediaTailor

Informationen zur erweiterten Routing-Optimierung mithilfe dynamischer Variablen und Konfigurationsaliase finden Sie unter [MediaTailor dynamische Anzeigenvariablen für ADS-Anfragen](#) Strategien zur Optimierung von Abfrageparametern finden Sie unter [MediaTailor Manifest-Abfrageparameter](#).

Überblick über den Optimierungsablauf:

1. Caching konfigurieren — Legen Sie die entsprechenden TTL-Werte und das Cache-Verhalten fest
2. Routing optimieren — Konfigurieren Sie Richtlinien für das Routing von Anfragen und den Ursprung
3. Leistung messen — Vergleich etablierter Benchmarks
4. Wenden Sie fortgeschrittene Techniken an — Implementieren Sie zusätzliche Optimierungsfunktionen

Themen

- [Caching-Optimierung für CDN und Integrationen MediaTailor](#)
- [Fordern Sie eine Routing-Optimierung für CDN und Integrationen an MediaTailor](#)
- [Leistungsbenchmarks für CDN und Integrationen MediaTailor](#)
- [Fortgeschrittene Optimierungstechniken für CDN und Integrationen MediaTailor](#)

Caching-Optimierung für CDN und Integrationen MediaTailor

AWS Elemental MediaTailor Die Caching-Anforderungen variieren je nach Workflowtyp und Inhaltsformat. Die richtige Caching-Konfiguration ist entscheidend für eine optimale Leistung, Kosteneffizienz und ein optimales Zuschauererlebnis. Die folgenden Abschnitte enthalten detaillierte Anleitungen zum Caching für verschiedene MediaTailor Implementierungen.

Serverseitiges Zwischenspeichern von Werbeanzeigen (SSAI)

Bei Workflows zur serverseitigen Anzeigeneinfügung können personalisierte Manifeste nicht zwischengespeichert werden, Inhalte und Anzeigensegmente sollten jedoch aggressiv zwischengespeichert werden:

SSAI-CDN-Caching-Einstellungen

Inhaltstyp	TTL	Pfadmuster	Schlüsselemente zwischenspeichern
Multivariante Playlisten	0 Sekunden	/v1/master/ *	URL-Pfad + alle Abfrageparameter
Medien-Playlisten	0 Sekunden	/v1/manifest/ *	URL-Pfad + alle Abfrageparameter
STRICH MPDs	0 Sekunden	/v1/dash/ *	URL-Pfad + alle Abfrageparameter
Inhaltssegmente	Mehr als 24 Stunden	Inhaltsspezifische Pfade	Nur URL-Pfad
Anzeigensegmente	Mehr als 24 Stunden	/v1/segment/ *	Nur URL-Pfad

- Legen Sie für personalisierte Manifeste eine TTL von 30 Sekunden fest, um sicherzustellen, dass Zuschauer up-to-date Werbeinhalte erhalten
- Konfigurieren Sie längere TTL-Werte für Inhalte und Anzeigensegmente, um die Cache-Effizienz zu maximieren
- Richten Sie Cache-Verhalten ein, das Personalisierungsparameter im Cache-Schlüssel enthält, wenn Sie gezielte Werbung unterstützen
- Implementieren Sie das Zusammenklappen von Anfragen auf CDN-Ebene, um gleichzeitige Anfragen effizient zu verarbeiten

Empfohlene TTL-Konfigurationseinstellungen

Für eine optimale SSAI-Leistung konfigurieren Sie Ihre CDN-Cache-Richtlinien mit diesen spezifischen TTL-Einstellungen:

SSAI TTL-Konfigurationseinstellungen

Inhaltstyp	TTL-Einstellung	Empfohlener Wert
Anzeigensegmente	Minimale TTL	1 Sekunde
Segmente hinzufügen	Max. TTL	86400 Sekunden (24 Stunden)
Anzeigensegmente	Standard-TTL	86400 Sekunden (24 Stunden)
Inhaltssegmente	Minimale TTL	1 Sekunde
Inhaltssegmente	Max. TTL	86400 Sekunden (24 Stunden)
Inhaltssegmente	Standard-TTL	86400 Sekunden (24 Stunden)

Diese Einstellungen stellen sicher:

- Mindest-TTL von 1 Sekunde: Ermöglicht bei Bedarf eine schnelle Cache-Invalidierung und verhindert gleichzeitig übermäßige Anfragen vom Ursprung
- Max. TTL von 24 Stunden: Bringt die Cache-Effizienz mit den Anforderungen an die Aktualität der Inhalte in Einklang
- Standard-TTL von 24 Stunden: Bietet optimales Caching für Segmente, die keine expliziten Cache-Control-Header haben

Servergesteuertes Zwischenspeichern der Anzeigeneinfügung (SGAI)

Servergeführte Anzeigeneinfügung (SGAI) ermöglicht effizientes CDN-Caching durch zwischenspeicherbare Medienmanifeste, die vorhersehbare URL-Muster verwenden. Dieser Abschnitt konzentriert sich auf CDN-spezifische Konfigurationsanforderungen für eine optimale SGAI-Leistung.

CDN-Caching-Konfiguration für SGAI

Konfigurieren Sie Ihr CDN mit diesen SGAI-spezifischen Caching-Verhaltensweisen:

SGAI CDN-Caching-Einstellungen

Inhaltstyp	TTL	Pfadmuster	Schlüsselemente zwischenspeichern
Multivariante SGAI-Playlisten (nicht zwischenspeichern)	0 Sekunden (nicht zwischenspeichern)	/v1/master/ *	URL-Pfad + ausgewählte Abfrageparameter
SGAI-Medien-Playlisten	1-4 Sekunden (halbe Segmentlänge)	/v1/i-media/ *	URL-Pfad + ausgewählte Abfrageparameter
Antworten auf die Asset-Liste (nicht zwischenspeichern)	0 Sekunden (nicht zwischenspeichern)	/v1/interstitials/ *	URL-Pfad + alle Abfrageparameter
Anzeigensegmente	Mehr als 24 Stunden	Anzeigenspezifische Pfade	Nur URL-Pfad

Konfiguration des Cache-Verhaltens

Richten Sie spezielle Cache-Verhaltensweisen für SGAI-Inhalte ein:

- **SGAI-Manifestverhalten** — Erstellen Sie ein Cache-Verhalten für `/v1/i-media/*` Pfade mit einer TTL von 1—4 Sekunden
- **Verhalten bei der Asset-Liste** — Erstellen Sie ein Cache-Verhalten für `/v1/interstitials/*` Pfade mit einer TTL von 0 Sekunden
- **Behandlung von Abfrageparametern** — Nehmen Sie nur wichtige Targeting-Parameter in die Cache-Schlüssel auf, um die Cache-Effizienz zu maximieren
- **Header der ursprünglichen Anfrage** — Leitet die für die Anzeigenausrichtung erforderlichen Header weiter und behält dabei die Cache-Fähigkeit bei

Zwischenspeichern von Kanal-Assemblys

Bei Workflows zur Kanalmontage können Manifeste für kurze Zeit zwischengespeichert werden, während Segmente aggressiv zwischengespeichert werden sollten:

CDN-Caching-Einstellungen für die Kanalmontage

Inhaltstyp	VOD TTL	Live-TTL	Pfadmuster	Schlüsselemente zwischenspeichern
Multivariante Playlisten	5-30 Minuten	5-10 Sekunden	Kanalspezifische Pfade	URL-Pfad + alle Abfrageparameter
Medien-Playlisten	5-30 Minuten	2-5 Sekunden	Kanalspezifische Pfade	URL-Pfad + alle Abfrageparameter
STRICH MPDs	5-30 Minuten	5-10 Sekunden	Kanalspezifische Pfade	URL-Pfad + alle Abfrageparameter
Inhaltssegmente	Mehr als 24 Stunden	5-15 Minuten	Inhaltsspezifische Pfade	Nur URL-Pfad
Anzeigensegmente	Mehr als 24 Stunden	24+ Stunden	Anzeigenspezifische Pfade	Nur URL-Pfad

- Legen Sie kurze TTL-Werte für Manifeste fest, um sicherzustellen, dass Zuschauer Programme erhalten up-to-date

- Konfigurieren Sie längere TTL-Werte für Inhaltssegmente, um die Cache-Effizienz zu maximieren
- Richten Sie Cache-Verhalten ein, das Zeitverschiebungsparameter im Cache-Schlüssel beinhaltet, wenn Sie die zeitversetzte Anzeige unterstützen
- Nehmen Sie Abfrageparameter in den Cache-Schlüssel auf, um zeitversetzte Anzeigeanforderungen ordnungsgemäß zu verarbeiten

Detaillierte TTL-Konfigurationseinstellungen und bewährte Methoden finden Sie unter [the section called “Caching-Optimierung”](#)

Kombiniertes SSAI- und Channel-Assembly-Caching

Stellen Sie bei der Implementierung von Channel Assembly und SSAI sicher, dass Ihre Caching-Strategie für beide Dienste konsistent ist, um Konflikte zu vermeiden und die Leistung zu optimieren:

Vergleich der kombinierten Workflow-Caching-Einstellungen

Inhaltstyp	Montage von Kanälen	SSAI	Kombinierte Empfehlung
VOD-Manifestationen	5-30 Minuten	0 Sekunden	(verwende eine separate Konfiguration)
Live-Manifeste	2-10 Sekunden	0 Sekunden	(verwende eine separate Konfiguration)
SGAI VOD-Manifestationen	5-30 Minuten	5-30 Minuten	5-30 Minuten (zwischen speicherbare Manifeste)
SGAI Live-Manifeste	2-4 Sekunden	2-4 Sekunden	2-4 Sekunden (zwischen speicherbare Manifeste)
Inhaltssegmente	Mehr als 24 Stunden	24+ Stunden	Über 24 Stunden (konsistent)
Anzeigensegmente	Mehr als 24 Stunden	24+ Stunden	Über 24 Stunden (konsistent)

Diese Konfiguration maximiert die Cache-Effizienz und stellt gleichzeitig sicher, dass Zuschauer up-to-date Manifeste für die personalisierte Anzeigeneinblendung erhalten.

Fordern Sie eine Routing-Optimierung für CDN und Integrationen an MediaTailor

Implementieren Sie diese Routing-Optimierungen für alle AWS Elemental MediaTailor CDN-Integrationen:

- Erstellen Sie separate Cache-Verhaltensweisen für Manifest- und Segmentanforderungen
- Konfigurieren Sie Richtlinien für ursprüngliche Anfragen, um die Header-Weiterleitung zu steuern
- Richten Sie geeignete Mechanismen zur Fehlerbehandlung und Failover ein
- Implementieren Sie Origin Shields, sofern in Ihrem CDN verfügbar, um die Origin-Auslastung zu reduzieren
- Implementieren Sie Request Collapsing auf CDN-Ebene, um gleichzeitige Anfragen effizient zu bearbeiten

Leistungsbenchmarks für CDN und Integrationen MediaTailor

Achten Sie bei der Optimierung Ihrer AWS Elemental MediaTailor CDN-Integration auf diese Leistungsbenchmarks. Diese Ziele gelten für alle MediaTailor Implementierungen, einschließlich SSAI, Channel-Assembly und kombinierter Workflows:

Ziele für die Cache-Trefferquote

Inhaltssegmente: Cache-Trefferquote von mehr als 95%

Anzeigensegmente: Cache-Trefferquote von mehr als 90%

Manifeste: Gilt nicht für SSAI (sollte nicht zwischengespeichert werden, um personalisierte Werbung einzufügen); mehr als 85% für Channel-Assembly

Benchmarks für die Latenz

Latenz bei manifesten Anfragen: weniger als 100 ms (P95)

Bereitstellung von Inhaltssegmenten: weniger als 50 ms (P95)

Bereitstellung von Anzeigensegmenten: weniger als 75 ms (P95)

End-to-end Startzeit: weniger als 2 Sekunden

Origin-Lade-Metriken

Origin-Anfragen pro Zuschauer: weniger als 0,1 Anfragen pro Minute pro Zuschauer

Ursprungsbandbreite pro Zuschauer: weniger als 5% der gesamten Zuschauerbandbreite

Ziele für die Fehlerquote

Offensichtliche Fehler: weniger als 0,1%

Segmentfehler: weniger als 0,01%

Von Spielern gemeldete Zurückweisungen: weniger als 1%

Benchmarks für Skalierbarkeit

Support für das Zehnfache des normalen Datenverkehrs bei Spitzenereignissen ohne Beeinträchtigung

Fähigkeit, mehr als 1000 Anfragen pro Sekunde und Kanal zu verarbeiten

Verwenden Sie CloudWatch Amazon-Metriken, um diese Leistungsindikatoren zu verfolgen. Detaillierte Anweisungen zur Überwachung finden Sie unter [Richten Sie Überwachungstools ein](#).

Fortgeschrittene Optimierungstechniken für CDN und Integrationen MediaTailor

Nach der Implementierung grundlegender Caching- und Routing-Optimierungen sollten Sie die folgenden fortgeschrittenen Techniken in Betracht ziehen, um die Leistung weiter zu verbessern:

Origin Shield-Implementierung

Origin Shield fügt eine Caching-Ebene zwischen CDN-Edge-Standorten und deinem Ursprungsserver hinzu, wodurch die Origin-Auslastung reduziert und die Leistung verbessert wird:

- Aktiviere Origin Shield für stark frequentierte Inhalte und Live-Streaming
- Wähle Origin Shield-Standorte in der Nähe deiner MediaTailor Regionen
- Überwache die Trefferquoten im Origin Shield-Cache und passe sie nach Bedarf an
- Ziehen Sie mehrere Origin Shield-Standorte für globale Bereitstellungen in Betracht

Optimierung der Inhaltskomprimierung

Optimieren Sie die Inhaltskomprimierung, um die Bandbreite zu reduzieren und die Übertragungsgeschwindigkeit zu verbessern:

- Aktivieren Sie die Gzip-Komprimierung für Manifestdateien
- Konfigurieren Sie die Accept-Encoding-Headerweiterleitung für die Manifestkomprimierung MediaTailor
- Verwenden Sie die Brotli-Komprimierung, sofern sie unterstützt wird, um zusätzliche Bandbreiteneinsparungen zu erzielen
- Vermeiden Sie es, bereits komprimierte Videosegmente zu komprimieren

Strategien zur regionalen Optimierung

Optimieren Sie die Leistung für globale Zielgruppen durch regionale Strategien:

- Stellen Sie MediaTailor Konfigurationen in mehreren Regionen für globale Zielgruppen bereit
- Verwenden Sie Geo-Routing, um Zuschauer in die nächstgelegene Region zu leiten MediaTailor
- Konfigurieren Sie regionales Failover für hohe Verfügbarkeit
- Überwachen Sie die regionalen Leistungskennzahlen separat

Überlegungen zur Architektur von CDN und Integrationen MediaTailor

Positionieren Sie Ihr Content Delivery Network (CDN) korrekt in Ihrer Architektur, um eine optimale Leistung und Zuverlässigkeit zu gewährleisten. AWS Elemental MediaTailor Die empfohlene Architektur platziert das CDN zwischen Zuschauern und MediaTailor nicht zwischen Ihrer MediaTailor Herkunft.

Ausführliche Architekturdiagramme und Erläuterungen zum Arbeitsablauf finden Sie in den folgenden Themen.

- [Einfügung von Anzeigen mit CDN](#) für Architekturdiagramme zum Einfügen von Anzeigen und detaillierte Arbeitsabläufe
- [Verstehen Sie die CDN-Architektur](#) für Architekturdiagramme und Arbeitsabläufe der Kanalbaugruppe

Positionieren Sie Ihr CDN korrekt in Ihrer Architektur:

1. Platziere dein CDN zwischen Spielern und MediaTailor (nicht zwischen MediaTailor und deinem Ursprung).

Diese Architektur ermöglicht es Ihrem CDN, Anzeigensegmente und Inhaltssegmente zwischenspeichern. Gleichzeitig MediaTailor können personalisierte Manifeste für jeden Zuschauer generiert werden.

2. Erstellen Sie separate Cache-Verhaltensweisen für verschiedene Anforderungstypen:

- Manifeste Anfragen (kein Caching)
- Inhaltssegmente (längere TTL)
- Anzeigensegmente (längere TTL)

3. Konfigurieren Sie die richtige Fehlerbehandlung:

- Richten Sie negatives Caching ein (temporäres Speichern von Fehlerantworten), um zu verhindern, dass Ihr Absender bei Dienstunterbrechungen mit wiederholten Anfragen überlastet wird. Negatives Caching bedeutet, dass das CDN Fehlerantworten (wie 404- oder 500-Fehler) vorübergehend speichert, um wiederholte Anfragen nach Inhalten zu verhindern, die nicht existieren oder vorübergehend nicht verfügbar sind.
- Konfigurieren Sie die entsprechenden Fehlerantwortcodes und das Verhalten bei Wiederholungsversuchen

4. Implementieren Sie Zwischen-Caching (Origin Shield):

Origin Shield ist eine Funktion, die eine zusätzliche Caching-Ebene zwischen CDN-Edge-Standorten und Ihrem Ursprungsserver erstellt. Dies reduziert die Anzahl redundanter Anfragen, die Ihren Ursprungsserver erreichen.

- Konfigurieren Sie eine Zwischen-Caching-Ebene zwischen den Edge-Standorten und Ihrem Ursprung
- Reduzieren Sie die Anzahl redundanter Anfragen an Ihren Ursprung bei Cache-Fehlern
- Verbessern Sie die Cache-Trefferquoten in Ihrer gesamten CDN-Infrastruktur

Erweiterte CDN-Funktionen für MediaTailor

Erkunden Sie nach der Implementierung der grundlegenden CDN-Konfiguration (Content Delivery Network) diese erweiterten Funktionen, um die Leistung und Zuverlässigkeit Ihrer AWS Elemental MediaTailor Streaming-Plattform weiter zu verbessern.

Medienqualitätsbewusstes Routing (MQAR)

MQAR ist eine CloudFront Amazon-Funktion, die automatisch die qualitativ hochwertigste Inhaltsquelle auf der Grundlage von Netzwerkleistungsmetriken in Echtzeit auswählt. Anstatt einen festen Ursprungsserver zu verwenden, leitet MQAR Anfragen anhand von Faktoren wie Latenz und Durchsatz dynamisch an den Ursprung mit der besten Leistung weiter. Auf diese Weise wird sichergestellt, dass Zuschauer auch bei Netzwerkschwankungen Streams von höchster Qualität erhalten.

Wenn Sie Amazon verwenden CloudFront, implementieren Sie MQAR, um automatisch die qualitativ hochwertigste Inhaltsquelle auf der Grundlage von Echtzeitmetriken auszuwählen. Einzelheiten finden Sie [CloudFront-Integration](#) im Abschnitt CloudFront Integration.

Manifestfilterung

Mit der Manifestfilterung können Sie anpassen, welche Wiedergabeversionen (unterschiedliche Qualitätsversionen desselben Inhalts) in den Manifesten enthalten sein sollen, die den Zuschauern zur Verfügung gestellt MediaTailor werden. Die Filterung trägt zur Optimierung der Bandbreitennutzung bei, indem Wiedergabeversionen entfernt werden, die für bestimmte Geräte oder Netzwerkbedingungen nicht geeignet sind. Sie könnten beispielsweise 4K-Wiedergabeversionen für Mobilgeräte oder Verbindungen mit geringer Bandbreite entfernen.

Ausführliche Informationen zur Implementierung der Manifestfilterung mit AWS Elemental MediaPackage finden Sie unter. [MediaPackage CDN-Integration](#)

Multi-CDN-Strategie

Bei einer Multi-CDN-Strategie werden mehrere CDN-Anbieter gleichzeitig verwendet, um die Zuverlässigkeit und Leistung zu verbessern. Wenn bei einem CDN Probleme auftreten, kann der Datenverkehr automatisch zu einem anderen Anbieter umgeleitet werden. Dieser Ansatz ist besonders nützlich für hochkarätige Live-Events, bei denen Zuverlässigkeit entscheidend ist.

Informationen zur Implementierung einer Multi-CDN-Strategie mit MediaTailor finden Sie unter. [Planen Sie die CDN-Integration](#)

Überwachen Sie den Betrieb und die Leistung von MediaTailor CDN

Eine effektive Überwachung Ihrer Integration AWS Elemental MediaTailor und der Integration von Content Delivery Network (CDN) gewährleistet eine zuverlässige Inhaltsbereitstellung, optimale

Leistung und eine schnelle Problemerkennung. Dieser Überwachungsansatz gilt für alle MediaTailor Implementierungen, einschließlich serverseitiger Anzeigeneinfügung (SSAI), Kanalzusammenstellung und kombinierter Workflows.

Die Überwachung Ihrer CDN-Integration ermöglicht Ihnen:

- Probleme erkennen und lösen, bevor sie sich auf Zuschauer auswirken
- Verfolgen Sie wichtige Leistungsindikatoren und sorgen Sie für eine gleichbleibende Servicequalität
- Sorgen Sie für eine optimale Cache-Leistung und reduzieren Sie die Ausgangslast
- Stellen Sie sicher, dass die Erfolgsquoten bei der Anzeigenschaltung den Geschäftsanforderungen entsprechen

Informationen zur Behebung von parameterbezogenen Problemen, die in Überwachungsdaten auftreten können, finden Sie unter. [MediaTailor Leitfaden zur Fehlerbehebung bei Parametern](#)
Hinweise zur Überwachung der Verwendung von Abfrageparametern finden Sie unter. [MediaTailor Manifest-Abfrageparameter](#)

Wichtige CDN-Leistungskennzahlen

Verfolgen Sie diese Kernkennzahlen, um die Effektivität Ihres CDN bei MediaTailor Implementierungen zu bewerten:

Trefferquote im Cache

Was es misst: Der Prozentsatz der Anfragen, die vom CDN-Cache aus bedient werden, im Vergleich zu Anfragen vom Ursprung.

Zielwerte:

- Inhaltssegmente: Cache-Trefferquote von 95% oder höher
- Anzeigensegmente: Cache-Trefferquote von mindestens 90%
- Manifeste: Variiert je nach Implementierung (personalisierte Manifeste sollten nicht zwischengespeichert werden)

Warum das wichtig ist: Höhere Cache-Trefferquoten reduzieren die Ausgangslast, verbessern die Reaktionszeiten und senken die Bandbreitenkosten.

Volumen der ursprünglichen Anfragen

Was es misst: Die Anzahl der Anfragen, die Ihre MediaTailor Ursprungsserver erreichen.

Zielmuster: Sollte niedrig und stabil bleiben, mit gelegentlichen Spitzenwerten aufgrund von Cache-Fehlschlägen oder neuen Inhalten.

Warum das wichtig ist: Ein hohes Volumen an ursprünglichen Anfragen weist auf Ineffizienzen beim Zwischenspeichern hin und kann sich negativ auf die Leistung auswirken. MediaTailor

Latenz bei der Antwort

Was es misst: Wie schnell das CDN auf Zuschaueranfragen reagiert.

Zielwerte:

- Inhalt im Cache: Weniger als 100 ms (P95)
- Ursprüngliche Anfragen: Weniger als 500 ms (P95)
- Manifeste Anfragen: Weniger als 100 ms (P95)
- Anfragen segmentieren: Weniger als 50 ms (P95)

Warum das wichtig ist: Die geringe Latenz sorgt für eine reibungslose Wiedergabe und ein gutes Zuschauererlebnis.

Fehlerquoten

Was es misst: Der Prozentsatz der Anfragen, die zu HTTP-Fehlern führen.

Zielwerte:

- 4xx-Fehler: Weniger als 0,1% der gesamten Anfragen
- 5xx-Fehler: Weniger als 0,01% der gesamten Anfragen
- Ursprungsfehler: Weniger als 0,05% der ursprünglichen Anfragen

Warum das wichtig ist: Hohe Fehlerraten deuten auf Konfigurationsprobleme oder Serviceprobleme hin, die das Zuschauererlebnis beeinträchtigen.

MediaTailor spezifische Metriken

Überwachen Sie diese MediaTailor Metriken zusammen mit CDN-Metriken, um einen vollständigen Überblick zu erhalten:

Füllraten von Anzeigen

Wichtige Kennzahlen: `Avail.FillRate` und `AdDecisionServer.FillRate`

Zielwerte: Über 90% für beide Kennzahlen

Warum das wichtig ist: Wirkt sich direkt auf die Werbeeinnahmen und das Zuschauererlebnis aus
Manifeste Leistung bei der Generierung

Wichtige Kennzahlen: `GetManifest.Latency` und `GetManifest.Errors`

Zielwerte: Latenz unter 200 ms, Fehlerrate unter 1%

Warum das wichtig ist: Beeinträchtigt die Startzeit und Zuverlässigkeit der Wiedergabe
Und Entscheidung: Zustand des Servers

Wichtige Kennzahlen: `AdDecisionServer.Latency`, `AdDecisionServer.Errors`, und
`AdDecisionServer.Timeouts`

Zielwerte: Latenz unter 1000 ms, Fehlerrate unter 5%, minimale Timeouts

Warum das wichtig ist: Die ADS-Leistung wirkt sich direkt auf den Erfolg der Anzeigeneinblendung aus

Richten Sie Überwachungstools ein

Konfigurieren Sie diese Tools, um Ihre MediaTailor und die CDN-Integration effektiv zu überwachen:

CloudWatch Amazon-Integration

Amazon CloudWatch bietet die Grundlage für die Überwachung Ihrer MediaTailor und CDN-Integration:

MediaTailor Metriken

MediaTailor veröffentlicht automatisch Metriken CloudWatch , um Anfragen, Antworten und Fehler nachzuverfolgen. Zu den wichtigsten Metriken gehören:

- `RequestCount`: Gesamtzahl der Anfragen an MediaTailor
- `ResponseTime`: MediaTailor Antwortlatenz
- `4xxErrorCount` und `5xxErrorCount`: Fehlerverfolgung

Eine vollständige Liste der MediaTailor Messwerte finden Sie unter [Überwachung AWS Elemental MediaTailor mit CloudWatch Amazon-Metriken](#).

CDN-Metriken

Aktivieren Sie die Erfassung von CDN-Metriken, einschließlich CloudWatch :

- Cache-Trefferquoten für verschiedene Inhaltstypen
- Anzahl der ursprünglichen Anfragen und Antwortzeiten
- Fehlerquoten nach Statuscode

Aktivieren Sie für CloudFront bestimmte Messwerte die detaillierte Überwachung in Ihren Vertriebseinstellungen.

Konfiguration des Dashboards

Erstellen Sie Dashboards, die Ihnen Einblick in Ihre Leistung MediaTailor und die Leistung Ihres CDN bieten:

1. Erstellen Sie ein einheitliches Dashboard: Kombinieren Sie CDN-Metriken MediaTailor und -CDN-Metriken in einem einzigen CloudWatch Dashboard, um einen vollständigen Überblick zu erhalten.
2. Nach Arbeitsablauf organisieren: Gruppieren Sie die Kennzahlen nach Implementierungstyp (SSAI, Channel-Assembly oder kombinierte Workflows).
3. Fügen Sie wichtige Leistungsindikatoren hinzu:
 - Trends der Cache-Trefferquote im Zeitverlauf
 - Perzentile für die Antwortlatenz (P50, P95, P99)
 - Trends und Spitzen bei der Fehlerquote

Eine ausführliche Anleitung zur Erstellung von Dashboards finden Sie unter [CloudWatchDashboards](#).

Konfiguration von Warnmeldungen

Konfigurieren Sie Benachrichtigungen, um Probleme zu erkennen, bevor sie sich auf Zuschauer auswirken:

Kritische Warnmeldungen

Konfigurieren Sie sofortige Benachrichtigungen für schwerwiegende Probleme:

- Hohe Fehlerraten: Warnung, wenn 5xx-Fehler innerhalb von 5 Minuten 0,1% der Anfragen überschreiten

- Probleme mit dem Ursprungsserver: Warnung, wenn die Antwortzeit des Origins 2 Sekunden überschreitet
- Verringerung der Cache-Trefferquote: Warnung, wenn die Cache-Trefferquote bei Manifesten unter 70% oder bei Segmenten unter 85% fällt

Warnmeldungen

Konfigurieren Sie Frühwarnungen für Leistungseinbußen:

- Erhöhung der Latenz: Warnung, wenn die P95-Reaktionszeit 200 ms überschreitet
- Rückgang der Cache-Effizienz: Warnung, wenn die Cache-Trefferquote für Segmente unter 90% fällt

Checkliste für die Implementierung

Verwenden Sie diese Checkliste, um eine umfassende Überwachung sicherzustellen:

1. Erfassung von Kennzahlen:

- ✓ MediaTailor Metriken aktiviert in CloudWatch
- ✓ Detaillierte CDN-Überwachung aktiviert

2. Einrichtung des Dashboards:

- ✓ Einheitliches CloudWatch Dashboard erstellt
- ✓ Die wichtigsten Kennzahlen wurden mit entsprechenden Zeiträumen visualisiert
- ✓ Das Dashboard wird mit den entsprechenden Teams geteilt

3. Konfiguration der Warnmeldungen:

- ✓ Konfiguration kritischer Warnmeldungen mit sofortiger Benachrichtigung
- ✓ Zur Früherkennung wurden Warnmeldungen eingerichtet
- ✓ Die Verfahren zur Eskalation von Alarmen sind dokumentiert

4. Operative Verfahren:

- ✓ Dokumentierte Verfahren zur Reaktion auf Vorfälle
- ✓ Zeitplan für regelmäßige Überprüfungen festgelegt
- ✓ Teamtraining abgeschlossen

Verwandte Themen

Weitere Hinweise zu bestimmten Überwachungsszenarien finden Sie unter:

- Leistungsoptimierung: Detaillierte Optimierungstechniken, die auf Überwachungsdaten basieren, finden Sie unter [CDN-Leistungsoptimierung](#).
- Problembehandlung: Ausführliche Verfahren zur Fehlerbehebung anhand von Überwachungsdaten finden Sie in Ihrer Workflow-spezifischen Dokumentation zur Problembehandlung.
- Protokollanalyse: Eine umfassende Protokollanalyse und Überwachung finden Sie unter. [Analyse des CDN-Integrationsprotokolls und Fehlercode-Referenz für MediaTailor](#)

Testen und Validieren für CDN und Integrationen MediaTailor

AWS Elemental MediaTailor Die Integration von Content Delivery Network (CDN) erfordert gründliche Tests, um eine zuverlässige Anzeigenbereitstellung und optimale Leistung sicherzustellen.

Systematische Tests helfen dabei, Integrationsprobleme zu identifizieren, bevor sie sich auf die Zuschauer auswirken. Nutzen Sie diesen umfassenden Ansatz, um Ihr Content Delivery Network und Ihre MediaTailor Integration für verschiedene Szenarien, Konfigurationen und Ladebedingungen zu validieren.

Verwandte Themen:

- Informationen zur Betriebseinrichtung einschließlich Überwachung und Protokollierung finden Sie unter [CDN-Überwachung](#)
- Informationen zur Behebung von Problemen finden Sie unter [Beheben Sie Probleme mit der CDN-Integration](#)
- Anleitungen zur Leistungsoptimierung finden Sie unter [Leitfaden zur Leistungsoptimierung für CDN und Integrationen MediaTailor](#)

Themen

- [Testvoraussetzungen und Einrichtung für CDN und Integrationen MediaTailor](#)
- [Systematische Testmethodik für CDN und Integrationen MediaTailor](#)
- [Checkliste für Tests vor der Bereitstellung für CDN und Integrationen MediaTailor](#)
- [Referenz zu Testtools und Hilfsprogrammen für CDN und Integrationen MediaTailor](#)

Testvoraussetzungen und Einrichtung für CDN und Integrationen MediaTailor

AWS Elemental MediaTailor CDN-Integrationstests (Content Delivery Network) erfordern eine angemessene Vorbereitung und Ressourcenzuweisung. Bevor Sie mit systematischen Tests beginnen, stellen Sie sicher, dass Sie über die erforderlichen Ressourcen und Tools verfügen.

Erforderliche Ressourcen:

- Testen Sie die CDN-Verteilung, die so konfiguriert ist, dass sie die Produktionseinstellungen widerspiegelt
- MediaTailor Testkonfiguration mit bekannten Inhalts- und Anzeigenquellen
- Testen Sie Inhalte mit vorhersehbaren Merkmalen (Dauer, Format, Markierungen für Werbeunterbrechungen)
- Testen Sie den Anzeigenentscheidungsserver oder simulieren Sie ADS-Antworten
- Mehrere Testgeräte und Playertypen

Tools zum Testen:

- `curl` zum Testen von HTTP-Anfragen
- `ffprobe` für die Validierung des HLS-Manifests
- `mp4box` zur Validierung des DASH-Manifests
- Tools für Browser-Entwickler zur Netzwerkanalyse
- Videoplayer zum end-to-end Testen

Systematische Testmethodik für CDN und Integrationen MediaTailor

AWS Elemental MediaTailor Die Integrationstests für Content Delivery Networks (CDN) sollten einem systematischen, schrittweisen Ansatz folgen, um eine umfassende Abdeckung sicherzustellen. Folgen Sie diesem strukturierten Ansatz, um Ihr Content Delivery Network und Ihre Integration umfassend zu testen. MediaTailor Jede Phase baut auf der vorherigen auf, um potenzielle Probleme zu isolieren.

Weitere Hinweise zu systematischen Testansätzen finden Sie unter [Zuverlässigkeitstests](#) im AWS Well-Architected Framework.

Phase 1: Testen Sie die direkte Konnektivität MediaTailor

Testen Sie zunächst die MediaTailor Funktionalität ohne CDN-Beteiligung, um eine Ausgangsbasis festzulegen.

1. Testen Sie Manifestanfragen direkt an MediaTailor Endpunkte:

- Testen Sie multivariante HLS-Playlist-Anfragen: `curl -v "https://your-emt-endpoint.mediatailor.region.amazonaws.com/v1/master/hls/config-name/master.m3u8"`
- Testen Sie DASH MPD-Anfragen: `curl -v "https://your-emt-endpoint.mediatailor.region.amazonaws.com/v1/dash/config-name/manifest.mpd"`
- Stellen Sie sicher, dass die Antworten die erwarteten Markierungen für Werbeunterbrechungen enthalten
- Prüfen Sie, ob das Segment URLs auf die richtigen Ursprünge verweist

2. Stellen Sie sicher, dass das Einfügen von Anzeigen korrekt funktioniert:

- Testen Sie mit verschiedenen Parametern für die Anzeigenausrichtung
- Stellen Sie sicher, dass die Anzeigensegmente ordnungsgemäß transcodiert und verfügbar sind
- Überprüfen Sie den Zeitpunkt und die Dauer der Werbeunterbrechungen
- Testen Sie das Fallback-Verhalten, wenn Anzeigen nicht verfügbar sind

3. Messen Sie die Ausgangsleistung:

- Erfassen Sie die Antwortzeiten für Manifestanfragen
- Messen Sie die Antwortzeiten des Ad Decision Servers
- Erstellung und Verwaltung von Testsitzungen

Erfolgskriterien: Alle direkten MediaTailor Anfragen geben HTTP 200-Antworten mit korrekt formatierten Manifesten zurück, die den erwarteten Anzeigeninhalt enthalten.

Phase 2: Testen Sie die grundlegende CDN-Integration

Fügen Sie CDN zum Anforderungspfad hinzu und testen Sie die grundlegenden Funktionen.

1. Testen Sie Manifestanfragen über CDN:

- Konfigurieren Sie CDN mit einfachen Routing-Regeln
- Testen Sie Manifestanfragen über CDN-Endpunkte

- Stellen Sie sicher, dass CDN Anfragen korrekt weiterleitet an MediaTailor
- Stellen Sie sicher, dass die Manifestantworten nicht zwischengespeichert sind (TTL = 0)

2. Segment-Routing testen:

- Überprüfen Sie, ob die Inhaltssegmente zum Ursprungsserver weitergeleitet werden
- Überprüfen Sie die Route der Anzeigensegmente zum MediaTailor Anzeigenspeicher
- Testen Sie das Verhalten beim Zwischenspeichern von Segmenten

3. Vergleichen Sie CDN mit direkter Leistung:

- Messen Sie die Unterschiede in der Reaktionszeit
- Prüfen Sie, ob die Antworten inhaltlich unterschiedlich sind
- Stellen Sie sicher, dass die Fehlerbehandlung korrekt funktioniert

Erfolgskriterien: Das CDN sollte Anfragen erfolgreich per Proxy an MediaTailor und an die Ursprungsserver weiterleiten, ohne die Leistung zu beeinträchtigen.

Phase 3: Testen Sie die Weiterleitung von Abfrageparametern

Fügen Sie die Weiterleitung von Abfrageparametern hinzu und testen Sie die Personalisierung von Anzeigen.

1. Konfigurieren Sie die Weiterleitung von Abfrageparametern im CDN:

- Aktivieren Sie die Weiterleitung aller Abfrageparameter an MediaTailor
- Initialisierung der Testsitzung (die Sitzungs-ID wird bei MediaTailor der ersten Anfrage automatisch generiert)
- Testen Sie mit benutzerdefinierten Targeting-Parametern

2. Testen Sie die Personalisierung von Anzeigen:

- Stellen Sie sicher, dass verschiedene Parameter zu unterschiedlichen Anzeigenreaktionen führen
- Testen Sie die Kodierung der Parameter und Sonderzeichen
- Prüfen Sie, ob die Parameter korrekt an ADS übergeben wurden

3. Überprüfen Sie die Sitzungsverwaltung:

- Erstellung und Persistenz von Testsitzungen
- Überprüfen Sie die Konsistenz der Sitzungs-IDs für alle Anfragen
- Behandlung von Sitzungsabläufen testen

Erfolgskriterien: Der Anzeigeninhalt variiert je nach Abfrageparametern, und die Sitzungen werden korrekt verwaltet.

Phase 4: Testen Sie die Header-Weiterleitung

Fügen Sie Header-Weiterleitung im CDN hinzu und testen Sie das gerätespezifische Targeting.

1. Konfigurieren Sie die Header-Weiterleitung für alle Header. Die Mindestanforderungen finden Sie unter [Erforderliche Header für MediaTailor die CDN-Integration](#).
2. Geräte-Targeting testen:
 - Testen Sie mit verschiedenen User-Agent-Strings (Handy, Desktop, TV)
 - Überprüfen Sie gerätespezifische Anzeigenantworten
 - Testen Sie das geografische Targeting mit verschiedenen IP-Adressen
3. Überprüfen Sie die CORS-Behandlung:
 - Testen Sie die CORS-Header auf Kompatibilität mit dem Webplayer
 - Stellen Sie sicher, dass Preflight-OPTIONS-Anfragen ordnungsgemäß funktionieren
 - Testen Sie von verschiedenen Domänen aus

Erfolgskriterien: Geräte- und geografisches Targeting sollten korrekt funktionieren, und bei Web-Playern sollten keine CORS-Fehler auftreten.

Phase 5: Umfassendes Testen von Szenarien

Testen Sie in mehreren Szenarien, um einen robusten Betrieb sicherzustellen.

1. Testen Sie mit verschiedenen Spielertypen:
 - Spieler von HLS.js in Webbrowsern
 - Video.js Spieler mit HLS- und DASH-Unterstützung
 - Native Spieler auf Mobilgeräten
 - Smart-TV- und Set-Top-Box-Player
2. Testen Sie auf verschiedenen Geräten und Plattformen:
 - Mobilgeräte (iOS, Android)
 - Desktop-Browser (Chrome, Firefox, Safari, Edge)
 - Intelligente Geräte TVs und Streaming-Geräte
 - Verschiedene Betriebssystemversionen

3. Testen Sie verschiedene Inhaltstypen:

- Inhalte live streamen
- Video-on-Demand-Inhalte (VOD)
- Verschiedene Videoformate und Bitraten
- Inhalte mit unterschiedlichen Mustern für Werbeunterbrechungen

4. Testen Sie Szenarien für die Ausrichtung von Anzeigen:

- Verschiedene demografische Targeting-Parameter
- Geografisches Targeting in verschiedenen Regionen
- Zeitbasiertes Targeting (verschiedene Tageszeiten)
- Benutzerdefinierte Parameter für die Geschäftslogik

Erfolgskriterien: Alle Player- und Gerätekombinationen sollten bei entsprechender Anzeigenausrichtung korrekt funktionieren.

Phase 6: Last- und Leistungstests

Überprüfen Sie die Leistung unter realistischen Lastbedingungen.

Important

Wenden Sie sich vor dem Auslastungstest an den [AWS-Support](#): Bevor Sie Last- und Leistungstests durchführen, erstellen Sie ein AWS Support-Ticket, um das MediaTailor Serviceteam über Ihre geplanten Tests zu informieren. Dadurch wird Folgendes gewährleistet:

- Der Service ist auf Ihre zu erwartenden Auslastungsgrade vorbereitet
- Während Ihres Testfensters steht angemessene Kapazität zur Verfügung
- Ihre Tests werden nicht mit einem Produktionsvorfall verwechselt
- Sie erhalten Anleitungen zu bewährten Testpraktiken und Einschränkungen

Geben Sie in Ihrem Support-Ticket Folgendes an: erwartete gleichzeitige Benutzer, Testdauer, geografische Regionen und alle spezifischen Szenarien, die Sie testen möchten.

1. Testen Sie Szenarien mit gleichzeitigen Benutzern:

- Simulieren Sie mehrere gleichzeitige Zuschauer
- Testen Sie die CDN-Skalierung und die Cache-Leistung
- Überwachen Sie die Leistung des Originalservers unter Last
- Verify MediaTailor kann gleichzeitige Sitzungen verarbeiten

2. Leistungskennzahlen messen:

- Überwachen Sie die Reaktionszeiten unter Last
- Prüfen Sie, ob die Cache-Trefferquoten den Erwartungen entsprechen (> 80% für beliebte Inhalte)
- Messen Sie die Zeit bis zum ersten Frame für verschiedene Szenarien
- Verfolgen Sie die Fehlerraten bei Spitzenlast

3. Testen Sie Failover-Szenarien:

- Testen Sie das Verhalten, wenn ADS nicht verfügbar ist
- Testen Sie das Failover des Ursprungsservers
- Überprüfen Sie die Fehlerbehandlung und Wiederherstellung
- Testen Sie den CDN-Edge-Standort-Failover

Erfolgskriterien: Das System sollte unter erwarteter Last eine akzeptable Leistung aufrechterhalten und bei Ausfällen schrittweise abnehmen. Stellen Sie sicher, dass Sie sich an den [AWS-Support](#) wenden, damit dieser Ihren Lasttestplan vor der Ausführung genehmigt.

Checkliste für Tests vor der Bereitstellung für CDN und Integrationen MediaTailor

AWS Elemental MediaTailor Die Integration von Content Delivery Network (CDN) muss vor der Produktionsbereitstellung umfassende Tests bestehen. Verwenden Sie diese Checkliste, bevor Sie Konfigurationsänderungen in der Produktion implementieren.

Grundlegende Funktionen:

- ☐ Manifest-Anfragen geben HTTP 200-Antworten zurück
- ☐ Inhaltssegmente werden korrekt geladen
- ☐ Anzeigensegmente werden korrekt geladen
- ☐ Werbeunterbrechungen werden zu den erwarteten Zeiten angezeigt
- ☐ Die Wiedergabe geht reibungslos zwischen Inhalten und Anzeigen über

Überprüfung der Konfiguration:

- ☐ Abfrageparameter werden korrekt weitergeleitet
- ☐ Erforderliche Header werden korrekt weitergeleitet
- ☐ Das Zwischenspeichern von Manifesten ist deaktiviert (TTL = 0)
- ☐ Das Segment-Caching ist entsprechend konfiguriert
- ☐ CORS-Header sind für Webplayer konfiguriert

Plattformübergreifendes Testen:

- ☐ Auf Mobilgeräten getestet
- ☐ Auf Desktop-Browsern getestet
- ☐ Mit verschiedenen Spielertypen getestet
- ☐ Sowohl das HLS- als auch das DASH-Format wurden getestet

Validierung der Leistung:

- ☐ Für die Genehmigung von Belastungstests wurde ein AWS Support-Ticket erstellt
- ☐ Die Reaktionszeiten entsprechen den Leistungszielen
- ☐ Die Cache-Trefferquoten sind akzeptabel
- ☐ Die Fehlerraten liegen innerhalb akzeptabler Grenzen
- ☐ Überwachung und Alarmierung sind konfiguriert

Referenz zu Testtools und Hilfsprogrammen für CDN und Integrationen MediaTailor

AWS Elemental MediaTailor Für Integrationstests im Content Delivery Network (CDN) sind verschiedene Tools für eine umfassende Validierung und Debugging erforderlich. Referenzhandbuch für Tools, die häufig bei Content Delivery Network- und MediaTailor Integrationstests verwendet werden.

curl- Testen von HTTP-Anfragen

Testen Sie Manifest-Anfragen: `curl -v "https://your-cdn-domain.com/v1/master/hls/config/master.m3u8"`

Testen Sie mit Headern: `curl -H "User-Agent: TestAgent/1.0" "https://your-cdn-domain.com/..."`

Testen Sie mit Parametern: `curl "https://your-cdn-domain.com/...?aws.sessionId=test123"`

ffprobe- Validierung des HLS-Manifests

Überprüfen Sie die HLS-Syntax: `ffprobe -v quiet -print_format json -show_format "https://your-cdn-domain.com/master.m3u8"`

Überprüfen Sie die Segmentinformationen: `ffprobe -v quiet -show_entries packet=pts_time "segment.ts"`

mp4box- Validierung des DASH-Manifests

DASH MPD validieren: `mp4box -info "https://your-cdn-domain.com/manifest.mpd"`

Überprüfen Sie das Timing der Segmente: `mp4box -info segment.m4s`

Tools für Browser-Entwickler

Überwachen Sie Netzwerkanfragen auf der Registerkarte Netzwerk

Suchen Sie auf der Registerkarte „Konsole“ nach CORS-Fehlern

Untersuchen Sie die Header request/response

Analysieren Sie Timing- und Leistungskennzahlen

Zusätzliche Ressourcen:

- [Fehlerbehebung CloudFront](#) — Umfassender Leitfaden zur CDN-Fehlerbehebung
- [Erhöhen Sie die CloudFront Cache-Trefferquote](#) — Anleitung zur Leistungsoptimierung
- [AWS Ressourcen überwachen](#) — Bewährte Methoden zur Leistungsüberwachung

Probleme mit MediaTailor der CDN-Integration beheben

Dieser umfassende Leitfaden zur Problembehebung behandelt häufig auftretende Probleme bei der Integration von Content Delivery Network (CDN) in allen AWS Elemental MediaTailor Implementierungen, einschließlich serverseitiger Anzeigeneinfügung (SSAI), Channel-Assembly und Integration. AWS Elemental MediaPackage Wenn bei Ihrem CDN und Ihrer MediaTailor Integration

Probleme auftreten, verwenden Sie diesen systematischen Diagnoseansatz, um die Grundursachen schnell zu identifizieren und validierte Lösungen zu implementieren.

Dieser Leitfaden gilt für alle MediaTailor CDN-Integrationen, unabhängig von Ihrem spezifischen Arbeitsablauf. Informationen zu Problemen, die sich auf bestimmte Dienste oder Workflows beziehen, finden Sie in den entsprechenden Abschnitten zur Fehlerbehebung, auf die am Ende dieses Handbuchs verwiesen wird.

Bevor Sie beginnen: Halten Sie für eine effiziente Problembehebung die folgenden Punkte bereit:

- Wiedergabebeispiele URLs , die das Problem verdeutlichen
- CDN-Zugriffsprotokolle aus dem Zeitraum, in dem Probleme aufgetreten sind
- MediaTailor Name und AWS Region der Konfiguration
- Spielertyp und Version (z. B. HLS.js 1.4.0, Video.js 8.0)
- Geräte- und Browserinformationen, bei denen Probleme auftreten

Verwandte Themen:

- Informationen zur Einrichtung des Betriebs und zur Vorbereitung der Fehlerbehebung finden Sie unter [Beheben Sie Probleme mit der CDN-Integration](#)
- Eine Protokollanalyse und eine Referenz zu Fehlercodes finden Sie unter [Referenz zur Analyse des CDN-Integrationsprotokolls](#)
- Informationen zur Eskalation und weitere Hilfe finden Sie unter [Holen Sie sich Unterstützung bei der CDN-Integration](#)

Diagnose-Checkliste für MediaTailor und CDN-Integrationen

AWS Elemental MediaTailor Probleme bei der Integration von Content Delivery Network (CDN) können sich auf verschiedene Weise äußern. Anhand dieser Checkliste können Sie schnell herausfinden, welche Art von Problem bei Ihnen auftritt:

1. Betrifft das Problem alle Zuschauer oder bestimmte Zuschauer?
 - Alle Zuschauer → Wahrscheinlich CDN- oder MediaTailor Konfigurationsproblem
 - Bestimmte Zuschauer → Wahrscheinlich ein Problem mit Personalisierung oder Targeting
2. Werden die Manifeste korrekt geladen?
 - Nein → CDN-Routing- oder MediaTailor Konnektivitätsproblem

- Ja, aber falscher Inhalt → Problem mit dem Caching oder der Personalisierung
3. Werden Segmente korrekt geladen?
- Inhaltssegmente schlagen fehl → Problem mit der Origin-Konnektivität
 - Anzeigensegmente schlagen fehl → Problem mit der Anzeigenbereitstellung oder der Transcodierung
4. Werden Anzeigen korrekt eingefügt?
- Es erscheinen keine Anzeigen → Überprüfen Sie die ADS-Konnektivität und -Konfiguration
 - Falsche Anzeigen erscheinen → Überprüfen Sie die Parameter für die Anzeigenausrichtung und die Personalisierung
 - Anzeigen können nicht abgespielt werden → Überprüfe die Transcodierung von Anzeigen und die Verfügbarkeit von Segmenten
5. Ist die Wiedergabe flüssig und unterbrechungsfrei?
- Probleme beim Puffern → Überprüfen Sie die Leistung des CDN-Caches und die Antwortzeiten des Originals
 - Wiedergabefehler → Überprüfen Sie die Manifestsyntax und die Verfügbarkeit von Segmenten
 - Probleme beim Übergang von Anzeigen → Überprüfen Sie den Zeitpunkt der Anzeigenunterbrechung und die Ausrichtung der Segmente
6. Gibt es bestimmte Fehlercodes oder Meldungen?
- HTTP 4xx-Fehler → Überprüfen Sie das CDN-Routing und die Konfiguration
 - HTTP 5xx-Fehler → Überprüfen Sie den Zustand des Ursprungsservers und des Dienstes MediaTailor
 - Spielerspezifische Fehler → Überprüfe das Manifestformat und die Player-Kompatibilität

Die nächsten Schritte basieren auf Ihrer Diagnose:

Probleme mit der CDN-Konfiguration

Eine ausführliche Problembehandlung bei CDN-Routing und -Caching finden Sie unter [Probleme mit MediaTailor der CDN-Integration beheben](#)

Probleme mit dem Manifest und der Wiedergabe

Informationen zur Überprüfung von Manifesten und zur Problembehebung bei der Wiedergabe finden Sie unter [Testverfahren für die CDN-Integration](#).

Probleme beim Einfügen von Anzeigen und beim Targeting

Informationen zur Behebung anzeigenspezifischer Probleme, einschließlich ADS-Konnektivität und Anzeigenbereitstellung, finden Sie in Ihrer Workflow-spezifischen Dokumentation zur Fehlerbehebung.

Leistungs- und Überwachungsprobleme

Informationen zur Einrichtung der Leistungsanalyse und Überwachung finden Sie unter [Überwachen Sie den Betrieb und die Leistung von MediaTailor CDN](#).

Protokollanalyse und Fehlercodes

Eine ausführliche Protokollanalyse und eine Referenz zu Fehlercodes finden Sie unter [Analyse des CDN-Integrationsprotokolls und Fehlercode-Referenz für MediaTailor](#).

Testen und Validieren

Umfassende Testverfahren finden Sie unter [Testen und Validieren für CDN und Integrationen MediaTailor](#).

Wenn Sie sofortige Hilfe benötigen oder das Problem mithilfe der verlinkten Ressourcen nicht lösen können, finden Sie Informationen zu den [Holen Sie sich Support und Hilfe zur Fehlerbehebung für CDN und Integrationen MediaTailor](#) Eskalationsverfahren unter.

Testverfahren für die CDN-Integration

Richtige Tests sind unerlässlich, bevor Sie Ihre MediaTailor CDN-Integration in der Produktion einsetzen. Diese Testverfahren helfen dabei, Konfigurationsprobleme, Leistungsprobleme und Kompatibilitätsprobleme auf verschiedenen Geräten und Plattformen zu identifizieren.

Grundlegende Validierung der Integration

Führen Sie diese grundlegenden Tests durch, um sicherzustellen, dass Ihre CDN-Integration ordnungsgemäß funktioniert:

1. Versand von Testmanifesten:

- Fordern Sie ein Manifest über Ihr CDN an und stellen Sie sicher, dass es eine gültige Antwort zurückgibt
- Stellen Sie sicher, dass das Manifest die erwarteten Inhalte und Anzeigeneinfügapunkte enthält
- Vergewissern Sie sich, dass das Manifest Ihre CDN-Domain und nicht den Ursprung verwendet

- Überprüfen Sie die Manifestsyntax mit den HLS- oder DASH-Validierungstools
2. Überprüfen Sie das Umschreiben der URL:
 - Vergewissern Sie sich, dass das Inhaltssegment URLs in den Manifesten auf Ihre CDN-Domain verweist
 - Stellen Sie sicher, dass das Anzeigensegment URLs auf Ihre CDN-Domain verweist
 - Stellen Sie sicher, dass alle URLs Probleme korrekt gelöst wurden
 3. Wiedergabe von Inhalten testen:
 - Spielen Sie Inhalte über einen Videoplayer ab und überprüfen Sie die reibungslose Wiedergabe
 - Stellen Sie sicher, dass sowohl Inhalte als auch Werbung ohne Unterbrechung abgespielt werden
 - Prüfen Sie, ob die Übergänge zwischen Inhalten und Anzeigen korrekt sind
 - Testen Sie die Such- und Bereinigungsfunktionen
 4. CDN-Routing validieren:
 - Überwachen Sie die CDN-Zugriffsprotokolle, um sicherzustellen, dass Anfragen korrekt weitergeleitet werden
 - Stellen Sie sicher, dass die hit/miss Cachemuster den Erwartungen entsprechen
 - Stellen Sie sicher, dass ursprüngliche Anfragen nur bei Cache-Fehlern auftreten

Fortgeschrittene Testverfahren

Führen Sie diese zusätzlichen Tests für eine umfassende Validierung durch:

1. Plattformübergreifende Kompatibilitätstests:
 - Testen Sie auf mehreren Geräten (Desktop, Handy, Tablet, Smart-TV)
 - Überprüfen Sie die Kompatibilität zwischen verschiedenen Browsern
 - Testen Sie mit verschiedenen Videoplayern (HLS.js, Video.js, native Player)
 - Validieren Sie auf verschiedenen Betriebssystemen
2. Leistungstests:
 - Messen Sie die Antwortzeiten für Manifestanfragen (Ziel: <100 ms für zwischengespeicherte Anfragen)
 - Testen Sie die Leistung beim Laden von Segmenten bei verschiedenen Bitraten
 - Stellen Sie sicher, dass die Startzeit den Leistungszielen entspricht

- Testen Sie unter verschiedenen Netzwerkbedingungen
3. Validierung der Anzeigenverfolgung:
- Stellen Sie sicher, dass die Beacons zur Anzeigenverfolgung korrekt ausgelöst werden
 - Überprüfen Sie die Richtigkeit der Anzeigenanalysedaten
 - Erfassung der Testeindrücke und Nachverfolgung des Abschlusses
 - Überprüfen Sie die Click-Through-Funktionalität
4. Testen der Fehlerbedingungen:
- Testverhalten, wenn Origin vorübergehend nicht verfügbar ist
 - Stellen Sie sicher, dass falsch formatierte Anfragen ordnungsgemäß bearbeitet werden
 - Testen Sie CDN-Failover-Szenarien
 - Überprüfen Sie die Klarheit und Nützlichkeit von Fehlermeldungen

Erstellen Sie eine Testumgebung

Richten Sie für eine umfassende Validierung eine Testumgebung ein, die Ihre Produktionskonfiguration widerspiegelt:

1. Richten Sie separate CDN-Distributionen für Tests ein:
 - Erstellen Sie Test-CDN-Distributionen mit demselben Cache-Verhalten wie in der Produktionsumgebung
 - Konfigurieren Sie Testursprünge, die Ihrem Produktions-Setup entsprechen
 - Verwenden Sie separate Domainnamen, um Konflikte mit dem Produktionsdatenverkehr zu vermeiden
2. MediaTailor Testkonfigurationen erstellen:
 - Richten Sie Testwiedergabekonfigurationen mit denselben Einstellungen wie in der Produktion ein
 - Konfigurieren Sie Test- und Entscheidungsserver-Endpunkte
 - Verwenden Sie Testanzeigeninhalte, die Ihren Produktionsanzeigenformaten entsprechen
3. Implementieren Sie systematische Testprozesse:
 - Erstellen Sie Test-Checklisten für Konfigurationsänderungen
 - Dokumentieren Sie die Testverfahren für Ihr Team

Testen Sie in mehreren Szenarien

Validieren Sie Ihre Integration für verschiedene Szenarien und Bedingungen, um eine umfassende Abdeckung sicherzustellen:

1. Testen Sie mit mehreren Spielertypen und Geräten:

- Testen Sie mit verschiedenen Videoplayern (Internet, Handy, verbundenes Fernsehen)
- Validieren Sie für verschiedene Betriebssysteme und Browser
- Testen Sie unter verschiedenen Netzwerkbedingungen und Verbindungsgeschwindigkeiten

2. Erstellen Sie automatisierte Testskripts:

- Automatisieren Sie die Validierung von Manifestanfragen
- Erstellen Sie Skripts, um Szenarien zum Einfügen von Anzeigen zu testen
- Implementieren Sie Leistungstests für Szenarien mit hohem Besucheraufkommen

3. Überprüfen Sie die Ausrichtung und Personalisierung von Anzeigen:

- Testen Sie mit verschiedenen Benutzerprofilen und Targeting-Parametern
- Validieren Sie die Ad Decision Server-Integration
- Testen Sie Fallback-Szenarien, wenn Anzeigen nicht verfügbar sind

Tools und Techniken zum Testen

Verwenden Sie diese Tools und Techniken für effektives Testen:

Tools für Browser-Entwickler

Verwenden Sie die Registerkarte Netzwerk, um HTTP-Anfragen und -Antworten zu überprüfen

Überwachen Sie die Konsole JavaScript auf Fehler und Warnungen

Überprüfen Sie die Answerheader und das Caching-Verhalten

Überprüfen Sie die Zeitinformationen für die Leistungsanalyse

Testen über die Befehlszeile

Verwenden Sie curl, um bestimmte Header zu testen URLs und zu überprüfen:

```
curl -I "https://your-cdn-domain.com/path/to/manifest.m3u8"
```

Verwenden Sie wget für Download-Tests und Timing-Analysen

Verwenden Sie Tools wie httpie für besser lesbare HTTP-Tests

Testen von Videoplayern

Testen Sie mit Implementierungen mit mehreren Playern

Verwenden Sie die Debug-Modi von Spielern, um das interne Verhalten zu überprüfen

Überwachen Sie Spielerereignisse und Fehlerrückrufe

Überprüfen Sie das Verhalten beim Umschalten der adaptiven Bitrate

CDN-Analyse und Überwachung

Überwachen Sie CDN-Metriken in Echtzeit während des Tests

Überprüfen Sie die Zugriffsprotokolle auf Anforderungsmuster

Verwenden Sie CDN-spezifische Testtools, sofern verfügbar

Richten Sie temporäre Benachrichtigungen für die Testvalidierung ein

Weitere umfassende Testmethoden und systematische Validierungsansätze finden Sie unter [Testen und Validieren für CDN und Integrationen MediaTailor](#).

Beheben Sie CDN-Manifest-404-Fehler für MediaTailor

AWS Elemental MediaTailor Manifeste 404-Fehler im Content Delivery Network (CDN) sind ein häufiges Integrationsproblem, das den Start der Wiedergabe verhindert. In diesem Abschnitt finden Sie Informationen step-by-step zur Behebung von Fehlern bei der Übermittlung von Manifesten.

Multivariante Playlisten-, Medien-Playlisten- oder MPD-Anfragen geben 404-Fehler zurück

Schnelle Lösung (versuchen Sie es zuerst):

1. Stellen Sie sicher, dass der MediaTailor Konfigurationsname in Ihrer URL exakt übereinstimmt (Groß- und Kleinschreibung beachten)
2. Testen Sie die Manifest-URL direkt MediaTailor ohne CDN: `curl -v "https://your-emt-endpoint.mediatailor.region.amazonaws.com/v1/master/hls/config-name/master.m3u8"`
3. Wenn der direkte Test funktioniert, überprüfen Sie die CDN-Routing-Regeln für Manifestanfragen

Wenn Quick Fix nicht funktioniert:

Symptome: Spieler können die Wiedergabe nicht starten. Manifestanfragen geben HTTP-404-Fehler in den CDN-Protokollen zurück.

Beispiele für Fehlermeldungen:

- Browser-Konsole: "Failed to load resource: the server responded with a status of 404 (Not Found)"
- Spielerfehler: "MANIFEST_LOAD_ERROR" oder "NETWORK_ERROR"
- CDN-Protokolle: GET /v1/master/hls/example-config/master.m3u8 404

Auflösung

Vergewissern Sie sich, dass Ihre CDN-Routingregeln korrekt konfiguriert sind, um Anfragen für multivariante Playlisten, Medien-Playlisten und MPD-Anfragen weiterzuleiten. MediaTailor

Vergewissern Sie sich, dass die MediaTailor Konfiguration vorhanden und korrekt eingerichtet ist.

Stellen Sie sicher, dass Ihre CDN-Verhaltensmuster den erwarteten Manifest-Anforderungspfaden entsprechen (z. B. *.m3u8, *.mpd).

Diagnostizieren Sie Probleme und Fehler bei der CDN-Manifestzustellung für MediaTailor

AWS Elemental MediaTailor Probleme bei der Übermittlung von CDN-Manifestdateien (Content Delivery Network) können die korrekte Einfügung und Wiedergabe von Anzeigen verhindern. Wenn Zuschauer falsche oder inkonsistente Anzeigen in multivarianten Playlisten oder Medien-Playlisten erhalten, oder: MPDs

1. Suchen Sie nach zwischengespeicherten Manifesten:

- Stellen Sie sicher, dass die TTL-Einstellungen für alle multivarianten Playlisten, Medien-Playlisten und MPD-Pfade auf 0 gesetzt sind
- Vergewissere dich, dass dein CDN keine multivarianten Playlisten, Medien-Playlisten oder trotz TTL-Einstellungen zwischenspeichert MPDs
- Überprüfe die CDN-Logs auf den Cache-Status — Manifestanfragen sollten angezeigt werden oder nicht Miss RefreshHit Hit

2. Überprüfen Sie die CDN-Routing-Konfiguration:

- Stellen Sie sicher, dass Manifestanfragen an MediaTailor Endpunkte weitergeleitet, nicht zwischengespeichert oder vom Ursprung aus bedient werden
- Stellen Sie sicher, dass die CDN-Verhaltensmuster korrekt mit den Manifestpfaden (*.m3u8, *.mpd) übereinstimmen
- Stellen Sie sicher, dass die Abfrageparameter zur Personalisierung weitergeleitet werden MediaTailor
- Testen Sie das Manifest URLs direkt MediaTailor , um CDN- und Dienstprobleme zu isolieren

3. Überprüfen Sie die Konfiguration der Header-Weiterleitung:

- Stellen Sie sicher, dass die erforderlichen Header weitergeleitet werden (siehe [Erforderliche Header für MediaTailor die CDN-Integration](#))
- Stellen Sie sicher, dass der User-Agent-Header für die gerätespezifische Anzeigenausrichtung weitergeleitet wurde
- Vergewissern Sie sich, dass der X-Forwarded-For Header für Geo-Targeting weitergeleitet wurde
- Stellen Sie sicher, dass der Accept-Encoding-Header zur Unterstützung der Komprimierung weitergeleitet wird

4. Überprüfen Sie den Inhalt und die Struktur des Manifests:

- Stellen Sie sicher, dass die Manifeste die erwarteten Markierungen für die Anzeigeneinfügung enthalten (EXT-X-CUE-OUT/IN für HLS)
- Vergewissern Sie sich, dass das Segment URLs in Manifesten Ihre CDN-Domain und nicht Ihre Ursprungsdomänen verwendet
- Vergewissern Sie sich, dass Anzeigensegmente ordnungsgemäß eingefügt wurden und dass darauf zugegriffen werden
- Testen Sie die Manifest-Syntax mithilfe von Validierungstools (ffprobe für HLS, mp4box für DASH)

5. Testen Sie verschiedene Szenarien:

- Testen Sie mit einer anderen Sitzung IDs , um sicherzustellen, dass die Personalisierung funktioniert
- Testen Sie von verschiedenen geografischen Standorten aus, um das Geo-Targeting zu überprüfen
- Testen Sie mit verschiedenen User-Agent-Zeichenfolgen, um die Geräteausrichtung zu überprüfen
- Vergleichen Sie manifeste Antworten mit und ohne CDN, um Unterschiede zu ermitteln

Zusätzliche Ressourcen zur Problembehandlung:

- Eine ausführliche CDN-Caching-Konfiguration finden Sie unter [Caching-Optimierung für CDN und Integrationen MediaTailor](#)
- Eine umfassende Einrichtung des CDN-Routings finden Sie unter [Richten Sie das CDN-Routing-Verhalten ein für MediaTailor](#)
- Die Anforderungen für die Header-Weiterleitung finden Sie unter [Erforderliche Header für MediaTailor die CDN-Integration](#)
- Informationen zur Protokollanalyse und zu Fehlercodes finden Sie unter [Analyse des CDN-Integrationsprotokolls und Fehlercode-Referenz für MediaTailor](#)
- Informationen zu Testverfahren und Validierung finden Sie unter [Testen und Validieren für CDN und Integrationen MediaTailor](#)

Erfolgskriterien: Wenn das Problem gelöst ist, sollten die Spieler die Wiedergabe normal starten und die Werbung sollte wie erwartet erscheinen. Manifestanfragen sollten HTTP-200-Statuscodes in CDN-Protokollen zurückgeben, und Manifeste sollten ordnungsgemäß personalisierte Anzeigeninhalte enthalten.

Beheben Sie Probleme mit der Bereitstellung und dem Laden von CDN-Segmenten für MediaTailor

AWS Elemental MediaTailor Probleme bei der Bereitstellung von CDN-Segmenten (Content Delivery Network) können zu Puffer- und Wiedergabeunterbrechungen führen. Wenn Spieler keine Segmente laden können oder es zu einer Pufferung kommt:

1. Überprüfe die CDN-Routing-Regeln:

- Stellen Sie sicher, dass Inhaltssegmente an den richtigen Ursprung weitergeleitet werden
- Vergewissern Sie sich, dass Anzeigensegmente an den richtigen MediaTailor Anzeigenspeicherort weitergeleitet werden
- Vergewissern Sie sich, dass die Dateierweiterungen der Segmente Ihren CDN-Verhaltensmustern entsprechen
- Stellen Sie sicher, dass das Segment URLs in den Manifesten die richtige CDN-Domain verwendet
- Ausführliche Anweisungen zum Einrichten von Routing- und Verhaltenspfadmustern finden Sie unter [Richten Sie das CDN-Routing-Verhalten ein für MediaTailor](#)

2. Überprüfen Sie die CORS-Konfiguration:

- Stellen Sie bei Web-Playern sicher, dass Ihr CDN die CORS-Header durchläuft oder diese richtig einstellt
- Testen Sie mit Browser-Entwicklertools, um CORS-bezogene Fehler zu identifizieren
- Stellen Sie sicher, dass Preflight-OPTIONS-Anfragen korrekt verarbeitet werden

3. Testen Sie die Zugänglichkeit und Leistung von Segmenten:

- Testen Sie einzelne Segmente URLs direkt, um sicherzustellen, dass sie zugänglich sind
- Überprüfen Sie die Reaktionszeiten der Segmente und identifizieren Sie Leistungsengpässe
- Stellen Sie sicher, dass die Größe der Segmentdateien für Ihre Bandbreitenziele geeignet ist
- Testen Sie das Laden von Segmenten von verschiedenen geografischen Standorten

4. Überprüfen Sie das CDN-Caching-Verhalten für Segmente:

- Stellen Sie sicher, dass Inhaltssegmente über angemessene TTL-Einstellungen verfügen (in der Regel länger als Manifeste)
- Vergewissern Sie sich, dass die Anzeigensegmente entsprechend den Personalisierungsanforderungen ordnungsgemäß zwischengespeichert werden
- Überwachen Sie die Cache-Trefferquoten sowohl für Inhalte als auch für Anzeigensegmente
- Stellen Sie sicher, dass die Cache-Schlüssel keine unnötigen Parameter enthalten, die die Cache-Effizienz beeinträchtigen

5. Überprüfen Sie die Konnektivität und den Zustand des Ursprungsservers:

- Stellen Sie sicher, dass die Ursprungsserver korrekt auf Segmentanfragen reagieren
- Überprüfen Sie die Kapazität und die Antwortzeiten des Originalservers unter Last
- Stellen Sie sicher, dass auf den Originalservern die erwarteten Segmentdateien verfügbar sind
- Testen Sie Origin-Failover-Szenarien, wenn mehrere Ursprünge konfiguriert sind

6. Beheben Sie spezifische Probleme mit Anzeigensegmenten:

- Stellen Sie sicher, dass die Anzeigensegmente ordnungsgemäß transkodiert sind und verfügbar sind MediaTailor
- Vergewissern Sie sich, dass die URLs Anzeigensegmente in den Manifesten korrekt generiert wurden
- Testen Sie das Laden von Anzeigensegmenten mit verschiedenen Parametern für die Anzeigenausrichtung
- Achten Sie auf Verzögerungen bei der Transcodierung von Anzeigen, die zur Nichtverfügbarkeit von Segmenten führen könnten

7. Überprüfen Sie die Kompatibilität und das Verhalten der Spieler:

- Testen Sie das Laden von Segmenten mit verschiedenen Spielertypen und -versionen
- Überprüfen Sie die Puffereinstellungen des Players und die Muster für die Segmentanfragen
- Überprüfen Sie die Fehlerbehandlung des Spielers für fehlgeschlagene Segmentanfragen
- Testen Sie die adaptive Bitratenumschaltung und die Logik für die Segmentauswahl

Zusätzliche Ressourcen zur Fehlerbehebung:

- Informationen zur Konfiguration von CDN-Routing und Verhalten finden Sie unter [Richten Sie das CDN-Routing-Verhalten ein für MediaTailor](#)
- Informationen zur Optimierung des CDN-Cachings finden Sie unter [Caching-Optimierung für CDN und Integrationen MediaTailor](#)
- Anleitungen zur CORS-Konfiguration finden Sie unter [Bewährte Sicherheitsmethoden für die CDN-Integration für MediaTailor](#)
- Informationen zur Leistungsüberwachung und -analyse finden Sie unter [Überwachen Sie den Betrieb und die Leistung von MediaTailor CDN](#)
- Umfassende Testverfahren finden Sie unter [Testen und Validieren für CDN und Integrationen MediaTailor](#)
- Informationen zur Protokollanalyse und Fehlerdiagnose finden Sie unter [Analyse des CDN-Integrationsprotokolls und Fehlercode-Referenz für MediaTailor](#)

Erfolgskriterien: Wenn das Problem gelöst ist, sollten die Spieler die Segmente reibungslos und ohne Unterbrechungen beim Puffern laden. Segmentanfragen sollten HTTP-200-Statuscodes mit angemessenen Antwortzeiten zurückgeben. Sowohl Inhalt als auch Anzeigensegmente sollten zugänglich sein und ordnungsgemäß zwischengespeichert werden können.

Behebt Probleme mit der Verwaltung und Nachverfolgung von CDN-Sitzungen für MediaTailor

AWS Elemental MediaTailor Das Content Delivery Network (CDN) -Sitzungsmanagement ist entscheidend für die korrekte Personalisierung und Nachverfolgung von Anzeigen. Wenn Sie auf sitzungsbezogene Fehler oder inkonsistentes Verhalten bei Anfragen stoßen, gehen Sie wie folgt vor:

1. Prüfen Sie, ob die Sitzungs-IDs konsistent sind:

- Stellen Sie sicher, dass Ihr Player bei allen Anfragen für eine einzelne Wiedergabe-Sitzung dieselbe Sitzungs-ID verwendet
- Überprüfe in den CDN-Protokollen, ob IDs die Sitzung korrekt weitergeleitet wurde
- Stellen Sie sicher, dass IDs die Sitzungen in den Abfrageparametern ordnungsgemäß URL-kodiert sind
- Verwenden Sie CloudWatch Protokolle, um die Konsistenz der Sitzungs-IDs bei allen Anfragen zu überprüfen (siehe Validierungsschritte unten)

2. Überprüfen Sie die Sitzungsinitialisierung:

- Vergewissern Sie sich, dass die erste Manifestanforderung erfolgreich eine Sitzung erstellt
- Prüfen Sie, ob die Sitzungsparameter ordnungsgemäß weitergeleitet wurden (z. `B.aws.sessionId`)
- Überprüfen Sie die Sitzungsinitialisierung mithilfe von Debug-Protokollen (siehe Einrichtung des Debug-Protokolls unten)

3. Aktivieren Sie die Debug-Protokollierung für eine detaillierte Sitzungsbehandlung:

- Für serverseitige Berichte: Fügen Sie Ihrer `?aws.logMode=DEBUG` Wiedergabeanforderung Folgendes hinzu:

```
GET <mediatailorURL>/v1/master/<hashed-account-id>/<origin-id>/<asset-id>?
aws.logMode=DEBUG
```

- Für clientseitige Berichte: Fügen Sie den Text Ihrer Anfrage zur `"logMode": "DEBUG"` Initialisierung Ihrer Sitzung hinzu
- Wichtig: Bei dem Wert wird zwischen Groß- und DEBUG Kleinschreibung unterschieden
- Maximal 10 aktive Debug-Sitzungen sind gleichzeitig zulässig

4. Verwenden Sie CloudWatch Protokollabfragen, um das Sitzungsverhalten zu überprüfen:

- Stellen Sie sicher, dass die Debug-Sitzung aktiv ist:

```
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
```

- Alle Ereignisse für eine Sitzung anzeigen:

```
fields @timestamp, @message, eventType, mediaTailorPath
```

```
| filter sessionId = "your-session-id-here"  
| sort @timestamp asc
```

- Überprüfen Sie die Manifestgenerierung für eine Sitzung:

```
fields @timestamp, responseBody, @message  
| filter mediaTailorPath like "/v1/master/" and eventType = "GENERATED_MANIFEST"  
and sessionId = "your-session-id-here"
```

5. Testen Sie die Weiterleitung von Sitzungsparametern über CDN:

- Testen Sie Manifestanfragen mit Sitzungsparametern direkt gegen MediaTailor (unter Umgehung von CDN)
- Vergleichen Sie das Sitzungsverhalten mit und ohne CDN, um Weiterleitungsprobleme zu identifizieren
- Stellen Sie sicher, dass die Konfiguration für die Weiterleitung von CDN-Abfrageparametern sitzungsbezogene Parameter enthält
- Stellen Sie sicher, dass CDN keine Antworten zwischenspeichert, die sitzungsspezifisch sein sollten

Häufige Sitzungsfehlermeldungen:

- `ConflictException(HTTP 409)` — Mehrere gleichzeitige Playlist-Anfragen für dieselbe Sitzung. Lösung: Stellen Sie sicher, dass Ihr Player Playlisten einzeln gemäß der HLS-Spezifikation anfordert
- `NotFoundException(HTTP 404)` — Die Sitzung ist nicht verfügbar oder die Konfiguration ist nicht vorhanden. Lösung: Überprüfen Sie die Gültigkeit Ihrer Konfiguration und initialisieren Sie die Sitzung erneut
- `BadRequestException(HTTP 400)` — Ungültige Sitzungs-ID oder falsch formatierte Anfrage. Lösung: Überprüfen Sie die Formatierung der Anfrage und die Gültigkeit der Sitzungs-ID

Zusätzliche Ressourcen zur Problembehandlung:

- Eine vollständige Einrichtung der Debug-Protokollierung und eine Feldreferenz finden Sie unter [AWS Elemental MediaTailor Debug-Protokolle werden generiert](#)
- Beispiele für CloudWatch Protokollabfragen und Protokollanalysen finden Sie unter [AWS Elemental MediaTailor Logs direkt in Amazon CloudWatch Logs schreiben](#)

- Informationen zur Konfiguration der Weiterleitung von CDN-Abfrageparametern finden Sie unter [Richten Sie das CDN-Routing-Verhalten ein für MediaTailor](#)
- Eine umfassende Referenz zu Fehlercodes finden Sie unter [Problembeseitigung bei der Wiedergabe von MediaTailor](#)

Erfolgskriterien: Wenn das Problem behoben ist, sollten die Sitzungen korrekt initialisiert werden, eine konsistente Sitzung IDs über alle Anfragen hinweg aufrechterhalten und die Debug-Protokolle sollten die richtigen SESSION_INITIALIZED Ereignisse und die Manifestgenerierung ohne Fehler anzeigen.

Beheben Sie Probleme beim Timing und bei der Synchronisation von CDN-Werbeunterbrechungen für MediaTailor

AWS Elemental MediaTailor Das Timing von Werbeunterbrechungen im Content Delivery Network (CDN) muss exakt mit den Inhaltsmarkern synchronisiert werden. Wenn Anzeigen zu falschen Zeiten erscheinen oder das Timing der Werbeunterbrechungen inkonsistent ist:

1. Überprüfen Sie die Markierungen für Werbeunterbrechungen im Inhalt:

- Vergewissern Sie sich, dass die SCTE-35-Markierungen korrekt in Ihren Originalinhalten platziert sind
- Stellen Sie sicher, dass die Dauer der Werbepause der tatsächlichen Länge des Anzeigeninhalts entspricht
- Vergewissern Sie sich, dass der Zeitpunkt der Werbeunterbrechung den Inhaltsgrenzen entspricht
- Überprüfen Sie das SCTE-35-Markierungsformat und die Genauigkeit des Timings in Ihren Origin-Manifesten
- Testen Sie Werbeblock-Markierungen mit unterschiedlichen Inhaltstypen (Live oder VOD)

2. Prüfen Sie, wie sich das CDN-Caching auf das Timing auswirkt:

- Stellen Sie sicher, dass die Manifest-TTL auf 0 gesetzt ist, um Zeitabweichungen zu vermeiden
- Stellen Sie sicher, dass zeitkritische Parameter nicht zwischengespeichert werden
- Suchen Sie nach Problemen mit der Uhrsynchronisierung zwischen der Inhaltsquelle MediaTailor, und dem CDN
- Achten Sie bei Livestreams mit langer Laufzeit auf Zeitabweichungen
- [Überprüfen Sie die CDN-Edge-Server-Zeitsynchronisierung mit NTP](#)

3. Überprüfen Sie die Implementierung des SCTE-35-Markers:

- Überprüfen Sie, ob die EXT-X-DATERANGE Tags die richtigen SCTE35 OUT- und DURATION-Spezifikationen enthalten
- Achten Sie bei der Verwendung eines expliziten Cue-In-Timings auf gepaarte SCTE35 OUT- und SCTE35 -IN-Markierungen
- Stellen Sie sicher, dass die Zeitstempel für START-DATE mit dem tatsächlichen Timing der Inhalte übereinstimmen
- Testen Sie verschiedene SCTE-35-Markerformate (dauerbasierte oder gepaarte Markierungen)

4. Testen Sie das Timing von Werbeunterbrechungen in verschiedenen Szenarien:

- Vergleichen Sie das Timing von Werbeunterbrechungen mit direkten MediaTailor Anfragen und CDN-Anfragen
- Testen Sie die Konsistenz des Timings an verschiedenen CDN-Edge-Standorten
- Überprüfe das Timing von Werbeunterbrechungen anhand verschiedener Spielertypen und Pufferverhalten
- Überwachen Sie die Genauigkeit des Timings in Zeiten mit hohem Besucheraufkommen

5. Debuggen Sie Timing-Probleme mithilfe von Protokollen und Überwachung:

- Aktivieren Sie die Debug-Protokollierung, um den Zeitpunkt der Verarbeitung von Werbeunterbrechungen zu verfolgen
- Überwachen Sie CloudWatch Metriken für Zeitmuster bei der Anzeigeneinblendung
- Suchen Sie in den CDN-Protokollen nach zeitbezogenen Anforderungsmustern
- Verwenden Sie Player-Debugging-Tools, um das Timing von Werbeunterbrechungen aus Kundensicht zu überprüfen

Erwartete zeitliche Toleranzen:

- Das Timing von Werbeunterbrechungen sollte mit den SCTE-35-Markierungen in Ihren Inhalten übereinstimmen
- Die Dauer der Anzeige sollte der Dauer entsprechen, die in der Antwort des Anzeigenentscheidungsservers angegeben ist
- Die Uhrsynchronisierung zwischen Inhaltsquelle, MediaTailor, und CDN sollte innerhalb von 1 Sekunde erfolgen
- Das Timing der SCTE-35-Markierung sollte mit einer Genauigkeit von 100 ms vom tatsächlichen

Inhalts-Timing abweichen

Zusätzliche Ressourcen zur Fehlerbehebung:

- Informationen zum Format und zur Implementierung des SCTE-35-Markers finden Sie unter [Integration einer Inhaltsquelle für die MediaTailor Anzeigeneinfügung](#)
- Informationen zur Einrichtung der Debug-Protokollierung und zur Timing-Analyse finden Sie unter [AWS Elemental MediaTailor Debug-Protokolle werden generiert](#)
- Informationen zur Konfiguration des CDN-Cachings und zu den Auswirkungen auf das Timing finden Sie unter [Caching-Optimierung für CDN und Integrationen MediaTailor](#)
- Umfassende Testverfahren, einschließlich der zeitlichen Validierung, finden Sie unter [Testen und Validieren für CDN und Integrationen MediaTailor](#)
- Informationen zur Überwachung des Zeitpunkts und der Leistung der Anzeigenschaltung finden Sie unter [Überwachen Sie den Betrieb und die Leistung von MediaTailor CDN](#)

Erfolgskriterien: Wenn das Problem behoben ist, sollten Werbeunterbrechungen genau zu den Zeiten erscheinen, die durch die SCTE-35-Markierungen angegeben sind, und zwar mit einheitlichem Timing für alle CDN-Edge-Standorte und Spielertypen. In den Debug-Protokollen sollte der genaue Zeitpunkt der Verarbeitung von Werbeunterbrechungen ohne Abweichungen oder Synchronisationsfehler angegeben werden.

Optimieren Sie die CDN-Leistung und beheben Sie Latenzprobleme für MediaTailor

AWS Elemental MediaTailor Die Leistung der CDN-Integration (Content Delivery Network) wirkt sich direkt auf das Zuschauererlebnis und die Qualität der Anzeigenbereitstellung aus. Wenn Sie langsame Reaktionszeiten oder Leistungseinbußen feststellen:

Techniken zur Leistungsmessung

Bevor Sie Leistungsprobleme beheben, sollten Sie grundlegende Messungen und eine kontinuierliche Überwachung vornehmen:

1. Messen Sie die wichtigsten Leistungskennzahlen:

- Antwortzeiten: Manifestanfragen sollten innerhalb von 200 ms abgeschlossen sein, segmentierte Anfragen innerhalb von 100 ms
- Cache-Trefferquoten: Inhaltssegmente > 95%, Anzeigensegmente > 90%

- Volumen der ursprünglichen Anfragen: Sollte weniger als 5% der gesamten Anfragen betragen, wenn der Cache optimiert ist
- Zeit bis zum ersten Frame: Die erste Wiedergabe sollte innerhalb von 2-3 Sekunden beginnen

2. Verwenden Sie Tools zur Leistungsmessung:

- CDN-Analyse-Dashboards: Überwachen Sie die Cache-Leistung, die Antwortzeiten und die Fehlerraten
- CloudWatch Metriken: Verfolgen Sie MediaTailor Servicemetriken, einschließlich Latency GetManifest
- Tools für Browser-Entwickler: Messen Sie die Leistung und das Netzwerk-Timing auf Kundenseite
- Befehlszeilentools: Verwenden Sie Curl mit Timing-Optionen, um bestimmte Anfragen zu messen

3. Implementieren Sie eine kontinuierliche Überwachung:

- Richten Sie automatische Leistungswarnungen für verschlechterte Reaktionszeiten ein
- Überwachen Sie die Leistung in verschiedenen geografischen Regionen
- Verfolgen Sie die Leistung in Zeiten mit hohem Verkehrsaufkommen
- Vergleichen Sie Leistungskennzahlen vor und nach Konfigurationsänderungen

Ressourcen zur Leistungsmessung:

- Informationen zur umfassenden Einrichtung der Leistungsüberwachung finden Sie unter [Überwachen Sie den Betrieb und die Leistung von MediaTailor CDN](#)
- Informationen zu den Verfahren für Leistungstests finden Sie unter [Testen und Validieren für CDN und Integrationen MediaTailor](#)
- Informationen zu CloudWatch Metriken und Überwachung finden Sie unter [Überwachung AWS Elemental MediaTailor mit CloudWatch Amazon-Metriken](#)

Leistungsprobleme beim CDN-Cache

Probleme mit der Cache-Leistung gehören zu den häufigsten Problemen bei der CDN-Integration. Diese Probleme betreffen alle MediaTailor Implementierungen und können sich erheblich auf das Nutzererlebnis und die Kosten auswirken.

Niedrige Cache-Trefferquote

Symptome: Hohes Volumen der ursprünglichen Anfragen, erhöhte Latenz, höhere Bandbreitenkosten, schlechtes Zuschauererlebnis

Zielwerte:

- Inhaltssegmente: Cache-Trefferquote von 95% oder höher
- Anzeigensegmente: Cache-Trefferquote von mindestens 90%
- Manifeste: Variiert je nach Implementierung (personalisierte Manifeste sollten nicht zwischengespeichert werden)

Häufige Ursachen:

- Falsche TTL-Einstellungen für verschiedene Inhaltstypen
- Die Cache-Schlüsselkonfiguration enthält unnötige Abfrageparameter
- Die Cache-Control-Header von Origin sind nicht richtig konfiguriert
- Häufige Cache-Invalidierungen oder -Löschungen
- Probleme mit der geografischen Verteilung (Inhalt wird nicht an Edge-Standorten zwischengespeichert)

Lösungen:

1. Überprüfen und optimieren Sie die TTL-Einstellungen:

- Inhaltssegmente: Stellen Sie TTL so ein, dass es der Segmentdauer oder länger entspricht
- Anzeigensegmente: Lege TTL für wiederverwendbare Anzeigen auf 24 Stunden oder länger fest
- Statische Inhalte: Lege TTL auf 24 Stunden oder länger fest

Umfassende TTL-Empfehlungen und Strategien zur Caching-Optimierung finden Sie unter.

[Caching-Optimierung für CDN und Integrationen MediaTailor](#)

2. Optimieren Sie die Cache-Schlüsselkonfiguration:

- Entfernen Sie unnötige Abfrageparameter aus den Cache-Schlüsseln
- Stellen Sie sicher, dass nur Parameter enthalten sind, die sich auf den Inhalt auswirken
- Normalisieren Sie die Reihenfolge der Parameter und achten Sie auf Groß-

3. Stellen Sie sicher, dass die Origin-Cache-Control-Header richtig gesetzt sind

4. Implementieren Sie Origin Shield (oder eine gleichwertige CDN-Funktionalität) für Implementierungen mit hohem Datenverkehr. Die Origin Shield-Funktionalität ist in allen

gängigen CDNs Programmen verfügbar, kann aber unterschiedliche Namen haben (wie CloudFront Origin Shield, Fastly Shield, Cloudflare Argo Tiered Cache). Wenn Ihr CDN diese Funktionalität nicht bietet, kann sie aktiviert werden, MediaTailor wenn Sie sich an den [AWS-Support](#) wenden.

5. Überprüfen Sie die Strategien zur Cache-Invalidierung und reduzieren Sie unnötige Löschvorgänge

Schritte zur Validierung:

1. Überwachen Sie die Cache-Trefferquoten mithilfe von CDN-Analyse-Dashboards
2. Testen Sie spezifisch URLs mit Curl, um die Cache-Header zu überprüfen
3. Vergleichen Sie das Volumen der ursprünglichen Anfragen vor und nach den Änderungen

Hohes Volumen der ursprünglichen Anfragen

Symptome: Unerwartet hohe Anzahl von Anfragen, die den MediaTailor Ursprung erreichen, erhöhte Auslastung des Ursprungsservers, höhere Kosten

Erwartetes Muster: Origin-Anfragen sollten weniger als 5% der gesamten Viewer-Anfragen ausmachen, wenn die Cache-Trefferquoten optimal sind

Häufige Ursachen:

- Fehlschläge im Cache sind auf niedrige TTL-Werte zurückzuführen
- Fragmentierung des Cache-Schlüssels (zu viele eindeutige Cache-Schlüssel)
- Geografischer Traffic nimmt in Regionen ohne zwischengespeicherten Inhalt stark zu
- Häufige Cache-Invalidierungen

Lösungen:

1. Analysieren Sie Anforderungsmuster, um die Ursachen für Cache-Fehler zu identifizieren
2. Optimieren Sie die TTL-Einstellungen auf der Grundlage von Inhaltstyp und Aktualisierungshäufigkeit
3. Implementieren Sie Strategien zur Cache-Erwärmung für neue Inhalte
4. Erwägen Sie die Implementierung von Origin Shield (verfügbar [Origin Shield-Implementierung](#) für alle wichtigen Programme CDNs mit unterschiedlichen Namen — Einzelheiten finden Sie unter)

Alarmschwelle: Richten Sie Warnmeldungen ein, wenn die ursprünglichen Anfragen 10% der Gesamtzahl der Anfragen überschreiten oder sich gegenüber dem Ausgangswert um 50% erhöhen

Behebung häufiger HTTP-Fehler

HTTP-Fehler bei CDN-Integrationen deuten häufig auf Konfigurationsprobleme oder Serviceprobleme hin. Diese Fehlermuster sind in allen MediaTailor Implementierungen konsistent.

Fehler 404 Nicht gefunden

Symptome: Manifest- oder Segmentanfragen geben HTTP 404 zurück, Spieler können Inhalte nicht laden, „MANIFEST_LOAD_ERROR“ in den Spielerprotokollen

Häufige Ursachen:

- Falsche CDN-Ausgangskonfiguration (falsche Endpunkt-URL) MediaTailor
- Fehlende oder falsche Pfadmuster für das Cache-Verhalten
- Probleme beim Umschreiben von URLs in der CDN-Konfiguration
- MediaTailor Fehler beim Konfigurationsnamen oder beim Wiedergabeendpunkt
- Timing-Probleme mit Live-Inhalten (Anforderung future Segmente)

Diagnoseschritte:

1. Testen Sie dieselbe URL direkt mit dem MediaTailor Ursprung (CDN umgehen)
2. Stellen Sie sicher, dass die CDN-Ursprungskonfiguration mit dem Wiedergabeendpunkt übereinstimmt MediaTailor
3. Überprüfen Sie das Verhalten, die Pfadmuster und die Rangfolge des CDN-Cache
4. Einzelheiten zur Weiterleitung von Anfragen finden Sie in den CDN-Zugriffsprotokollen
5. Überprüfen Sie den MediaTailor Konfigurationsnamen und die Regionseinstellungen

Lösungen:

- Korrigieren Sie die CDN-Ursprungskonfiguration so, dass sie dem MediaTailor Wiedergabeendpunkt entspricht
- Aktualisieren Sie die Pfadmuster im Cache, um Anfragen korrekt weiterzuleiten
- Korrigieren Sie gegebenenfalls die Regeln für das Umschreiben von URLs
- Stellen Sie sicher, dass die MediaTailor Konfiguration vorhanden und aktiv ist

403 Verbotene Fehler

Symptome: Anfragen geben HTTP 403 zurück, Nachrichten mit verweigertem Zugriff, Authentifizierungsfehler

Häufige Ursachen:

- Fehlende oder falsche Abfrageparameter erforderlich von MediaTailor
- CDN leitet die erforderlichen Header oder Parameter nicht weiter
- IP-Adressbeschränkungen oder geografische Blockierung
- Probleme mit Authentifizierungstoken (bei Verwendung von signierten Token URLs)

Lösungen:

- Stellen Sie sicher, dass alle erforderlichen Abfrageparameter enthalten und weitergeleitet wurden
- Überprüfen Sie die CDN-Konfiguration für die Header- und Parameterweiterleitung
- Überprüfen Sie die IP-Einschränkungen und die geografischen Einstellungen
- Validieren Sie Authentifizierungstoken und Signaturprozesse

400 Fehler bei schlechten Anfragen

Symptome: Anfragen geben HTTP 400 zurück, falsch formatierte Anforderungsfehler, Fehler bei der Parametervalidierung

Häufige Ursachen:

- Falsch formatierte Abfrageparameter oder Probleme mit der URL-Codierung
- Ungültige Parameterwerte oder Formate
- Fehlende erforderliche Parameter für bestimmte MediaTailor Funktionen
- Die Beschränkungen für die URL-Länge wurden überschritten

Lösungen:

- Überprüfen Sie die Formate der Abfrageparameter und die URL-Kodierung
- Überprüfen Sie die Parameterwerte anhand der MediaTailor API-Anforderungen
- Stellen Sie sicher, dass alle erforderlichen Parameter enthalten sind
- Überprüfen Sie die URL-Länge und ziehen Sie eine Parameteroptimierung in Betracht

5xx Serverfehler

Symptome: Anfragen geben HTTP 500-, 502-, 503- oder 504-Fehler sowie zeitweise auftretende Dienstaussfälle zurück

Häufige Ursachen:

- MediaTailor Serviceprobleme oder Kapazitätsbeschränkungen
- Probleme mit der Konnektivität des CDN-Ursprungs

- Timeout-Probleme aufgrund langsamer Antworten von Origin
- Vorübergehende Verschlechterung des Dienstes

Lösungen:

- Überprüfen Sie den AWS Servicestatus auf dem MediaTailor Service Health Dashboard
- Überprüfen Sie die CDN-Ausgangskonnektivität und die Timeout-Einstellungen
- Implementieren Sie die Wiederholungslogik mit exponentiellem Backoff
- Überwachen Sie Metriken MediaTailor CloudWatch für den Dienststatus
- Wenden Sie sich an AWS den Support, wenn das Problem weiterhin besteht

1. Messen Sie die Ausgangsleistung:

- Testen Sie die Antwortzeiten für Manifestanfragen direkt an MediaTailor (Ziel: <200 ms)
- Messen Sie die CDN-Antwortzeiten für Manifestanfragen (Ziel: <100 ms für Cache-Treffer)
- Überprüfen Sie die Ladezeiten der Segmente sowohl vom Ursprung als auch vom CDN

2. Analysieren Sie die CDN-Leistung:

- Überprüfen Sie die Cache-Trefferquoten für Inhaltssegmente (Ziel: > 80% für beliebte Inhalte)
- Vergewissern Sie sich, dass Origin Shield (oder eine gleichwertige CDN-Funktionalität) in derselben AWS Region wie Ihr Ursprungsland aktiviert und konfiguriert ist. Verschiedene CDNs verwenden unterschiedliche Namen für diese Funktion
- Überwachen Sie die Leistung und die geografische Verteilung des CDN-Edge-Standorts

Leistungsmaßstäbe:

- Überwachen Sie die Reaktionszeiten bei der Manifestgenerierung und vergleichen Sie sie mit Ihrer Ausgangsleistung
- CDN-Cache-Treffer sind deutlich schneller als ursprüngliche Anfragen
- Die ADS-Antwortzeiten sollten nicht zu deutlichen Verzögerungen bei der Generierung führen

Zusätzliche Ressourcen zur Problembehandlung:

- Umfassende Strategien zur Leistungsoptimierung finden Sie unter [Leitfaden zur Leistungsoptimierung für CDN und Integrationen MediaTailor](#)
- Einzelheiten zur Implementierung von Origin Shield finden Sie unter [Origin Shield-Implementierung](#)

- Informationen zur CDN-Caching-Optimierung finden Sie unter [Caching-Optimierung für CDN und Integrationen MediaTailor](#)
- Informationen zur Leistungsüberwachung und zu Kennzahlen finden Sie unter [Überwachen Sie den Betrieb und die Leistung von MediaTailor CDN](#)
- Informationen zu Verfahren für Leistungstests finden Sie unter [Testen und Validieren für CDN und Integrationen MediaTailor](#)

Erfolgskriterien: Wenn das Problem gelöst ist, sollten die Antwortzeiten den Zielvorgaben entsprechen (Manifeste weniger als 200 ms, Segmente weniger als 100 ms), die Cache-Trefferquote sollte bei den meisten Inhaltstypen über 90% liegen und das Volumen der ursprünglichen Anfragen sollte weniger als 5% der gesamten Anfragen betragen. Die Leistung sollte in allen geografischen Regionen und Gerätetypen konsistent sein.

Korrigieren Sie das inkonsistente CDN-Verhalten auf allen Geräten und Plattformen für MediaTailor

AWS Elemental MediaTailor Die Integration eines Content Delivery Network (CDN) sollte für eine konsistente Anzeigenbereitstellung auf allen Geräten und Plattformen sorgen. Wenn sich Anzeigen auf verschiedenen Geräten unterschiedlich verhalten:

1. Sorgen Sie für eine konsistente Header-Weiterleitung für alle CDN-Verhaltensweisen.
 - Stellen Sie sicher, dass User-Agent- und Custom-Targeting-Header konsistent weitergeleitet werden X-Forwarded-For
 - Vergewissern Sie sich, dass die Regeln für die Header-Weiterleitung für alle relevanten Cache-Verhaltensweisen gelten
2. Überprüfen Sie die Player-Kompatibilität mit Ihrer CDN-Konfiguration.
 - Testen Sie mit mehreren Spielertypen (HLS.js, Video.js, native Spieler), um spieterspezifische Probleme zu identifizieren
 - Überprüfe, ob es spieterspezifische Header-Anforderungen oder Unterschiede bei der URL-Handhabung gibt
3. Testen Sie mit mehreren Gerätetypen, um gerätespezifische Probleme zu identifizieren.
 - Beziehen Sie mobile Geräte, Tablets TVs, Smart- und Desktop-Browser in Ihre Tests mit ein
 - Testen Sie verschiedene Betriebssysteme und Browserversionen
 - Stellen Sie sicher, dass die gerätespezifische Anzeigenausrichtung korrekt funktioniert

Wenn Sie diese Schritte zur Fehlerbehebung befolgt haben und weiterhin Unterstützung benötigen, finden Sie weitere Informationen unter. [Holen Sie sich Unterstützung bei der CDN-Integration](#)

Vorbereitung der Fehlerbehebung

Richten Sie Tools und Prozesse ein, um die Fehlerbehebung zu vereinfachen, wenn Probleme mit der CDN-Integration auftreten. Eine proaktive Vorbereitung ermöglicht eine schnellere und effektivere Fehlerbehebung, wenn Probleme auftreten.

Ermöglichen Sie eine umfassende Protokollierung

Detaillierte Protokolle sind für die Diagnose von CDN-Integrationsproblemen unerlässlich. Konfigurieren Sie die Protokollierung so, dass die Informationen erfasst werden, die Sie bei der Fehlerbehebung benötigen.

1. Aktivieren Sie detaillierte CDN-Zugriffsprotokolle:

- Konfigurieren Sie die Protokollierung für alle Cache-Verhaltensweisen, die Anfragen verarbeiten MediaTailor
- Fügen Sie Abfragezeichenfolgen und benutzerdefinierte Header in Protokolleinträge ein
- Richten Sie Tools zur Protokollanalyse ein, um Muster und Anomalien zu identifizieren
- Aktivieren Sie Echtzeitprotokolle für die sofortige Problemerkennung bei Live-Events
- Konfigurieren Sie Richtlinien zur Aufbewahrung von Protokollen, um historische Daten zur Fehlerbehebung beizubehalten

2. MediaTailor Protokollierung konfigurieren:

- Aktivieren Sie die Zugriffsprotokolle für Ihre MediaTailor Konfigurationen
- Richten CloudWatch Sie Protokollgruppen für die zentrale Protokollverwaltung ein
- Konfigurieren Sie Protokollfilter, um Fehlermuster zu identifizieren

3. Richten Sie die Origin-Server-Protokollierung ein:

- Aktivieren Sie detaillierte Zugriffsprotokolle auf Ihren Inhalts-Ursprungsservern
- Nehmen Sie Anforderungsheader und Antwortcodes in die Protokolle auf
- Überwachen Sie die Leistungskennzahlen des Originalservers

Fügen Sie Header für Diagnoseanfragen hinzu

Benutzerdefinierte Header helfen dabei, Anfragen über Ihr CDN zu verfolgen und Routing-Probleme zu identifizieren.

1. CDN-Diagnose-Header konfigurieren:

- Fügen Sie jeder Anfrage eine eindeutige Kennung hinzu (z. B.) X-Request-ID
- Fügen Sie CDN-spezifische Informationen in die Header der Anfrage ein
- Fügen Sie Edge-Standort- oder POP-Informationen (Point of Presence) hinzu, um das geografische Routing zu verfolgen
- Fügen Sie Cache-Status-Header (Hit, Miss, RefreshHit) für die Analyse des Cache-Verhaltens hinzu

2. Fügen Sie Antwort-Header für das Debuggen hinzu:

- Header zur Serveridentifikation einbeziehen
- Fügen Sie Zeitinformationen für die Leistungsanalyse hinzu
- Fügen Sie Cache-Control-Header für Manifestanfragen hinzu

Legen Sie grundlegende Leistungskennzahlen fest

Dokumentieren Sie normale Leistungsbereiche, um Auffälligkeiten bei der Fehlerbehebung schnell zu erkennen:

1. Basiskennzahlen aufzeichnen:

- Cache-Trefferquoten für verschiedene Inhaltstypen
- Perzentile für die Antwortzeit (P50, P95, P99)
- Fehlerraten nach Statuscode
- Volumenmuster nach Tageszeit abfragen

2. Leistungserwartungen dokumentieren:

- Trefferquoten im Ziel-Cache (95% + für Inhalte, mehr als 90% für Anzeigen)
- Akzeptable Antwortzeiten (<100 ms zwischengespeichert, <500 ms Ursprung)
- Maximal zulässige Fehlerraten (< 1% für 4xx, < 0,1% für 5xx)

3. Leistungs-Dashboards erstellen: Richten Sie Überwachungs-Dashboards ein, die aktuelle Kennzahlen im Vergleich zu Basiswerten anzeigen.

Bereiten Sie Tools zur Fehlerbehebung vor

Richten Sie Tools und Zugriffsberechtigungen ein, die für eine effektive Fehlerbehebung erforderlich sind:

1. Befehlszeilentools:

- `curl` zum Testen von HTTP-Anfragen und -Antworten
- `dig` oder `nslookup` zur DNS-Fehlerbehebung
- HLS/DASH-Validierungstools für die Manifestverifizierung
- Tools zur Protokollanalyse (`grep`, `awk` oder spezielle Protokollanalyseprogramme)

2. Zugriffsberechtigungen:

- Zugriff auf die CDN-Managementkonsole zur Überprüfung der Konfiguration
- MediaTailor Konsolenzugriff zur Überprüfung der Konfiguration
- CloudWatch Zugriff für Metriken und Protokollanalysen
- Zugriff auf den Origin-Server für die Fehlerbehebung im Backend

3. Dokumentation:

- Diagramme zur Netzwerkarchitektur
- CDN- und MediaTailor Konfigurationsdokumentation
- Kontaktinformationen für Eskalationsverfahren
- Problembehandlung bei Runbooks für häufig auftretende Szenarien

Workflow-spezifische Anleitungen zur Fehlerbehebung

Dieser universelle Leitfaden zur Fehlerbehebung behandelt häufig auftretende Probleme bei allen MediaTailor CDN-Integrationen. Bei Problemen, die sich auf bestimmte Workflows oder Dienste beziehen, wenden Sie sich an diese speziellen Ressourcen zur Fehlerbehebung:

Fehlerbehebung bei serverseitiger Anzeigeneinfügung (SSAI)

Informationen zu SSAI-spezifischen Problemen, einschließlich Fehlern bei der Anzeigeneinfügung, Problemen beim Übergang von Anzeigen und Personalisierung, finden Sie in der Workflow-spezifischen Dokumentation zur SSAI-Fehlerbehebung.

Häufig auftretende SSAI-spezifische Probleme:

- Fehlgeschlagene Anzeigeneinfügungen und leere Werbeunterbrechungen
- Probleme beim Timing und bei der Synchronisation von Werbeanzeigen
- Probleme mit Personalisierung und Targeting
- Diskrepanzen bei der Anzeigenverfolgung und -analyse

Fehlerbehebung bei der Kanalmontage

Informationen zu Problemen, die für die Kanalbaugruppe spezifisch sind, einschließlich Problemen bei der Manifestgenerierung und der Time-Shift-Funktionalität, finden Sie in der Dokumentation zum Arbeitsablauf bei der Kanalmontage.

Häufig auftretende Probleme bei der Kanalmontage:

- Manifeste Generierungs- und Kompilierungsfehler
- Probleme mit dem Time-Shift-Fenster und der DVR-Funktionalität
- Probleme mit der Verfügbarkeit von Quellinhalten und dem Failover
- Synchronisation des Programmplans und der Metadaten

MediaPackage Fehlerbehebung bei der Integration

Informationen zu MediaPackage bestimmten Problemen wie der Manifestfilterung und Problemen mit EMP-Endpunkten finden Sie unter [Fehlerbehebung bei der CDN-Integration](#).

Häufig auftretende Probleme bei MediaPackage der Integration:

- Offensichtliche Fehler bei den Filterparametern
- MediaPackage Probleme mit der Endpunkt-Konnektivität
- EMP-spezifische Probleme mit dem Cache-Verhalten
- MediaPackage Probleme mit der Ursprungsauthentifizierung

CloudFront spezifische Problembehebung

Informationen zu CloudFront spezifischen Problemen bei der Konfiguration und Einrichtung finden Sie in der CloudFront Integrationsdokumentation.

Häufig auftretende CloudFront Probleme:

- Konfiguration der Verteilung und Einrichtung des Cache-Verhaltens
- Origin-Zugriffsidentität und Sicherheitskonfiguration
- CloudFront spezifische Fehlercodes und Antworten
- Geografische Einschränkungen und Probleme mit dem Standort am Stadtrand

Zusätzliche Ressourcen:

- Anleitungen zur Leistungsoptimierung finden Sie unter [CDN-Leistungsoptimierung](#)

- Informationen zur Einrichtung von Überwachungs- und Warnmeldungen finden Sie unter [CDN-Überwachung](#)
- Allgemeine Unterstützung und Unterstützung finden Sie unter [Holen Sie sich Unterstützung bei der CDN-Integration](#)

Analyse des CDN-Integrationsprotokolls und Fehlercode-Referenz für MediaTailor

AWS Elemental MediaTailor Die Integrationsprotokolle des Content Delivery Network (CDN) bieten wertvolle Einblicke in Leistung und Fehler. Dieses Handbuch behandelt sowohl CDN-Protokolle (aus Ihrem Content Delivery Network) als auch MediaTailor Protokolle und Fehlercodes, die für die Fehlerbehebung bei der CDN-Integration relevant sind. Verwenden Sie diese Referenz, wenn Sie wissen möchten, was Ihnen die Protokolle und Fehlercodes Ihres Content Delivery Network über Ihre MediaTailor Integration sagen. Dieses Handbuch hilft Ihnen bei der Interpretation von Protokolleinträgen und Fehlermeldungen, um die Hauptursache von Problemen zu ermitteln.

Verwandte Themen:

- Verfahren zur step-by-step Fehlerbehebung finden Sie unter [Beheben Sie Probleme mit der CDN-Integration](#)
- Informationen zu proaktiven Überwachungs- und Präventionsstrategien finden Sie unter [Überwachen Sie den Betrieb und die Leistung von MediaTailor CDN](#)
- Informationen zur Eskalation und zur Inanspruchnahme zusätzlicher Hilfe finden Sie unter [Holen Sie sich Unterstützung bei der CDN-Integration](#)
- Eine umfassende Konfiguration und Ressourcen für die MediaTailor Protokollierung finden Sie unter [MediaTailor Protokollierung von Konfigurationsressourcen](#)
- Eine Referenz zum CloudFront Protokollformat finden Sie unter [CloudFront Zugriffs-Log-Format](#)
- Eine Referenz zum HTTP-Statuscode finden Sie unter [Statuscodes für HTTP-Antworten](#)

Techniken zur Interpretation und Analyse von CDN-Protokollen für MediaTailor

AWS Elemental MediaTailor Durch die Integration von Content Delivery Network (CDN) werden detaillierte Protokolle generiert, die Ihnen helfen, Anforderungsmuster zu verstehen und Probleme zu

identifizieren. Konzentrieren Sie sich bei der Analyse von Content Delivery Network-Protokollen auf die folgenden Schlüsselindikatoren:

Hinweis: In den folgenden Beispielen wird die CloudFront Amazon-Terminologie verwendet. Andere CDN-Anbieter verwenden möglicherweise andere Begriffe für ähnliche Konzepte.

HTTP-Statuscodes

- 200- Erfolgreiche Anfrage
- 404- Die Ressource wurde nicht gefunden (überprüfen Sie die Routing-Regeln und die MediaTailor Konfiguration)
- 403- Zugriff verweigert (überprüfen Sie die CDN-Sicherheitseinstellungen und die Ursprungsberechtigungen)
- 502/503/504- Origin-Serverfehler (überprüfen Sie den MediaTailor Dienststatus und die Ursprungskonnektivität)

Indikatoren für das Cache-Verhalten

- Hit— Inhalt, der aus dem CDN-Cache bereitgestellt wird
- Miss- Vom Ursprung abgerufener Inhalt
- RefreshHit- Zwischengespeicherter Inhalt, der anhand der Herkunft validiert wurde
- Error- Die Anfrage hat zu einem Fehler geführt

Muster zur Überwachung anfordern

- Manifestanfragen sollten in der Regel zu niedrigen TTL-Einstellungen führen Miss oder RefreshHit darauf zurückzuführen sein
- Inhaltssegmente sollten Hit für beliebte Inhalte angezeigt werden
- Anzeigensegmente können Miss aufgrund der Personalisierung geschaltet werden

Überwachung des Dienstzustands

Überprüfen Sie bei der Analyse von CDN-Protokollen und bei der Behebung von Problemen den MediaTailor Dienststatus, um festzustellen, ob Probleme dienstbezogen sind:

AWS Service Health Dashboard

Überprüfen Sie den aktuellen MediaTailor Servicestatus und alle laufenden Serviceereignisse

Zugriff: [AWS Service Health Dashboard](#)

Verwenden Sie dies, wenn: Sie in CDN-Protokollen häufig auftretende 5xx-Fehler oder Service-Timeouts sehen

AWS Personal Health Dashboard

Sehen Sie sich kontospezifische Benachrichtigungen zum Servicestatus und Wartungsereignisse an

Zugang: [AWS Personal Health Dashboard](#)

Verwenden Sie dies, wenn: Sie kontospezifische Informationen zum Servicestatus oder Wartungsbenachrichtigungen benötigen

Überwachung Amazon CloudWatch Amazon-Metriken

Überwachen Sie die Indikatoren für den MediaTailor Servicestatus anhand von CloudWatch Kennzahlen:

- `GetManifest.Errors`- Verfolgen Sie Fehler bei der Manifestgenerierung
- `GetManifest.Latency`- Überwachen Sie die Leistung bei der Reaktionszeit
- `AdDecisionServer.Errors`- Überwachen Sie Verbindungsprobleme mit dem Anzeigenserver
- `Origin.Errors`- Verfolgen Sie Verbindungsprobleme mit dem Ursprungsserver

Zugriff: [CloudWatchKonsole](#)

Verwenden Sie diese Option, wenn: Sie detaillierte Leistungskennzahlen und historische Trends benötigen

Arbeitsablauf zur Behebung des Dienstzustands:

1. Den aktuellen AWS Servicestatus finden Sie im Service Health Dashboard
2. Auf dem AWS Personal Health Dashboard finden Sie kontospezifische Benachrichtigungen
3. Überwachen Sie CloudWatch Metriken für Serviceleistungsindikatoren
4. Korrelieren Sie den Dienststatus mit den CDN-Protokollmustern und Fehlerraten
5. Wenn der Dienststatus normal ist, konzentrieren Sie sich auf die CDN-Konfiguration und Probleme mit dem Ursprungsserver

Ressourcen zur CDN-Protokollanalyse

- [CloudFront Zugriffsprotokolle](#) — Vollständige Anleitung zum CDN-Protokollformat und zu den Feldern
- [CloudFront Echtzeit-Logs — Log-Streaming](#) und Analyse in Echtzeit
- [Analyse von Protokolldaten mit CloudWatch Logs Insights](#) — Erweiterte Protokollabfrage und -analyse
- [Workload-Ressourcen überwachen](#) — Well-Architected Überwachungsmuster

Themen

- [MediaTailor Fehlercodes und Fehlerbehebung bei der CDN-Integration](#)
- [Tools und Überwachungstechniken für CDN-Protokolle MediaTailor](#)
- [MediaTailor Protokollierung von Konfigurationsressourcen](#)

MediaTailor Fehlercodes und Fehlerbehebung bei der CDN-Integration

AWS Elemental MediaTailor Fehlercodes enthalten spezifische Informationen zu Integrationsproblemen, wenn sie über Ihr Content Delivery Network (CDN) weitergeleitet werden. Bei MediaTailor Rücksendefehlern über Ihr Content Delivery Network weisen diese Codes auf bestimmte Probleme hin:

Hinweis: Diese Fehlercodes werden von Ihrem CDN zurückgegeben MediaTailor und über dieses weitergeleitet. Informationen zu CDN-spezifischen Fehlern finden Sie in der Dokumentation Ihres CDN-Anbieters.

400 Bad Request (400 Ungültige Anfrage)

Häufige Ursachen: Falsch formatierte Abfrageparameter, fehlende erforderliche Parameter, ungültige Sitzung IDs

Prüfen Sie: Konfiguration der Weiterleitung von CDN-Abfragezeichenfolgen, Parameterkodierung

403 Forbidden

Häufige Ursachen: Der Client-Player fordert ein Segment an, das nicht existiert oder für das es keine Zugriffsberechtigung gibt. Für Anzeigensegmente wurde MediaTailor möglicherweise ein Segment angegeben, das nicht existiert (wenden Sie sich zur Untersuchung an den [AWS-Support](#)). Bei Inhaltssegmenten, Zugriffsbeschränkungen des ursprünglichen Anbieters oder Authentifizierungsproblemen

Prüfen Sie: Für Anzeigensegmente: Wenden Sie sich an den [AWS-Support](#), wenn MediaTailor ein ungültiges Segment generiert wird URLs. Für Inhaltssegmente: Überprüfen Sie die Berechtigungen, Authentifizierungsdaten und Zugriffsrichtlinien des Ursprungsservers. Prüfen Sie, ob auf den angeforderten Seiten Segmente vorhanden sind URLs

404 Not Found (404 Nicht gefunden)

Häufige Ursachen: MediaTailor Konfiguration existiert nicht oder ist inaktiv, falscher Wiedergabe-URL-Pfad, Manifest- oder Segmentanfragen an nicht existierende Ressourcen

Prüfung: Stellen Sie sicher, dass die MediaTailor Konfiguration existiert und aktiv ist, stellen Sie sicher, dass die Wiedergabe-URL mit der ManifestEndpointPrefix Absenderadresse übereinstimmt GetPlaybackConfiguration, überprüfen Sie, ob die CDN-Routing-Regeln Anfragen an die richtigen Endpunkte weiterleiten MediaTailor

500 Internal Server Error

Häufige Ursachen: Probleme mit dem Origin-Server, ADS-Verbindungsprobleme, Fehler bei der Verarbeitung von Manifesten

Prüfen Sie: Zustand des Origin-Servers, Gültigkeit der ADS-Antwort, MediaTailor Dienststatus

502 Schlechtes Gateway

Häufige Ursachen: Der Ursprungsserver ist nicht erreichbar, die ursprüngliche Antwort ist `UnsupportedManifestException` aufgrund von Problemen mit der Ausrichtung der HLS-Wiedergabeliste nicht erreichbar (SCTE-Markierungen sind nicht auf allen Playlisten ausgerichtet, fehlende SCTE-Markierungen auf einigen Playlisten, inkonsistentes Timing von Werbeunterbrechungen bei allen Bitratenvarianten)

Überprüfe: Konnektivitäts- und Firewallregeln des Origin-Servers, DNS-Auflösung, Konsistenz der HLS-Wiedergabeliste über alle Bitratenvarianten hinweg, Ausrichtung der SCTE-35-Markierungen auf denselben Segmenten in allen Playlisten, vergewissere dich, dass alle Playlisten die erforderlichen Werbeunterbrechungsmarkierungen enthalten

Ressourcen zur Fehlercode-Analyse:

- [Problembehandlung MediaTailor](#) — Vollständige MediaTailor Fehlercode-Referenz
- [Behebung von CloudFront Fehlerantworten](#) — CDN-spezifische Fehleranalyse
- [Statuscodes für HTTP-Antworten](#) — Umfassende Referenz zum HTTP-Statuscode
- [Fehlermanagement](#) — Well-Architected Fehlerbehandlungsmuster

Tools und Überwachungstechniken für CDN-Protokolle MediaTailor

AWS Elemental MediaTailor Durch die Integration von Content Delivery Network (CDN) werden große Mengen an Protokolldaten generiert, für die effiziente Analysetools und -techniken erforderlich sind. Verwenden Sie diese Ansätze, um Content Delivery Network und MediaTailor Logs effizient zu analysieren:

- Befehlszeilenanalyse: Verwenden Sie Tools wie `grep`, `undawk`, um Protokollmuster `sort` zu filtern und zu analysieren
- Amazon CloudWatch Logs Insights: CDN und MediaTailor Logs mit SQL-ähnlicher Syntax für erweiterte Analysen abfragen
- Tools von Drittanbietern: Ziehen Sie Plattformen zur Protokollanalyse für umfassende Überwachung und Warnmeldungen in Betracht
- Benutzerdefinierte Dashboards: Erstellen Sie Visualisierungen, die CDN-Metriken mit Leistungsdaten kombinieren MediaTailor

Wenn Sie zusätzliche Unterstützung bei der Protokollanalyse oder der Interpretation komplexer Fehlermuster benötigen, finden Sie weitere Informationen unter. [Holen Sie sich Unterstützung bei der CDN-Integration](#)

Tools und Ressourcen zur Protokollanalyse:

- [CloudWatch Logs Insights](#) — SQL-ähnliche Abfragen für die Protokollanalyse
- [Amazon OpenSearch Service](#) — Erweiterte Protokollsuche und Analyse
- [CloudWatch Dashboards](#) — Benutzerdefinierte Visualisierung und Überwachung
- [Instrumentierung verteilter Systeme für betriebliche Transparenz](#) — Fortschrittliche Beobachtbarkeitsmuster
- [Workload-Beobachtbarkeit entwerfen](#) — Well-Architected zur Beobachtbarkeit

MediaTailor Protokollierung von Konfigurationsressourcen

Neben CDN-Protokollen MediaTailor bietet es umfassende Protokollierungsfunktionen für die Überwachung der Anzeigeneinfügung, der Manifestgenerierung und der Serviceinteraktionen. Verwenden Sie diese Ressourcen, um MediaTailor Protokolle zu konfigurieren und zu analysieren:

MediaTailor Manifest- und Dienstprotokolle

Konfigurieren und analysieren Sie Protokolle für die Manifestgenerierung, Ursprungsinteraktionen und Serviceereignisse:

- [AWS Elemental MediaTailor Beschreibung und Ereignistypen der Manifestprotokolle](#)- Vollständige Referenz für MediaTailor Manifestprotokolle und Ereignistypen
- [AWS Elemental MediaTailor Protokolle anzeigen](#)- Leitfaden zum Anzeigen und Interpretieren von MediaTailor Protokollen

Konfiguration von Vended-Protokollen

Konfigurieren Sie die flexible Protokollzustellung an mehrere Ziele mit Kostenoptimierung:

- [Versendete Protokolle werden zum Senden von Protokollen verwendet AWS Elemental MediaTailor](#)- Konfigurieren Sie die Protokollzustellung an Amazon S3, Firehose oder CloudWatch Logs
- [Migrieren Sie Ihre AWS Elemental MediaTailor Protokollierungsstrategie](#)- Migrationsleitfaden für bestehende Protokollierungskonfigurationen

CloudWatch Integration von Protokollen

Integrieren Sie MediaTailor Protokolle CloudWatch zur Überwachung und Analyse in:

- [AWS Elemental MediaTailor Protokolle anzeigen](#)- Konfiguration und Analyse von CloudWatch Protokollen
- [CloudWatchInsights](#)- Erweiterte Protokollabfrage und -analyse

Anzeigenspezifische Protokollierung

Überwachen Sie die Leistung der Anzeigeneinblendung und beheben Sie Probleme im Zusammenhang mit Anzeigen:

- ADS-Interaktionsprotokolle — Überwachen Sie die Kommunikation und Fehler des Anzeigenentscheidungsservers

Hauptunterschiede zwischen CDN und MediaTailor Protokollen:

- CDN-Protokolle: Zeigt request/response Muster, Cache-Verhalten und Fehler auf Netzwerkebene in Ihrem Content Delivery Network
- MediaTailor Protokolle: Zeigt Details zur Anzeigeneinfügung, Ereignisse bei der Manifestgenerierung, Interaktionen mit dem Ursprung und dienstspezifische Fehler

- Kombinierte Analyse: Verwenden Sie beide Protokolltypen zusammen, um einen vollständigen Überblick über Ihren Streaming-Workflow zu erhalten

Eine umfassende Überwachung, die sowohl CDN als auch MediaTailor Protokollierung kombiniert, finden Sie unter [Überwachen Sie den Betrieb und die Leistung von MediaTailor CDN](#).

Automatisieren MediaTailor und CDN mit CloudFormation

Durch die Automatisierung AWS Elemental MediaTailor mithilfe eines Content Delivery Network (CDN) wird Ihr Arbeitsablauf bei der Anzeigenschaltung AWS CloudFormation optimiert und die Zuverlässigkeit verbessert.

In diesem Abschnitt erfahren Sie, wie Sie mithilfe von AWS CloudFormation (AWS Infrastructure as Code Service) die automatische Einrichtung AWS Elemental MediaTailor mit einem Content Delivery Network (CDN) durchführen können. Sie können diese Integration zwar manuell konfigurieren, wie in den vorherigen Abschnitten beschrieben, aber die Verwendung CloudFormation spart Zeit und reduziert Fehler, da der gesamte Prozess mit einer einzigen Vorlage automatisiert wird.

Wenn Sie mit diesem Service noch nicht vertraut sind CloudFormation, können Sie eine Vorlagendatei erstellen, die alle AWS-Ressourcen definiert, die Sie benötigen. Wenn Sie diese Vorlage bereitstellen, CloudFormation werden diese Ressourcen automatisch für Sie erstellt und konfiguriert, um sicherzustellen, dass sie korrekt zusammenarbeiten.

Weitere Informationen zu CloudFormation finden Sie im [CloudFormation Benutzerhandbuch](#). Informationen zu MediaTailor Ressourcentypen finden Sie unter [AWS: MediaTailor Ressourcentyp-Referenz](#). CloudFormation

Themen

- [Warum CloudFormation für MediaTailor und CDN-Integration verwenden](#)
- [Bereiten Sie sich auf die CloudFormation Bereitstellung von CDN und MediaTailor Integrationen vor](#)
- [Stellen Sie die CloudFormation Vorlage für CDN und Integrationen bereit MediaTailor](#)
- [Verwenden Sie die CloudFormation bereitgestellten Ressourcen für CDN und Integration MediaTailor](#)
- [Testen und validieren Sie Ihre CloudFormation Bereitstellung für CDN und Integration MediaTailor](#)
- [Beheben Sie häufig auftretende Probleme bei der CloudFormation Bereitstellung von CDN und Integrationen MediaTailor](#)

- [Passen Sie die CloudFormation Vorlage für CDN und MediaTailor Integrationen an](#)
- [AWS CloudFormation Vorlagenreferenz für AWS Elemental MediaTailor und CloudFront Amazon-Integration](#)

Warum CloudFormation für MediaTailor und CDN-Integration verwenden

AWS Elemental MediaTailor Automatisierung mit AWS CloudFormation erheblichen Vorteilen für Rundfunkprofis bei der Verwaltung von Streaming-Workflows. Die manuelle Konfiguration MediaTailor mit einem Content Delivery Network (CDN) kann zeitaufwändig und fehleranfällig sein. Der Einsatz von CloudFormation Automatisierung bietet die folgenden Vorteile.

- **Konsistenz:** Stellt sicher, dass jedes Mal dieselbe Konfiguration bereitgestellt wird, wodurch menschliche Fehler reduziert werden.
- **Versionskontrolle:** Speichern Sie Ihre Infrastruktur als Code in einem Versionskontrollsystem, um Änderungen nachzuverfolgen.
- **Schnelle Bereitstellung:** Stellen Sie komplexe Konfigurationen innerhalb von Minuten statt stundenlanger manueller Konfiguration bereit.
- **Umgebungsreplikation:** Einfache Replikation von Konfigurationen in Entwicklungs-, Test- und Produktionsumgebungen.
- **Dokumentation:** Die Vorlage selbst dient als Dokumentation Ihrer Infrastruktur.

So sieht der automatisierte Workflow im Vergleich zur manuellen Konfiguration aus:

Manuelle Einrichtung (mehrere Schritte)	Automatisierte Einrichtung (einzelne Vorlage)
MediaTailor Wiedergabekonfiguration erstellen	Stellen Sie eine CloudFormation Vorlage mit Parametern bereit
CloudFront Distribution erstellen	
Konfigurieren Sie das Verhalten des Caches	
Richten Sie Sicherheitskonfigurationen ein	

Der automatisierte Workflow für die Einrichtung MediaTailor mit CloudFront folgt diesen Schritten:

1. Stellen Sie die CloudFormation Vorlage mit Ihren Inhaltsursprungs- und Anzeigenserver-Parametern bereit
2. CloudFormation erstellt und konfiguriert alle erforderlichen Ressourcen:
 - MediaTailor Wiedergabekonfiguration für das Einfügen von Anzeigen
 - CloudFront Verteilung mit geeignetem Cache-Verhalten
 - Sicherheitskonfigurationen für den Schutz von Inhalten
3. Verwenden Sie die CloudFormation Ausgaben, um auf Ihren Stream mit aktivierter Werbung zuzugreifen URLs
4. Streamen Sie Ihre Inhalte mit dynamisch eingefügten Anzeigen

Bereiten Sie sich auf die CloudFormation Bereitstellung von CDN und MediaTailor Integrationen vor

AWS Elemental MediaTailor Für die Bereitstellung AWS CloudFormation sind bestimmte Voraussetzungen und Vorbereitungsschritte erforderlich. Bevor Sie beginnen, mit CloudFormation to integration MediaTailor und Amazon zu arbeiten CloudFront, stellen Sie sicher, dass Sie über Folgendes verfügen.

- Ein AWS Konto mit Berechtigungen zum Erstellen von MediaTailor CloudFront, und CloudFormation Ressourcen
- Ein Inhaltsursprung, an dem Ihre Videoinhalte gehostet werden (AWS Elemental MediaPackage z. B. Amazon S3 oder ein anderer Ursprungsserver)
- Ein Ad Decision Server (ADS), der auf VAST-Anfragen antworten kann

Stellen Sie vor dem Bereitstellen der CloudFormation Vorlage die folgenden erforderlichen Parameter zusammen:

`AdServerUrl`

URL des VAST-Anzeigenservers für die dynamische Anzeigeneinfügung. Zum Testen wird ein statischer VAST-Endpunkt bereitgestellt.

`ContentOriginDomainName`

Domainname Ihrer Inhaltsquelle ohne Protokoll (z. B. *mediapackage-domain.mediapackagev2.us-west-2.amazonaws.com*, *mybucket.s3.amazonaws.com*,

oder *custom-origin.example.com*). Verwenden Sie weder die Präfixe `http://` oder `https://` noch irgendwelche Pfade.

ContentOriginType

Die Art der Herkunft des Inhalts:

- `mediapackagev2`: Für Ursprünge AWS Elemental MediaPackage
- `s3`: Für Amazon S3 S3-Bucket-Ursprünge
- `benutzerdefiniert`: Für jeden anderen Origin-Typ

Mit der Vorlage werden mehrere AWS Ressourcen erstellt, die zusammenarbeiten, um Ihre Inhalte mit personalisierten Anzeigen bereitzustellen. Im Folgenden wird beschrieben, was die einzelnen Komponenten bewirken:

Origin-Zugriffskontrolle

Origin Access Control (OAC) ist eine Sicherheitsfunktion, die sicherstellt, dass auf deine Inhalte nur über CloudFront, nicht direkt von deinem Ursprungsserver aus zugegriffen werden kann. Dies trägt dazu bei, deine Inhalte vor unbefugtem Zugriff zu schützen.

Für MediaPackage Amazon S3 S3-Ursprünge erstellt die Vorlage eine Origin Access Control (OAC) - Ressource, um den Zugriff auf Ihre Inhalte zu sichern.

MediaTailor Konfiguration der Wiedergabe

Die MediaTailor Wiedergabekonfiguration ist die Kernkomponente, die das Einfügen von Anzeigen übernimmt. Es empfängt Inhalte von Ihrer Herkunft, fordert Anzeigen von Ihrem Werbeserver an und kombiniert sie zu einem personalisierten Stream für jeden Zuschauer.

Die Vorlage erstellt eine MediaTailor Wiedergabekonfiguration mit den folgenden Einstellungen:

- Quelle des Videoinhalts, die auf Ihre CloudFront Distribution verweist
- Die URL des Anzeigenentscheidungsservers ist für Ihren angegebenen VAST-Endpunkt konfiguriert
- Live-Pre-Roll-Konfiguration für die Anzeigeneinblendung während Livestreams
- CDN-Konfiguration mit entsprechenden Segment-URL-Präfixen

CloudFront Verteilung

Der CloudFront Vertrieb liefert Ihre Inhalte mit geringer Latenz an Zuschauer weltweit. Es verarbeitet verschiedene Arten von Anfragen (Manifeste, Inhaltssegmente, Anzeigensegmente) und leitet sie an die entsprechenden Quellen weiter.

Für Broadcast-Profis CDNs, die noch keine Erfahrung damit haben, sind hier einige wichtige Begriffe aufgeführt:

Ursprung

Ein Server, auf dem Ihre Originalinhalte gespeichert sind (wie MediaPackage Amazon S3)

Cache-Verhalten

Regeln, die bestimmen, wie verschiedene Arten von Inhalten zwischengespeichert und bereitgestellt werden

Cache-Richtlinie

Einstellungen, die steuern, wie lange Inhalte zwischengespeichert werden und welche Anforderungskomponenten das Caching beeinflussen

Die Vorlage erstellt eine CloudFront Distribution mit den folgenden Komponenten:

- Drei Ursprünge:
 - Herkunft des Inhalts (MediaPackageAmazon S3 oder benutzerdefiniert)
 - MediaTailor manifestiert den Ursprung
 - MediaTailor Herkunft der Segmente
- Verhalten mit entsprechenden Mustern zwischenspeichern:
 - Standardverhalten für Inhaltssegmente
 - Verhalten für MediaTailor Anzeigensegmente (/tm/*)
 - Verhalten bei MediaTailor interstitiellen Medien (/v1/i-media/*)
 - Verhalten bei personalisierten Manifesten (/v1/*)
 - Verhalten bei Segment-Umleitungsanfragen (/segment/*)
- Optimierte Cache-Richtlinien für jedes Verhalten:
 - CachingOptimizedfür zwischenspeicherbare Inhalte
 - CachingDisabledfür personalisierte Manifeste

- Richtlinien für Origin-Anfragen, um eine korrekte Header-Weiterleitung sicherzustellen
- Response-Header-Richtlinien für die CORS-Unterstützung

Stellen Sie die CloudFormation Vorlage für CDN und Integrationen bereit

MediaTailor

AWS Elemental MediaTailor Die Bereitstellung mithilfe der AWS CloudFormation Vorlage ist unkompliziert, sobald Sie wissen, was mit der Vorlage erstellt wird. Dieser Vorgang dauert etwa 15 bis 30 Minuten, wobei die meiste Zeit damit verbracht wird, auf die Bereitstellung der CloudFront Distribution zu warten.

Gehen Sie wie folgt vor, um die CloudFormation Vorlage bereitzustellen und Ihren automatisierten Arbeitsablauf für die Anzeigenschaltung einzurichten:

Um die MediaTailor CloudFormation Vorlage bereitzustellen

1. Laden Sie die CloudFormation Vorlage aus dem AWS Elemental MediaTailor GitHub Repository herunter oder kopieren Sie sie aus dem [AWS CloudFormation Vorlagenreferenz für AWS Elemental MediaTailor und CloudFront Amazon-Integration](#).
2. Öffnen Sie die [CloudFormation -Konsole](#).
3. Wählen Sie Stack erstellen > Mit neuen Ressourcen (Standard).
4. Wählen Sie unter Vorlage angeben die Option Vorlagendatei hochladen aus und laden Sie die Vorlage hoch.
5. Geben Sie einen Stacknamen ein und geben Sie Werte für die erforderlichen Parameter an:
 - AdServerUrl: URL Ihres VAST-Werbeservers (z. B. <https://your-ad-server.com/vast>)
 - ContentOriginDomainName: Domainname Ihrer Inhaltsquelle ohne Protokoll (z. B. mediapackage-domain.mediapackagev2.us-west-2.amazonaws.com)
 - ContentOriginType: Wählen Sie die Art der Herkunft der Inhalte aus:
 - mediapackagev2: Für Ursprünge AWS Elemental MediaPackage
 - s3: Für Amazon S3 S3-Bucket-Ursprünge
 - benutzerdefiniert: Für jeden anderen Origin-Typ
6. Überprüfen Sie die Konfiguration und wählen Sie Stack erstellen.
7. Warten Sie, bis die Erstellung des Stacks abgeschlossen ist, was normalerweise 5-10 Minuten dauert. Sie können den Fortschritt auf der Registerkarte Ereignisse überwachen.

8. Sobald Sie fertig sind, navigieren Sie zur Registerkarte Ausgaben, um die URLs für Ihre HLS- und DASH-Manifeste zu finden.

 Note

Wenn Sie den Inhalt AWS Elemental MediaPackage als Quelle verwenden, stellen Sie sicher, dass Ihre MediaPackage Endgeräte ordnungsgemäß konfiguriert und zugänglich sind. Weitere Informationen finden Sie unter [MediaPackage CDN-Integration](#).

Verwenden Sie die CloudFormation bereitgestellten Ressourcen für CDN und Integration MediaTailor

AWS Elemental MediaTailor Die vom AWS CloudFormation Stapel bereitgestellten Ressourcen bieten mehrere wichtige Ergebnisse, anhand derer Sie mithilfe der Anzeigeneinfügung auf Ihre Inhalte zugreifen können. Nachdem der CloudFormation Stapel erfolgreich erstellt wurde, müssen Sie wissen, wie Sie die Ausgaben verwenden können, um auf Ihre Inhalte zuzugreifen, in die Anzeigen eingefügt wurden. Dies ist vergleichbar mit der Verwendung MediaTailor URLs bei einer manuellen Einrichtung, die CloudFormation Bereitstellung stellt diese jedoch URLs automatisch bereit.

Nach erfolgreicher Bereitstellung bietet der CloudFormation Stack mehrere wichtige Ergebnisse, die Sie für den Zugriff auf Ihre Inhalte mit Anzeigeneinblendung verwenden werden:

CloudFrontDomainName

Der Domainname Ihrer CloudFront Distribution *d1234abcdef.CloudFront.net* (z. B.

HlsManifestUrl

Basis-URL für HLS-Manifeste mit Anzeigeneinfügung (z. B.

https://d1234abcdef.CloudFront.net12345/v1/master///) my-playback-config

DashManifestUrl

Basis-URL für DASH-Manifeste mit Anzeigeneinfügung (z. B. *https:///v1/dash///)*

d1234abcdef.CloudFront.net 12345 my-playback-config

MediaTailorPlaybackConfigName

Name der erstellten MediaTailor Wiedergabekonfiguration (z. B. *my-stack-PlaybackConfig*)

Konstruiert die Wiedergabe URLs

Um die vollständige Wiedergabe-URL für Ihre Inhalte mit Werbung zu erstellen, müssen Sie die Basis-URL aus den CloudFormation Ausgaben mit Ihrem spezifischen Manifestpfad kombinieren. Für Rundfunkfachleute ist es wichtig, diesen Schritt zu verstehen, da er Ihre vorhandenen Inhalte mit dem System zur Anzeigeneinblendung verbindet.

1. Beginnen Sie mit der entsprechenden Manifest-URL aus den Ausgaben:

```
HlsManifestUrl: https://d1234abcdef.CloudFront.net/v1/master/12345/my-playback-  
config/
```

2. Hängen Sie Ihren spezifischen Manifestpfad an:

```
Your manifest path: channel/index.m3u8
```

3. Die vollständige Wiedergabe-URL lautet:

```
https://d1234abcdef.CloudFront.net/v1/master/12345/my-playback-config/channel/  
index.m3u8
```

Verwenden Sie diese URL in Ihrem Videoplayer, um Inhalte mit dynamisch eingefügten Anzeigen abzuspielen.

Tip

Wenn Sie sich nicht sicher sind, wie Ihr Manifestpfad lauten soll, überprüfen Sie Ihren Ursprungsserver. Bei MediaPackage Origins ist dies der Pfad zum HLS- oder DASH-Manifest Ihres Endpunkts. Für Amazon S3 S3-Ursprünge ist dies der Pfad zu Ihrer Manifestdatei innerhalb des Buckets.

Weitere Informationen zur MediaTailor URL-Struktur finden Sie unter [Richten Sie die CDN-Integration ein mit MediaTailor](#).

Konfigurieren Sie einen Videoplayer

Nachdem Sie Ihre Wiedergabe-URL erhalten haben, müssen Sie einen Videoplayer für die Verwendung konfigurieren. Für Rundfunkprofis ähnelt dies der Konfiguration eines Players für einen

beliebigen HLS- oder DASH-Stream, aber jetzt wird der Stream personalisierte Werbung enthalten. Hier ist ein einfaches Beispiel mit dem beliebten HLS.js Player:

```
<!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>
```

Sie können auch professionelle Broadcast-Player verwenden wie:

- JW-Spieler
- Bitmovin-Spieler
- THEOplayer
- Video.js

Weitere Informationen zur Player-Integration mit finden Sie MediaTailor unter. [MediaTailor Anforderungen an die Ad-Server-Integration](#)

Testen und validieren Sie Ihre CloudFormation Bereitstellung für CDN und Integration MediaTailor

AWS Elemental MediaTailor Die Validierung der Bereitstellung ist für Rundfunkfachleute ein wichtiger Schritt, bevor sie live gehen. In diesem Abschnitt erfahren Sie, wie Sie Ihre Implementierung testen, um sicherzustellen, dass Anzeigen ordnungsgemäß eingefügt und Inhalte reibungslos bereitgestellt werden.

Gehen Sie nach der Bereitstellung der CloudFormation Vorlage wie folgt vor, um sicherzustellen, dass Ihre Einrichtung ordnungsgemäß funktioniert:

Um Ihre CloudFront Landintegration MediaTailor zu testen

1. Stellen Sie sicher, dass alle Ressourcen erfolgreich in der CloudFormation Konsole erstellt wurden.
2. Vergewissern Sie sich, dass die MediaTailor Wiedergabekonfiguration in der [MediaTailor Konsole](#) aktiv ist.
3. Stellen Sie sicher, dass die CloudFront Distribution in der [CloudFront Konsole](#) bereitgestellt und aktiviert ist.
4. Testen Sie die Wiedergabe anhand eines Beispielmanifests:
 1. Konstruieren Sie die vollständige Wiedergabe-URL wie unter beschrieben [Konstruiert die Wiedergabe URLs](#).
 2. Verwenden Sie einen Videoplayer, der HLS oder DASH unterstützt (wie VLC, JW Player oder den AWS Konsolenplayer).
 3. Vergewissern Sie sich, dass Inhalte wiedergegeben und Werbeanzeigen an den erwarteten Unterbrechungspunkten eingefügt werden.
5. Überprüfen Sie die MediaTailor Anmeldedaten CloudWatch auf Fehler beim Einfügen von Anzeigen.

Achten Sie beim Testen der Anzeigeneinblendung auf die folgenden Erfolgsindikatoren:

- Reibungslose Übergänge zwischen Inhalten und Anzeigen
- Anzeigen erscheinen an den erwarteten Breakpoints (Pre-Roll, Mid-Roll, Post-Roll)
- Die Qualität der Anzeige entspricht der Qualität der Inhalte
- Keine Puffer- oder Wiedergabefehler bei Anzeigenübergängen

Ausführlichere Testverfahren finden Sie unter [Das Verhalten bei der AWS Elemental MediaTailor Anzeigeneinblendung verstehen](#). Umfassende CDN-Integrationstests und -validierungen finden Sie unter [Testen und Validieren für CDN und Integrationen MediaTailor](#).

Beheben Sie häufig auftretende Probleme bei der CloudFormation Bereitstellung von CDN und Integrationen MediaTailor

AWS Elemental MediaTailor Probleme bei der Bereitstellung können selbst bei der Automatisierung während der Bereitstellung oder Wiedergabe auftreten. Wenn Sie als Broadcast-Profi wissen, wie Sie diese Probleme beheben können, können Sie einen zuverlässigen Streaming-Dienst mit Werbeeinblendung aufrechterhalten.

Wenn Sie Probleme mit Ihrer CloudFormation Bereitstellung oder der daraus resultierenden MediaTailor CloudFront Integration haben, finden Sie Informationen zu diesen häufig auftretenden Problemen und Lösungen:

CloudFormation Probleme bei der Bereitstellung

Die Stack-Erstellung schlägt mit dem Fehler „Ressourcenerstellung fehlgeschlagen“ fehl

Mögliche Ursachen:

- Ungültiges Format des Domainnamens für den Ursprung des Inhalts
- Unzureichende Berechtigungen zum Erstellen von Ressourcen

Lösung: Überprüfen Sie den spezifischen Ressourcenfehler auf der Registerkarte CloudFormation Ereignisse. Stellen Sie sicher, dass der Domainname für den Ursprung des Inhalts korrekt formatiert ist und keine Protokollpräfixe oder Pfade enthält. Stellen Sie sicher, dass Ihre IAM-Rolle über ausreichende Berechtigungen verfügt, um alle erforderlichen Ressourcen zu erstellen.

CloudFront Die Bereitstellung nimmt viel Zeit in Anspruch

Ursache: Die vollständige Bereitstellung von CloudFront Distributionen dauert in der Regel 15 bis 30 Minuten.

Lösung: Das ist ein normales Verhalten. Warten Sie, bis die Distribution den Status „Bereitgestellt“ erreicht hat, bevor Sie den Test durchführen.

Probleme bei der Wiedergabe und beim Einfügen von Anzeigen

Inhalte werden abgespielt, aber es werden keine Anzeigen eingefügt

Mögliche Ursachen:

- Der Anzeigenentscheidungsserver reagiert nicht oder gibt leeres VAST zurück
- Der Inhalt enthält keine Werbemarkierungen

Lösung: Stellen Sie sicher, dass Ihr Anzeigenserver zugänglich ist und gültige VAST-Antworten zurückgibt. Vergewissern Sie sich, dass Ihre Inhalte über die richtigen Werbemarkierungen verfügen (SCTE-35-Markierungen für Live-Inhalte oder Ad-Break-Tags für VOD).

403 Unzulässige Fehler beim Zugriff auf Inhalte

Mögliche Ursachen:

- Die Origin-Zugriffskontrolle ist nicht richtig konfiguriert
- Probleme mit Origin-Bucket- oder Endpunktberechtigungen

Lösung: Stellen Sie für Amazon S3 S3-Ursprünge sicher, dass die Bucket-Richtlinie den Zugriff über die CloudFront Distribution zulässt. Überprüfen Sie bei MediaPackage Origins, ob die Origin-Zugriffskontrolle richtig konfiguriert ist und ob auf den Endpunkt zugegriffen werden kann.

Wiedergabefehler oder Pufferung

Mögliche Ursachen:

- Pfadmuster im Cache stimmen nicht mit Inhaltspfaden überein
- Falsche Konfiguration der Ursprungsdomäne

Lösung: Stellen Sie sicher, dass Ihr Cache-Verhalten die richtigen Pfadmuster aufweist, um Anfragen an die entsprechenden Ursprünge weiterzuleiten. Überprüfen Sie die CloudFront Protokolle, um zu sehen, welcher Ursprung die Anfragen bearbeitet, und stellen Sie sicher, dass es sich um den erwarteten Ursprung handelt.

Für Rundfunkfachleute können diese zusätzlichen Tipps zur Fehlerbehebung hilfreich sein:

- Verwenden Sie Amazon CloudWatch Logs Insights, um nach bestimmten Fehlermustern in MediaTailor Protokollen zu suchen
- Testen Sie zuerst mit einem einfachen VAST-Anzeigenserver (wie dem in der Vorlage bereitgestellten Standardserver), bevor Sie Ihren Produktionsanzeigenserver verwenden

- Überprüfen Sie die Anzeigenmarkierungen Ihrer Inhalte mithilfe des MediaTailor Manifest-Inspector-Tool in der Konsole
- Überprüfen Sie den Netzwerkverkehr in den Entwicklertools Ihres Browsers, um festzustellen, ob Anfragen an den Anzeigenserver korrekt gestellt werden

Weitere Informationen zur Fehlerbehebung finden Sie in den CloudWatch Protokollen für beide MediaTailor und CloudFront um spezifische Fehler zu identifizieren.

Passen Sie die CloudFormation Vorlage für CDN und MediaTailor Integrationen an

AWS Elemental MediaTailor Die Anpassung der Vorlage ermöglicht es Rundfunkfachleuten, die AWS CloudFormation Vorlage an spezifische Workflow-Anforderungen anzupassen. Obwohl die Basisvorlage für viele Szenarien geeignet ist, können Ihnen diese Anpassungen dabei helfen, komplexere Anforderungen zu erfüllen.

Die folgenden Beispiele zeigen YAML-Codefragmente, die Sie der Vorlage hinzufügen können. Wenn Sie mit YAML oder der CloudFormation Syntax nicht vertraut sind, sollten Sie erwägen, mit einem Entwickler oder AWS Lösungsarchitekten zusammenzuarbeiten, um diese Änderungen vorzunehmen.

Sie können die CloudFormation Vorlage an Ihre spezifischen Workflow-Anforderungen anpassen.

Ursprünge hinzufügen oder ändern

Für Broadcast-Workflows, die mehrere Inhaltsquellen verwenden (wie Primär- und Backup-Quellen oder verschiedene Inhaltsbibliotheken), können Sie Ihrer CloudFront Distribution zusätzliche Ursprünge hinzufügen:

```
Origins:
  # Add a new origin for additional content
  - Id: SecondaryContentOrigin
    DomainName: secondary-content.example.com
    CustomOriginConfig:
      OriginProtocolPolicy: 'https-only'
      OriginSSLProtocols:
        - TLSv1.2
```

Fügen Sie dann ein entsprechendes Cache-Verhalten hinzu, um bestimmte Muster an diesen Ursprung weiterzuleiten:

```
CacheBehaviors:
  - PathPattern: '/secondary-content/*'
    TargetOriginId: SecondaryContentOrigin
    ViewerProtocolPolicy: 'https-only'
    CachePolicyId: 658327ea-f89d-4fab-a63d-7e88639e58f6 # Managed-CachingOptimized
```

Erstellen Sie benutzerdefinierte Cache-Richtlinien

Für Broadcast-Workflows mit bestimmten Caching-Anforderungen (wie Qualitätsauswahlparameter oder Zuschauerauthentifizierung) können Sie benutzerdefinierte Cache-Richtlinien erstellen, anstatt die verwalteten zu verwenden. Eine ausführliche Anleitung zu TTL-Werten und Caching-Strategien finden Sie unter. [Caching-Optimierung für CDN und Integrationen MediaTailor](#)

```
# 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
```

Konfiguration verbessern MediaTailor

Für Broadcast-Workflows, für die erweiterte Funktionen zum Einfügen von Anzeigen erforderlich sind, können Sie die MediaTailor Konfiguration um Optionen wie Anzeigenvorabruf (zur Verringerung der Latenz), Schwellenwerte für Personalisierung und Bumper-Anzeigen erweitern.

```
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...
```

Weitere Informationen MediaTailor zu Konfigurationsoptionen finden Sie unter. [AWS Elemental MediaTailor Zum Einfügen von Anzeigen verwenden](#)

Fügen Sie Sicherheitsfunktionen hinzu

Für Broadcast-Workflows mit bestimmten Sicherheitsanforderungen (wie geografische Einschränkungen oder Schutz vor DDoS-Angriffen) können Sie AWS WAF Integrations- und geografische Einschränkungen hinzufügen:

```
# 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...

```

Weitere Informationen zu CloudFormation Vorlagen finden Sie im [Benutzerhandbuch.AWS CloudFormation](#)

Broadcast-spezifische CloudFormation Vorlagen und Beispiele finden Sie im [AWS Media Services Tools-Repository](#). GitHub

AWS CloudFormation Vorlagenreferenz für AWS Elemental MediaTailor und CloudFront Amazon-Integration

AWS Elemental MediaTailor Die Integration mit Amazon CloudFront kann mithilfe der folgenden vollständigen AWS CloudFormation Vorlage automatisiert werden:

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://d1kbnmkziz9rksx.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.mediataylor.${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.ManifestEndpoint

  MediaTailorPlaybackConfigName:
    Description: Name of the MediaTailor playback configuration

```

```
Value: !Ref MediaTailorPlaybackConfig
```

Serienreife Konfiguration für CloudFront MediaTailor

Diese CloudFront Verteilungskonfiguration bietet alles, was Sie für die Bereitstellung von MediaTailor Inhalten mit serverseitiger Anzeigeneinblendung in großem Umfang benötigen. Kopieren Sie diese Konfiguration und passen Sie sie an Ihre spezifischen Ursprünge und Anforderungen an.

Was erreicht diese Konfiguration

Diese Konfiguration sorgt für eine produktionsbereite CloudFront Distribution, die alle MediaTailor Anforderungstypen mit optimaler Zwischenspeicherung und Leistung verarbeitet. Sie umfasst drei Ursprünge (Ihre Inhalte, MediaTailor Segmente und MediaTailor Manifeste) mit vier Cache-Verhaltensweisen, die Anfragen korrekt weiterleiten und Inhalte entsprechend zwischenspeichern.

Wann sollte diese Konfiguration verwendet werden

Dieses Setup ist ideal für Live-Streaming und Hybrid-Workflows video-on-demand, die serverseitige Anzeigeneinfügung erfordern.

Architektur mit drei Ursprüngen

MediaTailor verwendet ein Architekturmuster mit drei Ursprüngen, um die Bereitstellung von Inhalten und die Leistung der Anzeigeneinblendung zu optimieren. Jeder Ursprung dient einem bestimmten Zweck im Workflow zur Anzeigeneinfügung:

Herkunft Ihrer Inhalte

Dies ist die wahre Herkunft Ihrer Inhalte, aus denen Feeds stammen MediaTailor. Dies könnte beispielsweise AWS Elemental MediaPackage V2 oder ein anderer Service zur Bereitstellung von Inhalten sein. Dieser Origin stellt Ihren ursprünglichen Inhalt vor der Anzeigenschaltung bereit. Allgemeine Beispiele sind:

- MediaPackage V2-Verpackungskonfigurationen
- Netzwerke zur Bereitstellung von Inhalten von Drittanbietern
- Streaming-Server vor Ort
- Amazon S3 S3-Buckets mit statischem Inhalt

MediaTailor Herkunft der Segmente

Dieser Ursprung verwendet den Hostnamen

`segments.mediatailor.region.amazonaws.com` und stellt die eigentlichen Anzeigensegmente bereit, nachdem sie MediaTailor codiert wurden. Dies sind die Videosegmente, die die Werbung enthalten. Dieser Ursprung behandelt:

- Transkodierte Anzeigensegmente im gleichen Format wie Ihre Inhalte
- Anfragen wurden anhand des `/segment/*` Pfadmusters umgeleitet
- Anzeigensegmente, die für die serverseitige Anzeigeneinfügung verarbeitet wurden

MediaTailor manifestiert Herkunft

Dieser Ursprung verwendet den Hostnamen

`manifests.mediatailor.region.amazonaws.com` und kann als regionaler Hostname für Wiedergabekonfigurationen in der angegebenen AWS Region verwendet werden. MediaTailor wählt basierend auf dem Pfad in der Anfrage die richtige Wiedergabekonfiguration aus. Dieser Ursprung bietet:

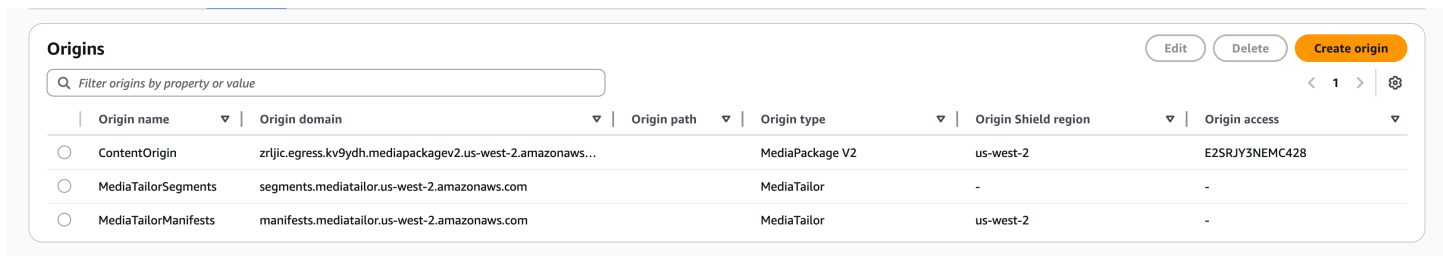
- Personalisierte HLS- und DASH-Manifeste mit zuschauerspezifischer Anzeigeneinblendung
- Servergesteuerte Anzeigeneinfügungen (SGAI) für zwischenspeicherbare Inhalte
- Ad-Tracking und Beacon-Verarbeitung für serverseitige Berichterstattung

Mit dem ursprünglichen Hostnamen können Sie mehrere Wiedergabekonfigurationen verwenden `manifests.mediatailor.region.amazonaws.com`, die mit derselben CloudFront Distribution funktionieren, sofern sie sich in der angegebenen Region befinden und Sie den Namen der Wiedergabekonfiguration in den Anforderungspfad aufnehmen. Beispiel:

- `https://your-distribution.cloudfront.net/v1/master/playback-config-1/manifest.m3u8`
- `https://your-distribution.cloudfront.net/v1/master/playback-config-2/manifest.m3u8`

Wählen Sie die Richtlinien für ursprüngliche Anfragen je nach Inhaltstyp aus, um Cache-Poisoning zu verhindern und gleichzeitig die einwandfreie Funktionalität sicherzustellen. Der Hauptunterschied besteht zwischen Inhalten, die zwischengespeichert werden können, und Inhalten, die nicht zwischengespeichert werden können:

- **Manifeste (nicht zwischenspeicherbar):** Wird verwendet, um alle Header AllViewer weiterzuleiten, die für dynamische Inhalte benötigt werden. Da Manifeste nicht zwischengespeichert werden, besteht kein Risiko einer Cache-Vergiftung.
- **Segmente (zwischenspeicherbar):** Verwenden Sie diese Option, um eine optimale None Leistung zu erzielen.
- **S3-Ursprünge:** CORS-S3Origin Für Amazon S3 S3-Buckets verwenden
- **MediaPackage Ursprünge:** Wird CORS-S3Origin für MediaPackage V2-Endpunkte verwendet



The screenshot shows the 'Origins' page in the AWS MediaTailor console. It includes a search bar, 'Edit', 'Delete', and 'Create origin' buttons, and a table with the following data:

	Origin name	Origin domain	Origin path	Origin type	Origin Shield region	Origin access
☐	ContentOrigin	zrljic.egress.kv9ydh.mediapackagev2.us-west-2.amazonaws...		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	-

Rangfolge und Konfiguration des Cache-Verhaltens

MediaTailor erfordert spezielle Konfigurationen für das Cache-Verhalten, um verschiedene Arten von Anfragen richtig verarbeiten zu können. Der Vorrang von Cache-Verhalten ist von entscheidender Bedeutung, da sie in der Reihenfolge (vom kleinsten zum größten) CDNs verarbeitet und das Verhalten für das erste passende Pfadmuster verwendet werden. Für die Problembehandlung ist es wichtig, diese Rangfolge zu verstehen:

- **Priorität 0:** Die meisten spezifischen Muster (like/tm/*) werden zuerst bewertet
- **Höhere Prioritätszahlen:** Weniger spezifische Muster werden der Reihe nach bewertet
- **Standardverhalten:** Fängt alle Anfragen ab, die nicht mit anderen Mustern übereinstimmen

Wenn sich Anfragen nicht wie erwartet verhalten, überprüfen Sie, ob sich Ihre Pfadmuster nicht unbeabsichtigt überschneiden.

General	Security	Origins	Behaviors	Error pages	Invalidations	Tags	Logging
Behaviors Filter behaviors by property or value Save Move up Move down Edit Delete Create behavior							
	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

Priorität 0: Pfadverhalten von Anzeigensegmenten

Dieses Verhalten verarbeitet die umgeleiteten Anfragen aus dem Segmentpfadverhalten und bedient die eigentlichen Anzeigensegmente. CloudFront wendet das folgende Verhalten auf alle Anfragen mit einem /tm/* Pfadmuster an. Dies ist das Verhalten mit der höchsten Priorität, da die Bereitstellung von Anzeigensegmenten für eine unterbrechungsfreie Wiedergabe entscheidend ist.

- Pfadmuster: /tm/*

Beispiel URLs , das diesem Muster entspricht:

- <https://your-distribution.cloudfront.net/tm/ad-segment-001.ts>
- <https://your-distribution.cloudfront.net/tm/transcoded-ad.m4s>
- Herkunft: Der Ursprung, den Sie mit der `segments.mediatailor.region.amazonaws.com` Domain erstellt haben.

Dies ist MediaTailorSegmentsim Beispiel im vorherigen Abschnitt zu Ursprüngen der Fall.

- Cache-Richtlinie: Managed-CachingOptimized

Einzelheiten darüber, was in der CloudFront verwalteten Cache-Richtlinie enthalten ist, finden Sie [CachingOptimized](#) im CloudFront Benutzerhandbuch. Sie können dieselben Einstellungen auch aus der verwalteten Richtlinie in Ihrem CDN eines Drittanbieters verwenden.

- Richtlinie für Origin-Anfragen: None
- Richtlinie für Antwort-Header: Managed-CORS-with-preflight-and-SecurityHeadersPolicy

Einzelheiten darüber, was in der Richtlinie für Antwortheader enthalten ist, finden Sie unter [CORS-with-preflight-and - SecurityHeadersPolicy](#) im CloudFront Benutzerhandbuch.

Anpassung dieser Einstellungen an andere CDNs

Wenn Sie ein anderes CDN als verwenden CloudFront, suchen Sie nach entsprechenden Einstellungen, die Folgendes bewirken.

Abgleich von Pfadmustern

Konfigurieren Sie ein bestimmtes Verhalten für das `/tm/*` Pfadmuster zur Verarbeitung von MediaTailor Anzeigensegmenten

Konfiguration des Cache-Schlüssels

Nehmen Sie den `Origin` Header in Ihren Cache-Schlüssel auf, um sicherzustellen, dass Antworten für unterschiedliche Ursprünge getrennt zwischengespeichert werden

Header-Weiterleitung

Leitet den `Origin` Header und andere CORS-bezogene Header an den Ursprung weiter

Verwaltung der Antwort-Header

Konfigurieren Sie Ihr CDN so, dass der `Access-Control-Allow-Origin` Header in Antworten vorhanden ist

Die spezifische Terminologie und die Konfigurationsoptionen variieren je nach CDN-Anbieter, die zugrunde liegenden Prinzipien bleiben jedoch dieselben.

Priorität 1: Servergesteuertes Verhalten bei der Anzeigeneinblendung

Dieses Verhalten verarbeitet Anfragen [MediaTailor Überblick und Implementierung der servergesteuerten Anzeigeneinfügung](#) (SGAI), wenn Kunden den geführten Modus konfigurieren, der zwischenspeicherbare Manifeste bereitstellt. CloudFront wendet das folgende Verhalten auf alle Anfragen mit einem `/v1/i-media/*` Pfadmuster an. SGA1 ermöglicht eine bessere Caching-Leistung, da die Manifeste nicht betrachterspezifisch sind.

- Pfadmuster: `/v1/i-media/*` (iMedia-Pfad für SGA1)

Beispiel URLs , das diesem Muster entspricht:

- `https://your-distribution.cloudfront.net/v1/i-media/your-config/manifest.m3u8`
- `https://your-distribution.cloudfront.net/v1/i-media/your-config/playlist.mpd`

- **Herkunft:** Der Ursprung, den Sie mit der `manifests.mediatailor.region.amazonaws.com` Domain erstellt haben.

Dies ist MediaTailorManifests im Beispiel im vorherigen Abschnitt zu Ursprüngen der Fall.

- **Cache-Richtlinie:** `Managed-CachingOptimized`

Einzelheiten darüber, was in der CloudFront verwalteten Cache-Richtlinie enthalten ist, finden Sie [CachingOptimized](#) im CloudFront Benutzerhandbuch. Sie können dieselben Einstellungen auch aus der verwalteten Richtlinie in Ihrem CDN eines Drittanbieters verwenden.

- **Richtlinie für Origin-Anfragen:** `None`
- **Richtlinie für Antwort-Header:** `Managed-CORS-with-preflight-and-SecurityHeadersPolicy`

Einzelheiten darüber, was in der Richtlinie für Antwortheader enthalten ist, finden Sie unter [CORS-with-preflight-and - SecurityHeadersPolicy](#) im CloudFront Benutzerhandbuch.

Priorität 2: Personalisiertes Manifestverhalten

Dieses Verhalten verarbeitet personalisierte Manifestanfragen. CloudFront wendet die folgenden Verhaltensweisen auf alle Anfragen mit einem `/v1/*` Pfadmuster an. CloudFront wendet die folgenden Verhaltensweisen an und speichert keine personalisierten Manifeste im Cache, da sie betrachterspezifische Anzeigeninhalte enthalten. URLs Dieses Verhalten gilt für alle Anfragen, die ein `/v1/*` Pfadmuster haben. Dies ist die MediaTailor Kernfunktion, bei der jeder Zuschauer ein einzigartiges Manifest mit personalisierter Anzeigeneinblendung erhält.

- **Pfadmuster:** `/v1/*` (MediaTailor Standard-V1-Anfragen)

Beispiel URLs , das diesem Muster entspricht:

- `https://your-distribution.cloudfront.net/v1/master/your-config/manifest.m3u8`
- `https://your-distribution.cloudfront.net/v1/dash/your-config/manifest.mpd`
- **Herkunft:** Der Ursprung, den Sie mit der `manifests.mediatailor.region.amazonaws.com` Domain erstellt haben.

Dies ist MediaTailorManifests im Beispiel im vorherigen Abschnitt zu Ursprüngen der Fall.

- **Cache-Richtlinie:** `Managed-CachingDisabled`

Einzelheiten darüber, was in der Cache-Richtlinie enthalten ist, finden Sie [CachingDisabled](#) im CloudFront Benutzerhandbuch.

- Richtlinie für Origin-Anfragen: AllViewer

Verwenden Sie bei personalisierten Manifesten die AllViewer Richtlinie, um alle Header weiterzuleiten, die für dynamische Inhalte benötigt werden.

- Richtlinie für Antwort-Header: Managed-CORS-with-preflight-and-SecurityHeadersPolicy

Einzelheiten darüber, was in der Richtlinie für Antwortheader enthalten ist, finden Sie unter [CORS-with-preflight-and - SecurityHeadersPolicy](#) im CloudFront Benutzerhandbuch.

Accept-Encoding-Header

Wir empfehlen, dass Ihr CDN den Header vor Zuschauern bewahrt. Accept-Encoding Dieser Header enthält MediaTailor Anweisungen zum Komprimieren personalisierter Manifeste.

In CloudFront beinhaltet die AllViewerAndCloudFrontHeaders Quellenforderungsrichtlinie die Weiterleitung des Accept-Encoding Headers durch den Betrachter. Wenn Sie ein anderes CDN verwenden, stellen Sie sicher, dass dieser Header beibehalten wird.

Im Folgenden wird beschrieben, wie mit dem Header MediaTailor umgegangen wird. Accept-Encoding

- Ältere Smart-Geräte: Ältere Smart-Geräte TVs , die Gzip nicht unterstützen, senden den Accept-Encoding-Header nicht und geben daher MediaTailor unkomprimierte Manifeste zurück
- Moderne Geräte: iPhones, Chrome-Browser und andere moderne Clients senden den Accept-Encoding-Header, sodass Manifeste vor der Auslieferung komprimiert werden können MediaTailor

Priorität 3: Verhalten des serverseitigen Beacon-Pfads

Dieses Verhalten verarbeitet Anfragen an, MediaTailor die zu Weiterleitungen für führen.

[Serverseitiges Tracking](#) Diese Anfragen sind für das Tracking von Beacons unerlässlich, daher muss jede Anfrage von MediaTailor bearbeitet werden. CloudFront wendet das folgende Verhalten auf alle Anfragen mit einem /segment/* Pfadmuster an.

- Pfadmuster: /segment/*

Beispiel URLs , das diesem Muster entspricht:

- `https://your-distribution.cloudfront.net/segment/tracking-beacon-123`
- `https://your-distribution.cloudfront.net/segment/ad-request-456.ts`
- Herkunft: Der Ursprung, den Sie mit der manifests.mediatailor.**region**.amazonaws.com Domain erstellt haben.

Dies ist MediaTailorManifests im Beispiel im vorherigen Abschnitt zu Ursprüngen der Fall.

- Cache-Richtlinie: Managed-CachingDisabled

Einzelheiten darüber, was in der Cache-Richtlinie enthalten ist, finden Sie [CachingDisabled](#) im CloudFront Benutzerhandbuch.

- Richtlinie für Origin-Anfragen: AllViewer

Verwenden Sie bei serverseitigen Beacon-Anfragen die AllViewer Richtlinie, um alle Header weiterzuleiten, die für die Nachverfolgung benötigt werden. Da diese Anfragen nicht zwischengespeichert werden, besteht kein Risiko einer Cache-Vergiftung.

- Richtlinie für Antwort-Header: Managed-CORS-with-preflight-and-SecurityHeadersPolicy

Einzelheiten darüber, was in der Richtlinie für Antwortheader enthalten ist, finden Sie unter [CORS-with-preflight-and - SecurityHeadersPolicy](#) im CloudFront Benutzerhandbuch.

Bei der MediaTailor Verarbeitung dieser Anfragen wird eine 302-Umleitungsantwort mit einem Pfad zurückgegeben, der auf die tatsächliche Segmentposition verweist. Eine Anfrage an `/segment/ad123.ts` könnte beispielsweise `/tm/encoded-ad-segment.ts` auf den Ursprung des Segments umgeleitet werden.

Priorität 4: Verhalten beim Herkunftspfad des Inhalts

Wenn der Anforderungspfad keinem der anderen Muster entspricht, wird das Standardverhalten CloudFront angewendet. Bei diesem Verhalten werden Anfragen direkt an den Inhaltsursprung gesendet, ohne dass sie von dort verarbeitet MediaTailor werden. Dies ermöglicht bei Bedarf direkten Zugriff auf Ihren Inhaltsursprung (z. B. MediaPackage V2). CloudFront wendet das folgende Verhalten auf alle Anfragen an, die keines der vorherigen Pfadmuster enthalten.

- Pfadmuster: (*)

- **Herkunft:** Der Ursprung, den Sie mit der Domain für die Herkunft Ihrer Inhalte erstellt haben.

Dies ist ContentOriginim Beispiel im vorherigen Abschnitt zu Ursprüngen der Fall.

- **Cache-Richtlinie:** Managed-CachingOptimized

Einzelheiten darüber, was in der CloudFront verwalteten Cache-Richtlinie enthalten ist, finden Sie [CachingOptimized](#) im CloudFront Benutzerhandbuch. Sie können dieselben Einstellungen auch aus der verwalteten Richtlinie in Ihrem CDN eines Drittanbieters verwenden.

 Note

Für HLS-Implementierungen mit niedriger Latenz sollten Sie erwägen, anstelle der Standardrichtlinie eine benutzerdefinierte Caching-Richtlinie mit HLS-Anweisungen (LLH) mit niedriger Latenz zu verwenden. CachingOptimized

- **Richtlinie für Origin-Anfragen:** None
- **Richtlinie für Antwort-Header:** Managed-CORS-with-preflight-and-SecurityHeadersPolicy

Obwohl das Standardverhalten bei der Herkunft von Inhalten in der Regel nicht den gleichen Risiken ausgesetzt ist wie beim Verhalten des Anzeigensegments, wird dennoch empfohlen, die Richtlinie für Managed-CORS-with-preflight-and-SecurityHeadersPolicy Antwortheader zu verwenden und den Origin Header in Ihren Cache-Schlüssel aufzunehmen. Dadurch wird eine konsistente CORS-Verarbeitung für alle Inhaltstypen gewährleistet und potenzielle Wiedergabeprobleme in webbasierten Playern vermieden.

Bei Inhaltssegmenten bietet die Managed-CachingOptimized Cache-Richtlinie eine gute Leistung, während die Managed-CORS-with-preflight-and-SecurityHeadersPolicy Response-Header-Richtlinie die korrekte CORS-Behandlung gewährleistet. Diese Kombination ermöglicht effizientes Caching bei gleichzeitiger Wahrung der Kompatibilität mit webbasierten Playern, die CORS-Header benötigen.

Die Anwendung einer konsistenten CORS-Handhabung sowohl für Anzeigensegmente als auch für Inhaltssegmente sorgt für ein zuverlässigeres Wiedergabeerlebnis und vereinfacht die Fehlerbehebung. Ohne die richtige CORS-Konfiguration kann es bei Spielern zu inkonsistentem Verhalten kommen, wenn sie zwischen Inhalten und Anzeigen wechseln.

Einzelheiten dazu, was in der Richtlinie für Antwort-Header enthalten ist, findest du unter [CORS-with-preflight-and — SecurityHeadersPolicy](#) im Benutzerhandbuch. CloudFront

Holen Sie sich Support und Hilfe zur Fehlerbehebung für CDN und Integrationen MediaTailor

AWS Elemental MediaTailor CDN-Integrationsprobleme können komplex zu diagnostizieren und zu lösen sein. Verwenden Sie dieses Handbuch, wenn Sie zusätzliche Hilfe bei CDN- und MediaTailor Integrationsproblemen benötigen, die Sie nicht durch Self-Service-Fehlerbehebung lösen können.

Probieren Sie vor der Eskalation die folgenden Self-Service-Optionen aus:

- Folgen Sie den Schritten zur Fehlerbehebung unter [Beheben Sie Probleme mit der CDN-Integration](#)
- Analysieren Sie Ihre Protokolle und Fehlercodes mit [Referenz zur Analyse des CDN-Integrationsprotokolls](#)
- Überprüfen Sie Ihr Monitoring-Setup mit [Richten Sie Überwachungstools ein](#)
- Informationen zu dienstspezifischen Problemen finden Sie im [Leitfaden MediaTailor zur Fehlerbehebung](#)
- Suchen Sie in [AWS re:Post](#) nach ähnlichen Problemen und Community-Lösungen
- Informationen zu gängigen Integrationsmustern finden Sie im [AWS Knowledge Center](#)

Themen

- [Wann sollten MediaTailor CDN-Probleme an den AWS-Support weitergeleitet werden](#)
- [Sammeln Sie MediaTailor CDN-Informationen, bevor Sie sich an den Support wenden](#)
- [MediaTailor Ressourcen und Kanäle zur Unterstützung der CDN-Integration](#)
- [MediaTailor Bewährte Verfahren zur Unterstützung der CDN-Integration](#)

Wann sollten MediaTailor CDN-Probleme an den AWS-Support weitergeleitet werden

AWS Elemental MediaTailor Probleme mit der CDN-Integration sollten an den AWS Support weitergeleitet werden, wenn die Self-Service-Fehlerbehebung das Problem nicht löst. Erwägen Sie, in folgenden Fällen an den AWS Support zu eskalieren:

Hinweis: Für technischen Support ist ein kostenpflichtiger AWS Supportplan erforderlich. Einzelheiten zum Supportplan und Reaktionszeiten finden Sie unter [AWS-Supportpläne](#).

- Probleme wirken sich auf den Produktionsverkehr oder den Umsatz aus
- Sie haben alle Schritte zur Fehlerbehebung befolgt, ohne dass eine Lösung gefunden wurde
- Fehlermuster deuten auf Probleme auf Service-Level hin
- Sie benötigen Unterstützung bei komplexen Konfigurationsszenarien
- Trotz Optimierungsbemühungen bestehen weiterhin Leistungsprobleme

Sammeln Sie MediaTailor CDN-Informationen, bevor Sie sich an den Support wenden

AWS Elemental MediaTailor Support-Anfragen zur CDN-Integration erfordern spezifische Konfiguration und Diagnoseinformationen für eine effektive Fehlerbehebung. Bevor Sie sich an den AWS Support wenden, sollten Sie die folgenden Informationen sammeln, um die Problemlösung zu beschleunigen:

Tipp: Wenn Sie diese Informationen bereithalten, bevor Sie Ihren Support-Fall erstellen, können Sie die Lösungszeit erheblich verkürzen und den Support-Technikern helfen, Ihre spezifische Konfiguration besser zu verstehen.

- MediaTailor Einzelheiten zur Konfiguration:
 - Name und AWS Region der Konfiguration
 - Konfiguration der Wiedergabe ARN
 - ADS-URL und Integrationstyp
- Einzelheiten zur CDN-Konfiguration:
 - CDN-Verteilungs-ID oder Konfigurationsname
 - Konfigurationen des Cache-Verhaltens für Manifeste und Segmente

- Origin-Konfiguration und Routing-Regeln
- Informationen zum Fehler:
 - Spezifische Fehlermeldungen und HTTP-Statuscodes
 - Zeitstempel, wenn Probleme auftreten
 - Beispiel URLs , das das Problem verdeutlicht
 - CDN- und MediaTailor Protokolleinträge im Zusammenhang mit dem Problem
- Informationen zum Testen:
 - Schritte, die Sie bereits zur Fehlerbehebung unternommen haben
 - Geräte und Player, bei denen das Problem auftritt
 - Ob das Problem alle Inhalte oder bestimmte Streams betrifft
 - Häufigkeit und Muster des Problems (intermittierend, konsistent, zeitabhängig)

MediaTailor Ressourcen und Kanäle zur Unterstützung der CDN-Integration

AWS Elemental MediaTailor Der CDN-Integrationssupport ist über mehrere Kanäle verfügbar, um Ihnen bei der Lösung von Problemen und der Optimierung Ihrer Implementierung zu helfen:

- AWS Support: Erstellen Sie über die AWS Management Console einen Support-Fall, um technische Unterstützung zu erhalten

Zugang: [AWS Support Center](#)

- AWS re:POST: Von der Community betriebene Q&A-Plattform für AWS verwandte Fragen und Community-Support

Zugang [AWS](#) : re:POST

- AWS Dokumentation: Umfassende Anleitungen für MediaTailor und CDN-Dienste

Zugang: [MediaTailorDokumentation und CloudFront Dokumentation](#)

- AWS Schulung: Kurse und Zertifizierungen für Mediendienste und CDN-Optimierung

Zugang: [AWS Schulung und Zertifizierung](#)

- AWS Wissenszentrum: Kuratierte Artikel zu häufig auftretenden AWS Problemen und bewährten Verfahren

Zugang: [AWS Knowledge Center](#)

- AWS Trusted Advisor: Automatisierte Optimierungsempfehlungen und Best Practices

Zugang: [AWS Trusted Advisor](#)

- AWS Personal Health Dashboard: Benachrichtigungen zum Servicestatus und zur Wartung

Zugang: [AWS Personal Health Dashboard](#)

MediaTailor Bewährte Verfahren zur Unterstützung der CDN-Integration

AWS Elemental MediaTailor Supportfälle bei der CDN-Integration werden effizienter gelöst, wenn Sie diese bewährten Methoden befolgen. So erhalten Sie die schnellste Lösung:

- Wählen Sie den geeigneten Schweregrad auf der Grundlage der Auswirkungen auf das Unternehmen
- Geben Sie bei der ersten Einreichung des Falls alle relevanten Informationen an
- Fügen Sie spezifische Beispiele und Reproduktionsschritte hinzu
- Hängen Sie relevante Protokolldateien und Konfigurations-Screenshots an
- Reagieren Sie umgehend auf Anfragen von Support-Technikern nach zusätzlichen Informationen

Zusätzliche Supportressourcen:

- [AWS Support Case Management](#) — Leitfaden zur Erstellung und Verwaltung von Support-Fällen
- [AWS Supportpläne](#) — Vergleichen Sie die Funktionen und Reaktionszeiten des Supportplans
- [Bereiten Sie sich auf die Unterstützung von Workloads](#) vor — Well-Architected Leitlinien für die Betriebsbereitschaft
- [AWS Well-Architected Framework](#) — Bewährte Methoden für die Erstellung und den Betrieb von Workloads auf AWS

Sicherheit in AWS Elemental MediaTailor

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

Sicherheit ist eine gemeinsame AWS Verantwortung von Ihnen und Ihnen. Das [Modell der geteilten Verantwortung](#) beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

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

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

Topics

- [Datenschutz in AWS Elemental MediaTailor](#)
- [Identity and Access Management für AWS Elemental MediaTailor](#)
- [Überprüfung der Einhaltung der Vorschriften für AWS Elemental MediaTailor](#)
- [Resilienz in AWS Elemental MediaTailor](#)
- [Sicherheit der Infrastruktur in MediaTailor](#)
- [Serviceübergreifende Confused-Deputy-Prävention](#)
- [Einloggen und Überwachen MediaTailor](#)

Datenschutz in AWS Elemental MediaTailor

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

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

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Wird verwendet SSL/TLS , um mit AWS Ressourcen zu kommunizieren. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein AWS CloudTrail. Informationen zur Verwendung von CloudTrail Pfaden zur Erfassung von AWS Aktivitäten finden Sie unter [Arbeiten mit CloudTrail Pfaden](#) im AWS CloudTrail Benutzerhandbuch.
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-3-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter [Federal Information Processing Standard \(FIPS\) 140-3](#).

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit der Konsole, der MediaTailor API oder auf andere AWS-Services Weise arbeiten oder diese verwenden. AWS CLI AWS SDKs Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die

für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Datenverschlüsselung

AWS Elemental MediaTailor verschlüsselt oder entschlüsselt keine Daten bei der Verwaltung von Inhaltsverzeichnissen oder bei der Kommunikation mit Servern oder CDNs Playern. MediaTailor verlangt nicht, dass Sie Kundendaten oder andere vertrauliche Informationen angeben.

Geben Sie keine vertraulichen Informationen wie Kundenkontonummern, Kreditkarteninformationen oder Anmeldeinformationen in Freiformfelder oder Abfrageparameter ein. Dies gilt für die gesamte Verwendung von AWS Elemental MediaTailor, einschließlich der Konsole, der API und der AWS Command Line Interface (AWS CLI). SDKs Alle Daten, die Sie in den Service eingeben, werden möglicherweise in Diagnoseprotokolle aufgenommen.

Wenn Sie eine URL für einen externen Server bereitzustellen, schließen Sie keine unverschlüsselten Anmeldeinformationen zur Validierung Ihrer Anforderung bei dem betreffenden Server ein.

Identity and Access Management für AWS Elemental MediaTailor

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

Themen

- [Zielgruppe](#)
- [Authentifizierung mit Identitäten](#)
- [Verwalten des Zugriffs mit Richtlinien](#)
- [Wie AWS Elemental MediaTailor funktioniert mit IAM](#)
- [Beispiele für identitätsbasierte Richtlinien für AWS Elemental MediaTailor](#)
- [Beispiele für ressourcenbasierte Richtlinien für AWS Elemental MediaTailor](#)

- [AWS verwaltete Richtlinien für AWS Elemental MediaTailor](#)
- [Verwenden von serviceverknüpften Rollen für MediaTailor](#)
- [Fehlerbehebung bei AWS Elemental MediaTailor Identität und Zugriff](#)

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von Ihrer Rolle ab:

- Servicebenutzer – Fordern Sie von Ihrem Administrator Berechtigungen an, wenn Sie nicht auf Features zugreifen können (siehe [Fehlerbehebung bei AWS Elemental MediaTailor Identität und Zugriff](#)).
- Serviceadministrator – Bestimmen Sie den Benutzerzugriff und stellen Sie Berechtigungsanfragen (siehe [Wie AWS Elemental MediaTailor funktioniert mit IAM](#)).
- IAM-Administrator – Schreiben Sie Richtlinien zur Zugriffsverwaltung (siehe [Beispiele für identitätsbasierte Richtlinien für AWS Elemental MediaTailor](#)).

Authentifizierung mit Identitäten

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

Sie können sich als föderierte Identität anmelden, indem Sie Anmeldeinformationen aus einer Identitätsquelle wie AWS IAM Identity Center (IAM Identity Center), Single Sign-On-Authentifizierung oder Anmeldeinformationen verwenden. Google/Facebook Weitere Informationen zum Anmelden finden Sie unter [So melden Sie sich bei Ihrem AWS-Konto an](#) im Benutzerhandbuch für AWS-Anmeldung .

AWS Bietet für den programmatischen Zugriff ein SDK und eine CLI zum kryptografischen Signieren von Anfragen. Weitere Informationen finden Sie unter [AWS Signature Version 4 for API requests](#) im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie einen erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, dem sogenannten AWS-Konto Root-Benutzer, der vollständigen Zugriff auf alle AWS-Services Ressourcen hat. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Eine Liste der

Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter [Tasks that require root user credentials](#) im IAM-Benutzerhandbuch.

Verbundidentität

Es hat sich bewährt, dass menschliche Benutzer für den Zugriff AWS-Services mithilfe temporärer Anmeldeinformationen einen Verbund mit einem Identitätsanbieter verwenden müssen.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensverzeichnis, Ihrem Directory Service Web-Identitätsanbieter oder der AWS-Services mithilfe von Anmeldeinformationen aus einer Identitätsquelle zugreift. Verbundene Identitäten übernehmen Rollen, die temporäre Anmeldeinformationen bereitstellen.

Für die zentrale Zugriffsverwaltung empfehlen wir AWS IAM Identity Center. Weitere Informationen finden Sie unter [Was ist IAM Identity Center?](#) im AWS IAM Identity Center -Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein [IAM-Benutzer](#) ist eine Identität mit bestimmten Berechtigungen für eine einzelne Person oder Anwendung. Wir empfehlen die Verwendung temporärer Anmeldeinformationen anstelle von IAM-Benutzern mit langfristigen Anmeldeinformationen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter [Erfordern, dass menschliche Benutzer den Verbund mit einem Identitätsanbieter verwenden müssen, um AWS mithilfe temporärer Anmeldeinformationen darauf zugreifen zu](#) können.

Eine [IAM-Gruppe](#) spezifiziert eine Sammlung von IAM-Benutzern und erleichtert die Verwaltung von Berechtigungen für große Gruppen von Benutzern. Weitere Informationen finden Sie unter [Anwendungsfälle für IAM-Benutzer](#) im IAM-Benutzerhandbuch.

IAM-Rollen

Eine [IAM-Rolle](#) ist eine Identität mit spezifischen Berechtigungen, die temporäre Anmeldeinformationen bereitstellt. Sie können eine Rolle übernehmen, indem Sie [von einer Benutzer- zu einer IAM-Rolle \(Konsole\) wechseln](#) AWS CLI oder einen AWS API-Vorgang aufrufen. Weitere Informationen finden Sie unter [Methoden, um eine Rolle zu übernehmen](#) im IAM-Benutzerhandbuch.

IAM-Rollen sind nützlich für den Verbundbenutzer-Zugriff, temporäre IAM-Benutzerberechtigungen, kontoübergreifenden Zugriff, serviceübergreifenden Zugriff und Anwendungen, die auf Amazon EC2 laufen. Weitere Informationen finden Sie unter [Kontoübergreifender Ressourcenzugriff in IAM](#) im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie definiert Berechtigungen, wenn sie mit einer Identität oder Ressource verknüpft sind. AWS bewertet diese Richtlinien, wenn ein Principal eine Anfrage stellt. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu JSON-Richtliniendokumenten finden Sie unter [Übersicht über JSON-Richtlinien](#) im IAM-Benutzerhandbuch.

Mit Hilfe von Richtlinien legen Administratoren fest, wer Zugriff auf was hat, indem sie definieren, welches Prinzipal welche Aktionen auf welchen Ressourcen und unter welchen Bedingungen durchführen darf.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator erstellt IAM-Richtlinien und fügt sie zu Rollen hinzu, die die Benutzer dann übernehmen können. IAM-Richtlinien definieren Berechtigungen unabhängig von der Methode, die zur Ausführung der Operation verwendet wird.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität (Benutzer, Gruppe oder Rolle) anfügen können. Diese Richtlinien steuern, welche Aktionen Identitäten für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien](#) im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können Inline-Richtlinien (direkt in eine einzelne Identität eingebettet) oder verwaltete Richtlinien (eigenständige Richtlinien, die mit mehreren Identitäten verbunden sind) sein. Informationen dazu, wie Sie zwischen verwalteten und Inline-Richtlinien wählen, finden Sie unter [Choose between managed policies and inline policies](#) im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele hierfür sind Vertrauensrichtlinien für IAM-Rollen und Amazon S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#).

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

Weitere Richtlinientypen

AWS unterstützt zusätzliche Richtlinientypen, mit denen die maximalen Berechtigungen festgelegt werden können, die durch gängigere Richtlinientypen gewährt werden:

- **Berechtigungsgrenzen** – Eine Berechtigungsgrenze legt die maximalen Berechtigungen fest, die eine identitätsbasierte Richtlinie einer IAM-Entität erteilen kann. Weitere Informationen finden Sie unter [Berechtigungsgrenzen für IAM-Entitäten](#) im -IAM-Benutzerhandbuch.
- **Richtlinien zur Dienstkontrolle (SCPs)** — Geben Sie die maximalen Berechtigungen für eine Organisation oder Organisationseinheit in an AWS Organizations. Weitere Informationen finden Sie unter [Service-Kontrollrichtlinien](#) im AWS Organizations -Benutzerhandbuch.
- **Richtlinien zur Ressourcenkontrolle (RCPs)** — Legen Sie die maximal verfügbaren Berechtigungen für Ressourcen in Ihren Konten fest. Weitere Informationen finden Sie im AWS Organizations Benutzerhandbuch unter [Richtlinien zur Ressourcenkontrolle \(RCPs\)](#).
- **Sitzungsrichtlinien** – Sitzungsrichtlinien sind erweiterte Richtlinien, die als Parameter übergeben werden, wenn Sie eine temporäre Sitzung für eine Rolle oder einen Verbundbenutzer erstellen. Weitere Informationen finden Sie unter [Sitzungsrichtlinien](#) im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn für eine Anfrage mehrere Arten von Richtlinien gelten, sind die sich daraus ergebenden Berechtigungen schwieriger zu verstehen. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie unter [Bewertungslogik für Richtlinien](#) im IAM-Benutzerhandbuch.

Wie AWS Elemental MediaTailor funktioniert mit IAM

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

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

IAM-Feature	MediaTailor Unterstützung
Identitätsbasierte Richtlinien	Ja

IAM-Feature	MediaTailor Unterstützung
Ressourcenbasierte Richtlinien	Ja
Richtlinienaktionen	Ja
Richtlinienressourcen	Nein
Richtlinienbedingungsschlüssel (servicespezifisch)	Ja
ACLs	Nein
ABAC (Tags in Richtlinien)	Teilweise
Temporäre Anmeldeinformationen	Ja
Prinzipalberechtigungen	Ja
Servicerollen	Nein
Serviceverknüpfte Rollen	Ja

Einen allgemeinen Überblick darüber, wie MediaTailor und andere AWS Dienste mit den meisten IAM-Funktionen funktionieren, finden Sie im [IAM-Benutzerhandbuch unter AWS Dienste, die mit IAM funktionieren](#).

Identitätsbasierte Richtlinien für MediaTailor

Unterstützt Richtlinien auf Identitätsbasis: Ja

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien](#) im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter

denen Aktionen zugelassen oder abgelehnt werden. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der [IAM-Referenz für JSON-Richtlinienelemente](#) im IAM-Benutzerhandbuch.

Beispiele für identitätsbasierte Richtlinien für MediaTailor

Beispiele für MediaTailor identitätsbasierte Richtlinien finden Sie unter. [Beispiele für identitätsbasierte Richtlinien für AWS Elemental MediaTailor](#)

Ressourcenbasierte Richtlinien finden Sie in MediaTailor

Unterstützt ressourcenbasierte Richtlinien: Ja

Der MediaTailor Dienst unterstützt nur einen Typ von ressourcenbasierten Richtlinien. Sie wird als Kanalrichtlinie bezeichnet, weil sie mit einem Kanal verknüpft ist. Diese Richtlinie definiert, welche Principals Aktionen auf dem Kanal ausführen können.

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

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Weitere Informationen finden Sie unter [Kontoübergreifender Ressourcenzugriff in IAM](#) im IAM-Benutzerhandbuch.

Informationen zum Anhängen einer ressourcenbasierten Richtlinie an einen Channel finden Sie unter. [Erstellen Sie mit der MediaTailor Konsole einen Kanal](#)

Beispiele für ressourcenbasierte Richtlinien finden Sie MediaTailor

Beispiele für MediaTailor ressourcenbasierte Richtlinien finden Sie unter. [Beispiele für ressourcenbasierte Richtlinien für AWS Elemental MediaTailor](#)

Politische Maßnahmen für MediaTailor

Unterstützt Richtlinienaktionen: Ja

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

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Nehmen Sie Aktionen in eine Richtlinie auf, um Berechtigungen zur Ausführung des zugehörigen Vorgangs zu erteilen.

Eine Liste der MediaTailor Aktionen finden Sie unter [Aktionen definiert von AWS Elemental MediaTailor](#) in der Serviceautorisierungsreferenz.

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

```
mediatailor
```

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

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

Beispiele für MediaTailor identitätsbasierte Richtlinien finden Sie unter [Beispiele für identitätsbasierte Richtlinien für AWS Elemental MediaTailor](#)

Politische Ressourcen für MediaTailor

Unterstützt Richtlinienressourcen: Nein

AWS Elemental MediaTailor unterstützt die Angabe von Ressourcen ARNs in einer Richtlinie nicht.

Schlüssel zur Richtlinienbedingung für MediaTailor

Unterstützt servicespezifische Richtlinienbedingungsschlüssel: Ja

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

Das Element `Condition` gibt an, wann Anweisungen auf der Grundlage definierter Kriterien ausgeführt werden. Sie können bedingte Ausdrücke erstellen, die [Bedingungsoperatoren](#) verwenden, z. B. `ist gleich` oder `kleiner als`, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter [Kontextschlüssel für AWS globale Bedingungen](#) im IAM-Benutzerhandbuch.

Eine Liste der MediaTailor Bedingungsschlüssel finden Sie unter [Bedingungsschlüssel für AWS Elemental MediaTailor](#) in der Service Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter [Aktionen definiert von AWS Elemental MediaTailor](#).

AWS Elemental MediaTailor stellt keine dienstspezifischen Bedingungsschlüssel bereit, unterstützt aber die Verwendung einiger globaler Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter [AWS Globale Bedingungskontextschlüssel](#) im AWS Identity and Access Management Benutzerhandbuch.

ACLs in MediaTailor

Unterstützt ACLs: Nein

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

ABAC mit MediaTailor

Unterstützt ABAC (Tags in Richtlinien): Teilweise

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen, auch als Tags bezeichnet, definiert werden. Sie können Tags an IAM-Entitäten und AWS -Ressourcen anhängen und dann ABAC-Richtlinien entwerfen, die Operationen zulassen, wenn das Tag des Prinzipals mit dem Tag auf der Ressource übereinstimmt.

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

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

Weitere Informationen zu ABAC finden Sie unter [Definieren von Berechtigungen mit ABAC-Autorisierung](#) im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe [Attributbasierte Zugriffskontrolle \(ABAC\)](#) verwenden im IAM-Benutzerhandbuch.

Verwenden Sie für MediaTailor den Wert Partial.

Verwenden Sie temporäre Anmeldeinformationen mit MediaTailor

Unterstützt temporäre Anmeldeinformationen: Ja

Temporäre Anmeldeinformationen ermöglichen kurzfristigen Zugriff auf AWS Ressourcen und werden automatisch erstellt, wenn Sie einen Verbund verwenden oder die Rollen wechseln. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter [Temporäre Anmeldeinformationen in IAM](#) und [AWS-Services , die mit IAM funktionieren](#) im IAM-Benutzerhandbuch.

Serviceübergreifende Prinzipalberechtigungen für MediaTailor

Unterstützt Forward Access Sessions (FAS): Ja

Forward Access Sessions (FAS) verwenden die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, in Kombination mit der Anforderung, Anfragen an nachgelagerte Dienste AWS-Service zu stellen. Einzelheiten zu den Richtlinien für FAS-Anforderungen finden Sie unter [Zugriffssitzungen weiterleiten](#).

Servicerollen für MediaTailor

Unterstützt Servicerollen: Nein

AWS Elemental MediaTailor unterstützt keine Servicerollen.

Mit Diensten verknüpfte Rollen für MediaTailor

Unterstützt serviceverknüpfte Rollen: Ja

Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Dienstbezogene

Rollen werden in Ihrem Dienst angezeigt AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

Einzelheiten zum Erstellen oder Verwalten von MediaTailor dienstbezogenen Rollen finden Sie unter [Verwenden von serviceverknüpften Rollen für MediaTailor](#)

Beispiele für identitätsbasierte Richtlinien für AWS Elemental MediaTailor

Benutzer und Rollen haben standardmäßig nicht die Berechtigung, MediaTailor-Ressourcen zu erstellen oder zu ändern. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter [Erstellen von IAM-Richtlinien \(Konsole\)](#) im IAM-Benutzerhandbuch.

Einzelheiten zu Aktionen und Ressourcentypen, die von definiert wurden MediaTailor, einschließlich des Formats von ARNs für die einzelnen Ressourcentypen, finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für AWS Elemental MediaTailor](#) in der Referenz zur Serviceautorisierung.

Themen

- [Best Practices für Richtlinien](#)
- [Verwenden der Konsole MediaTailor](#)
- [Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer](#)

Best Practices für Richtlinien

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

- Erste Schritte mit AWS verwalteten Richtlinien und Umstellung auf Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um damit zu beginnen, Ihren Benutzern und Workloads Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen

finden Sie unter [Von AWS verwaltete Richtlinien](#) oder [Von AWS verwaltete Richtlinien für Auftragsfunktionen](#) im IAM-Benutzerhandbuch.

- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter [Richtlinien und Berechtigungen in IAM](#) im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z. CloudFormation B. Weitere Informationen finden Sie unter [IAM-JSON-Richtlinienelemente: Bedingung](#) im IAM-Benutzerhandbuch.
- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter [Richtlinienvvalidierung mit IAM Access Analyzer](#) im IAM-Benutzerhandbuch.
- Multi-Faktor-Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter [Sicherer API-Zugriff mit MFA](#) im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter [Best Practices für die Sicherheit in IAM](#) im IAM-Benutzerhandbuch.

Verwenden der Konsole MediaTailor

Um auf die AWS Elemental MediaTailor Konsole zugreifen zu können, benötigen Sie ein Mindestmaß an Berechtigungen. Diese Berechtigungen müssen es Ihnen ermöglichen, Details zu den MediaTailor

Ressourcen in Ihrem aufzulisten und anzuzeigen AWS-Konto. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Entitäten (Benutzer oder Rollen) mit dieser Richtlinie.

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

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

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie umfasst Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der AWS CLI AWS OR-API.

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

```

```
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
    ],
    "Resource": "*"
}
]
```

Beispiele für ressourcenbasierte Richtlinien für AWS Elemental MediaTailor

Informationen zum Anhängen einer ressourcenbasierten Richtlinie an einen Channel finden Sie unter.

[Erstellen Sie mit der MediaTailor Konsole einen Kanal](#)

Themen

- [Anonymer Zugriff](#)
- [Kontoübergreifender Zugriff](#)

Anonymer Zugriff

Beachten Sie die folgende Allow Richtlinie. Wenn diese Richtlinie in Kraft ist, wird anonymer Zugriff auf die `mediatailor:GetManifest` Aktion auf der Kanalressource in der Richtlinie MediaTailor ermöglicht. Dies geschieht, wenn *region* der AWS-Region, *accountID* Ihre AWS-Konto ID und der Name der Kanalressource *channelName* ist.

Kontoübergreifender Zugriff

Beachten Sie die folgende Allow Richtlinie. Wenn diese Richtlinie in Kraft ist, MediaTailor ist die `mediatailor:GetManifest` Aktion für die Kanalressource in der Richtlinie kontenübergreifend möglich. Dies geschieht, wenn *region* der AWS-Region, *accountID* deine AWS-Konto ID und der Name der Kanalressource *channelName* ist.

AWS verwaltete Richtlinien für AWS Elemental MediaTailor

Um Benutzern, Gruppen und Rollen Berechtigungen hinzuzufügen, ist es einfacher, AWS verwaltete Richtlinien zu verwenden, als Richtlinien selbst zu schreiben. Es erfordert Zeit und Fachwissen, um [von Kunden verwaltete IAM-Richtlinien zu erstellen](#), die Ihrem Team nur die benötigten Berechtigungen bieten. Um schnell loszulegen, können Sie unsere AWS verwalteten Richtlinien verwenden. Diese Richtlinien decken allgemeine Anwendungsfälle ab und sind in Ihrem AWS-Konto verfügbar. Weitere Informationen zu AWS verwalteten Richtlinien finden Sie im IAM-Benutzerhandbuch unter [AWS Verwaltete Richtlinien](#).

AWS Dienste verwalten und aktualisieren AWS verwaltete Richtlinien. Sie können die Berechtigungen in AWS verwalteten Richtlinien nicht ändern. Services fügen einer von AWS verwalteten Richtlinien gelegentlich zusätzliche Berechtigungen hinzu, um neue Features zu unterstützen. Diese Art von Update betrifft alle Identitäten (Benutzer, Gruppen und Rollen), an welche die Richtlinie angehängt ist. Services aktualisieren eine von AWS verwaltete Richtlinie am ehesten, ein neues Feature gestartet wird oder neue Vorgänge verfügbar werden. Dienste entfernen keine Berechtigungen aus einer AWS verwalteten Richtlinie, sodass durch Richtlinienaktualisierungen Ihre bestehenden Berechtigungen nicht beeinträchtigt werden.

AWS Unterstützt außerdem verwaltete Richtlinien für Jobfunktionen, die sich über mehrere Dienste erstrecken. Die ReadOnlyAccess AWS verwaltete Richtlinie bietet beispielsweise schreibgeschützten Zugriff auf alle AWS Dienste und Ressourcen. Wenn ein Dienst eine neue Funktion startet, werden nur Leseberechtigungen für neue Operationen und Ressourcen AWS hinzugefügt. Eine Liste und Beschreibungen der Richtlinien für Auftragsfunktionen finden Sie in [Verwaltete AWS -Richtlinien für Auftragsfunktionen](#) im IAM-Leitfaden.

AWS verwaltete Richtlinie: AWSElementalMediaTailorFullAccess

Sie können die AWSElementalMediaTailorFullAccess-Richtlinie an Ihre IAM-Identitäten anfügen. Dies ist nützlich für Benutzer, die Wiedergabekonfigurationen und Ressourcen zur Kanalzusammenstellung wie Programme und Kanäle erstellen und verwalten müssen. Diese Richtlinie gewährt Berechtigungen, die vollen Zugriff auf ermöglichen AWS Elemental MediaTailor. Diese Benutzer können MediaTailor Ressourcen erstellen, aktualisieren und löschen.

Informationen zu den Berechtigungen für diese Richtlinie finden Sie unter [AWSElementalMediaTailorFullAccess](#) in der Referenz zu von AWS verwalteten Richtlinien.

AWS verwaltete Richtlinie: AWSElementalMediaTailorReadOnly

Sie können die AWSElementalMediaTailorReadOnly-Richtlinie an Ihre IAM-Identitäten anfügen. Dies ist nützlich für Benutzer, die Wiedergabekonfigurationen und Ressourcen zur

Kanalzusammenstellung wie Programme und Kanäle einsehen müssen. Diese Richtlinie gewährt Berechtigungen, die nur Lesezugriff auf ermöglichen. AWS Elemental MediaTailor Diese Benutzer können MediaTailor -Ressourcen nicht erstellen, aktualisieren oder löschen.

Informationen zu den Berechtigungen für diese Richtlinie finden Sie unter [AWSElementalMediaTailorReadOnly](#) in der Referenz zu von AWS verwalteten Richtlinien.

MediaTailor Aktualisierungen verwalteter Richtlinien AWS

Hier finden Sie Informationen zu Aktualisierungen der AWS verwalteten Richtlinien, die MediaTailor seit Beginn der Nachverfolgung dieser Änderungen durch diesen Dienst vorgenommen wurden. Um automatische Warnungen über Änderungen an dieser Seite erhalten, abonnieren Sie den RSS-Feed auf MediaTailor [Dokumentenverlauf für AWS Elemental MediaTailor](#).

Änderungen	Beschreibung	Date
MediaTailor neue verwaltete Richtlinien hinzugefügt	MediaTailor hat die folgenden verwalteten Richtlinien hinzugefügt: • AWSElementalMediaTailorReadOnly • AWSElementalMediaTailorFullAccess	24. November 2021
MediaTailor hat begonnen, Änderungen zu verfolgen	MediaTailor hat begonnen, Änderungen für die AWS verwalteten Richtlinien zu verfolgen.	24. November 2021

Verwenden von serviceverknüpften Rollen für MediaTailor

AWS Elemental MediaTailor verwendet AWS Identity and Access Management (IAM) [serviceverknüpfte](#) Rollen. Eine serviceverknüpfte Rolle ist ein einzigartiger Typ von IAM-Rolle, mit der direkt verknüpft ist. MediaTailor Mit Diensten verknüpfte Rollen sind vordefiniert MediaTailor und enthalten alle Berechtigungen, die der Dienst benötigt, um andere AWS Dienste in Ihrem Namen aufzurufen.

Eine dienstbezogene Rolle MediaTailor erleichtert die Einrichtung, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. MediaTailor definiert die Berechtigungen ihrer dienstbezogenen Rollen und MediaTailor kann, sofern nicht anders definiert, nur ihre Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem ihre verwandten Ressourcen gelöscht wurden. Dadurch werden Ihre MediaTailor Ressourcen geschützt, da Sie nicht versehentlich die Zugriffsberechtigung für die Ressourcen entziehen können.

Informationen zu anderen Services, die serviceverknüpften Rollen unterstützen, finden Sie unter [AWS -Services, die mit IAM funktionieren](#). Suchen Sie nach den Services, für die Ja in der Spalte Serviceverknüpfte Rolle angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Unterstützte Regionen für dienstbezogene Rollen MediaTailor

MediaTailor unterstützt die Verwendung von dienstbezogenen Rollen in allen Regionen, in denen der Dienst verfügbar ist. Weitere Informationen finden Sie unter [AWS -Regionen und Endpunkte](#).

Topics

- [Dienstbezogene Rollenberechtigungen für MediaTailor](#)
- [Erstellen einer dienstbezogenen Rolle für MediaTailor](#)
- [Bearbeiten einer dienstbezogenen Rolle für MediaTailor](#)
- [Löschen einer dienstbezogenen Rolle für MediaTailor](#)

Dienstbezogene Rollenberechtigungen für MediaTailor

MediaTailor verwendet die mit dem Dienst verknüpfte Rolle mit dem Namen `AWSServiceRoleForMediaTailor`— MediaTailor verwendet diese dienstverknüpfte Rolle, um Protokollgruppen, Protokolldatenströme und Protokollereignisse aufzurufen und CloudWatch zu verwalten. Diese verwaltete Richtlinie ist mit der folgenden serviceverknüpften Rolle verbunden: `AWSMediaTailorServiceRolePolicy`.

Die `AWSServiceRoleForMediaTailor` dienstverknüpfte Rolle vertraut darauf, dass die folgenden Dienste die Rolle übernehmen:

- `mediatailor.amazonaws.com`

Die Richtlinie für Rollenberechtigungen MediaTailor ermöglicht es, die folgenden Aktionen für die angegebenen Ressourcen durchzuführen:

- Aktion: `logs:PutLogEvents` für `arn:aws:logs:*:*:log-group:/aws/MediaTailor/*:log-stream:*`
- Aktion: `logs:CreateLogStream`, `logs:CreateLogGroup`, `logs:DescribeLogGroups`, `logs:DescribeLogStreams` für `arn:aws:logs:*:*:log-group:/aws/MediaTailor/*`

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter [serviceverknüpfte Rollenberechtigungen](#) im IAM-Benutzerhandbuch.

Erstellen einer dienstbezogenen Rolle für MediaTailor

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie die Sitzungsprotokollierung in der AWS-Managementkonsole, der AWS Command Line Interface (AWS CLI) oder der AWS API aktivieren, MediaTailor wird die dienstbezogene Rolle für Sie erstellt.

Important

Diese serviceverknüpfte Rolle kann in Ihrem Konto erscheinen, wenn Sie eine Aktion in einem anderen Service abgeschlossen haben, der die von dieser Rolle unterstützten Features verwendet. Außerdem gilt: Wenn Sie den MediaTailor Dienst vor dem 15. September 2021 genutzt haben, als er begann, dienstbezogene Rollen zu unterstützen, haben Sie die `AWSServiceRoleForMediaTailor` Rolle dann in Ihrem Konto MediaTailor erstellt. Weitere Informationen finden Sie unter [Eine neue Rolle ist in meinem IAM-Konto erschienen](#).

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie die Sitzungsprotokollierung aktivieren, MediaTailor wird die dienstbezogene Rolle erneut für Sie erstellt.

Sie können auch die IAM-Konsole verwenden, um eine serviceverknüpfte Rolle mit dem MediaTailor Anwendungsfall zu erstellen. Erstellen Sie in der AWS CLI oder der AWS API eine dienstverknüpfte Rolle mit dem `mediatailor.amazonaws.com` Dienstnamen. Weitere Informationen finden Sie unter [Erstellen einer serviceverknüpfte Rolle](#) im IAM-Leitfaden. Wenn

Sie diese serviceverknüpfte Rolle löschen, können Sie mit demselben Verfahren die Rolle erneut erstellen.

Bearbeiten einer dienstbezogenen Rolle für MediaTailor

MediaTailor erlaubt es Ihnen nicht, die AWSService RoleForMediaTailor dienstbezogene Rolle zu bearbeiten. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollename nach dem Erstellen einer serviceverknüpften Rolle nicht mehr geändert werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter [Bearbeiten einer serviceverknüpften Rolle](#) im IAM-Benutzerhandbuch.

Löschen einer dienstbezogenen Rolle für MediaTailor

Wenn Sie ein Feature oder einen Dienst, die bzw. der eine serviceverknüpften Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise haben Sie keine ungenutzte juristische Stelle, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch die Ressourcen für Ihre serviceverknüpften Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

Note

Wenn der MediaTailor Dienst die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu bereinigen, schlägt das Löschen möglicherweise fehl. Wenn dies passiert, warten Sie einige Minuten und versuchen Sie es erneut.

Um MediaTailor Ressourcen zu bereinigen, die von AWSService RoleForMediaTailor

- Bevor Sie die dienstbezogene Rolle löschen können, die von MediaTailor für die Protokollkonfiguration erstellt wurde, müssen Sie zunächst alle Protokollkonfigurationen in Ihrem Konto deaktivieren. Um eine Protokollkonfiguration zu deaktivieren, setzen Sie den Wert für „Prozent aktiviert“ auf 0. Dadurch wird die gesamte Sitzungsprotokollierung der entsprechenden Wiedergabekonfiguration deaktiviert. Weitere Informationen finden Sie unter [Deaktivierung einer Protokollkonfiguration](#).

So löschen Sie die serviceverknüpfte Rolle mit IAM

Verwenden Sie die IAM-Konsole, die AWS Command Line Interface (AWS CLI) oder die AWS API, um die AWSService RoleForMediaTailor serviceverknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter [Löschen einer serviceverknüpften Rolle](#) im IAM-Leitfaden.

Fehlerbehebung bei AWS Elemental MediaTailor Identität und Zugriff

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

Themen

- [Ich bin nicht berechtigt, eine Aktion durchzuführen in MediaTailor](#)
- [Ich bin nicht berechtigt, iam auszuführen: PassRole](#)
- [Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine MediaTailor Ressourcen ermöglichen](#)
- [MediaTailor kann nicht auf meinen Amazon S3 S3-Bucket zugreifen](#)

Ich bin nicht berechtigt, eine Aktion durchzuführen in MediaTailor

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

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

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

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

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

Ich bin nicht berechtigt, iam auszuführen: PassRole

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht zum Durchführen der iam:PassRole-Aktion autorisiert sind, müssen Ihre Richtlinien aktualisiert werden, um eine Rolle an MediaTailor übergeben zu können.

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

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

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

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

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

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

Sie können eine Rolle erstellen, mit der Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation auf Ihre Ressourcen zugreifen können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Für Dienste, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (ACLs) unterstützen, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

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

MediaTailor kann nicht auf meinen Amazon S3 S3-Bucket zugreifen

Wenn beim Versuch, auf Inhalte aus Ihrem Amazon S3 S3-Bucket zuzugreifen, Fehler MediaTailor zurückgegeben werden, überprüfen Sie die folgenden häufigen Ursachen:

Probleme mit Amazon S3 S3-Berechtigungen

Symptome des Fehlers:

- 502 BadGatewayException mit Nachrichten im Zusammenhang mit Amazon S3
- 403 Forbidden Fehler in den MediaTailor Protokollen
- Inhalt kann nicht vom Amazon S3 S3-Ursprung geladen werden

Auflösung

1. Stellen Sie sicher, dass Ihre MediaTailor Servicerolle über die erforderlichen Amazon S3 S3-Berechtigungen verfügt:
2. Prüfen Sie, ob Ihre Amazon S3 S3-Bucket-Richtlinie MediaTailor den Zugriff erlaubt:

Probleme mit dem HTTPS-Zugriff

Symptome des Fehlers:

- SSL/TLS-Handshake-Fehler
- 502 Bad Gateway beim Zugriff auf HTTPS Amazon S3 URLs

Auflösung

1. Stellen Sie sicher, dass Ihre Amazon S3 S3-Bucket-Richtlinie HTTPS erfordert:
2. Stellen Sie sicher, dass die MediaTailor Konfiguration HTTPS URLs für Amazon S3 S3-Inhalte verwendet.

Kontoübergreifender Amazon S3 S3-Zugriff

Symptome des Fehlers:

- Access Denied Fehler, wenn sich der Amazon S3 S3-Bucket in einem anderen Konto befindet
- MediaTailor kann kontoübergreifende Rollen nicht annehmen

Auflösung

1. Aktualisieren Sie die Amazon S3 S3-Bucket-Richtlinie im Inhaltskonto auf die MediaTailor Servicerolle des Trusts
2. Fügen Sie der Servicerolle kontoübergreifende Berechtigungen hinzu MediaTailor
3. Stellen Sie sicher, dass für beide Konten die richtigen Vertrauensbeziehungen konfiguriert sind

Überprüfung der Einhaltung der Vorschriften für AWS Elemental MediaTailor

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

Sie können Prüfberichte von Drittanbietern unter herunterladen AWS Artifact. Weitere Informationen finden Sie unter [Berichte herunterladen unter](#) .

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. Weitere Informationen zu Ihrer Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services finden Sie in der [AWS Sicherheitsdokumentation](#).

Resilienz in AWS Elemental MediaTailor

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

Weitere Informationen zu Availability Zones AWS-Regionen und Availability Zones finden Sie unter [AWS Globale Infrastruktur](#).

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

Sicherheit der Infrastruktur in MediaTailor

Als verwalteter Dienst AWS Elemental MediaTailor ist er durch AWS globale Netzwerksicherheit geschützt. Informationen zu AWS Sicherheitsdiensten und zum AWS Schutz der Infrastruktur finden Sie unter [AWS Cloud-Sicherheit](#). Informationen zum Entwerfen Ihrer AWS Umgebung unter Verwendung der bewährten Methoden für die Infrastruktursicherheit finden Sie unter [Infrastructure Protection](#) in Security Pillar AWS Well-Architected Framework.

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

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

Sie können diese API-Operationen von jedem Netzwerkstandort aus aufrufen, MediaTailor unterstützt jedoch ressourcenbasierte Zugriffsrichtlinien, die Einschränkungen auf der Grundlage der Quell-IP-Adresse beinhalten können. Sie können auch MediaTailor Richtlinien verwenden, um den Zugriff von bestimmten Amazon Virtual Private Cloud (Amazon VPC) -Endpunkten oder bestimmten zu kontrollieren. VPCs Dadurch wird der Netzwerkzugriff auf eine bestimmte MediaTailor Ressource effektiv nur von der spezifischen VPC innerhalb des AWS Netzwerks isoliert.

Serviceübergreifende Confused-Deputy-Prävention

Das Problem des verwirrten Stellvertreters ist ein Sicherheitsproblem, bei dem eine Entität, die keine Berechtigung zur Durchführung einer Aktion hat, eine privilegiertere Entität zur Durchführung der Aktion zwingen kann. In AWS kann ein dienstübergreifendes Identitätswechsels zu einem Problem mit dem verwirrten Stellvertreter führen. Ein dienstübergreifender Identitätswechsel kann auftreten, wenn ein Dienst (der Anruf-Dienst) einen anderen Dienst anruft (den aufgerufenen Dienst). Der aufrufende Service kann manipuliert werden, um seine Berechtigungen zu verwenden, um Aktionen auf die Ressourcen eines anderen Kunden auszuführen, für die er sonst keine Zugriffsberechtigung haben sollte. Um dies zu verhindern, bietet AWS Tools, mit denen Sie Ihre Daten für alle Services mit Serviceprinzipalen schützen können, die Zugriff auf Ressourcen in Ihrem Konto erhalten haben.

Wir empfehlen die Verwendung der SourceAccount globalen Bedingungsschlüssel [aws:SourceArn](#) und [aws:](#) in Ressourcenrichtlinien, um die Berechtigungen einzuschränken, die der

Ressource AWS Elemental MediaTailor einen anderen Dienst gewähren. Wenn Sie beide globalen Bedingungskontextschlüssel verwenden, müssen der `aws:SourceAccount`-Wert und das Konto im `aws:SourceArn`-Wert dieselbe Konto-ID verwenden, wenn sie in derselben Richtlinienanweisung verwendet werden.

Der Wert von `aws:SourceArn` muss die Wiedergabekonfiguration sein, mit der CloudWatch Protokolle für Ihre Region und Ihr Konto MediaTailor veröffentlicht werden. Dies gilt jedoch nur, wenn Sie die [MediaTailorLogger](#)-Rolle verwenden, mit der CloudWatch Amazon-Protokolle in Ihrem Konto MediaTailor veröffentlicht werden können. Dies gilt nicht, wenn Sie eine [dienstbezogene Rolle](#) verwenden, um die CloudWatch Protokolle MediaTailor veröffentlichen zu lassen.

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

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

Einloggen und Überwachen MediaTailor

Dieser Abschnitt bietet eine Übersicht über die Optionen zur Protokollierung und Überwachung in AWS Elemental MediaTailor zu Sicherheitszwecken. Weitere Informationen zur Protokollierung und Überwachung MediaTailor finden Sie unter [Ressourcen überwachen und kennzeichnen AWS Elemental MediaTailor](#).

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

CloudWatch Amazon-Alarme

Mithilfe von CloudWatch Alarmen beobachten Sie eine einzelne Metrik über einen von Ihnen festgelegten Zeitraum. Wenn die Metrik einen bestimmten Schwellenwert überschreitet, wird eine

Benachrichtigung an ein Amazon SNS SNS-Thema oder eine AWS Auto Scaling Scaling-Richtlinie gesendet. CloudWatch Alarmer lösen keine Aktionen aus, da sie sich in einem bestimmten Status befinden. Der Status muss sich stattdessen geändert haben und für eine festgelegte Anzahl an Zeiträumen aufrechterhalten worden sein. Weitere Informationen finden Sie unter [the section called “Überwachung mit CloudWatch Metriken”](#).

AWS CloudTrail Logs

CloudTrail bietet eine Aufzeichnung der Aktionen, die von einem Benutzer, einer Rolle oder einem AWS Dienst in ausgeführt wurden AWS Elemental MediaTailor. Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, an die die Anfrage gestellt wurde MediaTailor, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Informationen ermitteln. Weitere Informationen finden Sie unter [AWS Elemental MediaTailor API-Aufrufe aufzeichnen](#).

AWS Trusted Advisor

Trusted Advisor stützt sich auf bewährte Verfahren, die wir bei der Betreuung von Hunderttausenden von AWS Kunden gelernt haben. Trusted Advisor untersucht Ihre AWS Umgebung und gibt dann Empfehlungen, wenn Möglichkeiten bestehen, Geld zu sparen, die Systemverfügbarkeit und -leistung zu verbessern oder Sicherheitslücken zu schließen. Alle AWS -Kunden haben Zugriff auf fünf Trusted Advisor-Prüfungen. Kunden mit dem „Business“- oder „Enterprise“-Support-Plan können alle Trusted Advisor -Überprüfungen anzeigen.

Weitere Informationen finden Sie unter [AWS Trusted Advisor](#).

Ressourcen überwachen und kennzeichnen AWS Elemental MediaTailor

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung Ihrer anderen AWS Elemental MediaTailor Lösungen. AWS bietet die folgenden Überwachungstools, mit denen Sie beobachten MediaTailor, melden können, wenn etwas nicht stimmt, und gegebenenfalls automatische Maßnahmen ergreifen können:

- Amazon CloudWatch überwacht Ihre AWS Ressourcen und die Anwendungen, auf denen Sie laufen, AWS in Echtzeit. Sie können Kennzahlen erfassen und verfolgen, benutzerdefinierte Dashboards erstellen und Alarmer festlegen, die Sie benachrichtigen oder Maßnahmen ergreifen, wenn eine bestimmte Metrik einen von Ihnen festgelegten Schwellenwert erreicht. Sie können beispielsweise die CPU-Auslastung oder andere Kennzahlen Ihrer Amazon EC2 EC2-Instances CloudWatch verfolgen und bei Bedarf automatisch neue Instances starten. Weitere Informationen finden Sie im [CloudWatch Amazon-Benutzerhandbuch](#).
- Mit Amazon CloudWatch Logs können Sie Ihre Protokolldateien aus allen Interaktionen mit Ihrem Ad Decision Server (ADS) überwachen, speichern und darauf zugreifen. AWS Elemental MediaTailor sendet Protokolle für Anzeigenanfragen, Weiterleitungen und Antworten sowie für Berichts-anfragen und -antworten aus. Fehler von den ADS- und Ursprungsservern werden auch an Protokollgruppen in Amazon ausgegeben CloudWatch. MediaTailor bietet auch Informationen zu übersprungenen Anzeigen und den Gründen, warum sie übersprungen wurden. Weitere Informationen finden Sie unter [Fehlerbehebung beim Überspringen von Anzeigen](#). Sie können Ihre Protokolldaten auch in einem sehr robusten Speicher archivieren. Allgemeine Informationen finden Sie im [Amazon CloudWatch Logs-Benutzerhandbuch](#). Informationen zu den ADS-Protokollen und wie Sie auf sie für Analysen über Amazon CloudWatch Logs Insights zugreifen können, finden Sie unter [AWS Elemental MediaTailor ADS-Protokollanalyse in Amazon CloudWatch Logs Insights](#).

Themen

- [AWS Elemental MediaTailor Protokolle anzeigen](#)
- [Überwachung AWS Elemental MediaTailor mit CloudWatch Amazon-Metriken](#)
- [AWS Elemental MediaTailor API-Aufrufe aufzeichnen](#)
- [Empfangen von Benachrichtigungen zur AWS Elemental MediaTailor Kanalmontage](#)
- [Ressourcen taggen AWS Elemental MediaTailor](#)
- [Überwachung von AWS Mediendiensten mit Workflow-Monitor](#)

AWS Elemental MediaTailor Protokolle anzeigen

MediaTailor sendet Protokolle aus, die eine Vielzahl von Meilensteinen und Aktivitäten in Kanälen und Wiedergabekonfigurationen beschreiben. Sie können diese Protokolle verwenden, um sich einen Überblick über Ihren Arbeitsablauf zu verschaffen und Probleme mit dem Service zu beheben. In den folgenden Themen werden Protokolle und Protokollierungsoptionen beschrieben.

Themen

- [AWS Elemental MediaTailor Beschreibung und Ereignistypen der ADS-Protokolle](#)
- [AWS Elemental MediaTailor Beschreibung und Ereignistypen der Manifestprotokolle](#)
- [AWS Elemental MediaTailor Beschreibung und Ereignistypen der Transcodierungsprotokolle](#)
- [Versendete Protokolle werden zum Senden von Protokollen verwendet AWS Elemental MediaTailor](#)
- [AWS Elemental MediaTailor Logs direkt in Amazon CloudWatch Logs schreiben](#)
- [Steuerung des Volumens der Logs AWS Elemental MediaTailor](#)
- [Filterung von AWS Elemental MediaTailor Protokollen und Ereignissen](#)
- [AWS Elemental MediaTailor Debug-Protokolle werden generiert](#)

AWS Elemental MediaTailor Beschreibung und Ereignistypen der ADS-Protokolle

In den folgenden Abschnitten werden die Protokolle beschrieben, MediaTailor die zur Beschreibung von Ereignissen mit dem AD Decision Server (ADS) ausgegeben werden. Dies sind AdDecisionServerInteractions Protokolle.

Themen

- [AdDecisionServerInteractions-Ereignisse](#)
- [Beschreibung des ADS-Protokolls](#)
- [JSON-Schema für das ADS-Protokoll](#)

AdDecisionServerInteractions-Ereignisse

Die folgenden Ereignisse werden bei MediaTailor Interaktionen mit dem Ad Decision Server (ADS) ausgelöst.

Protokoll	Description
AD_MARKER_FOUND	MediaTailor Ich habe eine Werbemarkierung im Manifest gefunden.
BEACON_FIRED	Ein Tracking-Beacon wurde ausgelöst, um Werbeereignisse zu melden. MediaTailor feuert das Beacon im serverseitigen Berichtsmodus (Standard) ab. Im clientseitigen Berichtsmodus feuert das Wiedergabegerät den Beacon ab.
EMPTY_VAST_RESPONSE	Das ADS gab eine leere VAST-Antwort zurück, die keine Werbung enthielt.
EMPTY_VMAP_RESPONSE	Das ADS hat eine leere VMAP-Antwort zurückgegeben.
ERROR_ADS_INVALID_RESPONSE	Das ADS hat einen Statuscode zurückgegeben, der nicht 200 ist.
ERROR_ADS_IO	MediaTailor Beim Versuch, mit dem ADS zu kommunizieren, ist ein Fehler aufgetreten.
ERROR_ADS_RESPONSE_PARSE	MediaTailor ist beim Analysieren der ADS-Antwort auf einen Fehler gestoßen.
ERROR_ADS_RESPONSE_UNKNOWN_ROOT_ELEMENT	Die ADS-Antwort enthält ein ungültiges Stammelement.
ERROR_ADS_TIMEOUT	Bei der MediaTailor Anfrage an das ADS wurde das Zeitlimit überschritten.
ERROR_DISALLOWED_HOST	Der ADS-Host ist nicht zulässig.
ERROR_FIRING_BEACON_FAILED	MediaTailor konnte das Tracking-Beacon nicht abfeuern.

Protokoll	Description
ERROR_PERSONALIZATION_DISABLED	Das Einfügen von Anzeigen ist für diese Sitzung deaktiviert.
ERROR_UNKNOWN	MediaTailor ist bei der ADS-Anfrage auf einen unbekannten Fehler gestoßen.
ERROR_UNKNOWN_HOST	Der ADS-Host ist unbekannt.
ERROR_VAST_INVALID_MEDIA_FILE	Das Ad VAST-Objekt enthält ein ungültiges oder MediaFile fehlendes Element.
ERROR_VAST_INVALID_VAST_AD_TAG_URI	Die VAST-Antwort enthält ein ungültige sVASTAdTagURI .
ERROR_VAST_MISSING_CREATIVES	Das VAST Ad enthält null oder mehrere Creatives Elemente. Genau ein Creatives Element ist erforderlich.
ERROR_VAST_MISSING_IMPRESSION	Das VAST Ad enthält keine Impression Elemente. Es ist mindestens ein Impression Element erforderlich.
ERROR_VAST_MISSING_MEDIAFILES	Das VAST Ad enthält null oder mehrere MediaFiles Elemente. Genau ein MediaFiles Element ist erforderlich.
ERROR_VAST_MISSING_OVERLAYS	MediaTailor hat keine nichtlinearen Werbeanzeigen vom Ad-Server erhalten.
ERROR_VAST_MULTIPLE_LINEAR	Das VAST Ad enthält mehrere Linear Elemente.
ERROR_VAST_MULTIPLE_TRACKING_EVENTS	Das VAST Ad enthält mehrere TrackingEvents Elemente.
ERROR_VAST_REDIRECT_EMPTY_RESPONSE	Die VAST-Umleitungsanfrage gab eine leere Antwort zurück.

Protokoll	Description
ERROR_VAST_REDIRECT_FAILED	Bei der VAST-Umleitungsanfrage ist ein Fehler aufgetreten.
ERROR_VAST_REDIRECT_MULTIPLE_VAST	Bei der VAST-Weiterleitungsanfrage wurden mehrere Anzeigen zurückgegeben.
FILLED_AVAIL	MediaTailor Der Vorrat wurde erfolgreich ausgefüllt.
FILLED_OVERLAY_AVAIL	MediaTailor Das Overlay-Angebot wurde erfolgreich ausgefüllt.
INTERSTITIAL_VOD_FAILURE	Bei der ADS-Anfrage oder -Antwort ist beim Ausfüllen der Interstitial-Avails für die VOD-Wiedergabeliste ein Problem aufgetreten. Es werden keine Anzeigen eingefügt.
INTERSTITIAL_VOD_SUCCESS	MediaTailor Interstitielle Avails für die VOD-Playlist wurden erfolgreich ausgefüllt.
MAKING_ADS_REQUEST	MediaTailor fordert Werbung von der ADS an.
MODIFIED_TARGET_URL	MediaTailor hat die ausgehende Ziel-URL geändert.
NON_AD_MARKER_FOUND	MediaTailor Ich habe im Manifest eine Werbemarkierung gefunden, die nicht bearbeitet werden kann.
RAW_ADS_RESPONSE	MediaTailor hat eine unformatierte ADS-Antwort erhalten.
REDIRECTED_VAST_RESPONSE	MediaTailor erhielt eine VAST-Antwort, nachdem ich der VAST-Weiterleitung gefolgt war.

Protokoll	Description
VAST_REDIRECT	Die VAST-Anzeigenantwort enthält eine Weiterleitung.
VAST_RESPONSE	MediaTailor hat eine VAST-Antwort erhalten.
VOD_TIME_BASED_AVAIL_PLAN_SUCCESS	MediaTailor erfolgreich einen zeitbasierten Verfügbarkeitsplan für die VOD-Vorlage erstellt.
VOD_TIME_BASED_AVAIL_PLAN_VAST_RESPONSE_FOR_OFFSET	MediaTailor erstellt einen zeitbasierten Nutzungsplan für die VOD-Vorlage. MediaTailor hat eine VAST-Antwort für den Zeitversatz erhalten.
VOD_TIME_BASED_AVAIL_PLAN_WARNING_NO_ADVERTISEMENTS	Bei der ADS-Anfrage oder -Antwort ist beim Erstellen eines zeitbasierten Verfügbarkeitsplans für die VOD-Vorlage ein Problem aufgetreten. Es werden keine Anzeigen eingefügt.
WARNING_NO_ADVERTISEMENTS	MediaTailor ist beim Ausfüllen des Formulars auf ein Problem gestoßen. Es wurden keine Anzeigen eingefügt.
WARNING_URL_VARIABLE_SUBSTITUTION_FAILED	MediaTailor kann dynamische Variablen in der ADS-URL nicht ersetzen. Überprüfen Sie die URL-Konfiguration.
WARNING_VPAID_AD_DROPPED	Eine VPAID-Anzeige wurde aufgrund eines fehlenden Slate gelöscht, oder die Sitzung verwendet serverseitige Berichte.

Beschreibung des ADS-Protokolls

In diesem Abschnitt werden die Struktur und der Inhalt der ADS-Protokollbeschreibung beschrieben. Um sich selbst in einem JSON-Editor zu beschäftigen, verwenden Sie die Auflistung unter [the section called “JSON-Schema für das ADS-Protokoll”](#).

Jedes Ereignis im ADS-Protokoll enthält die Standardfelder, die von CloudWatch Logs generiert werden. Weitere Informationen finden Sie unter [Analysieren von Protokolldaten mit CloudWatch Logs Insights](#).

ADS protokolliert Eigenschaften

In diesem Abschnitt werden die Eigenschaften der ADS-Protokolle beschrieben.

Eigenschaft	Typ	Erforderlich	Beschreibung
<code>adsRequestUrl</code>	Zeichenfolge	false	Die vollständige URL der ADS-Anfrage von MediaTailor.
<code>avail</code>	Objekt des Typs avail	false	Informationen zu einem Angebot, das voller MediaTailor Werbung ist. Derzeit ist dies für den FILLED_AVAIL Ereignistyp der Plan, der erstellt wird, MediaTailor wenn er zum ersten Mal auf den Erfolg stößt. Wie der <code>avail</code> letztendlich gefüllt wird, kann sich von diesem Plan unterscheiden, je nachdem, wie die Inhalte abgespielt werden.
<code>awsAccountId</code>	Zeichenfolge	true	Die AWS-Konto-ID für die MediaTailor Konfiguration, die für die Sitzung verwendet wurde.

Eigenschaft	Typ	Erforderlich	Beschreibung
<code>customerId</code>	Zeichenfolge	true	Die gehashte Version der AWS-Konto-ID, die Sie verwenden können, um mehrere Protokolleinträge zu korrelieren.
<code>eventDescription</code>	Zeichenfolge	true	Eine kurze Beschreibung des Ereignisses, das diese Protokollnachricht ausgelöst hat, bereitgestellt vom MediaTailor Service. Standardmäßig ist dieses Feld leer. Beispiel: Got VAST response.
<code>eventTimestamp</code>	Zeichenfolge	true	Das Datum und die Uhrzeit des Ereignisses.
<code>eventType</code>	Zeichenfolge	true	Der Code für das Ereignis, das diese Protokollnachricht ausgelöst hat. Beispiel: VAST_RESPONSE .

Eigenschaft	Typ	Erforderlich	Beschreibung
originId	Zeichenfolge	true	Der Konfigurationsname aus der MediaTailor Konfiguration. Dies unterscheidet sich von der Quelle der Videoinhalte, die ebenfalls Teil der Konfiguration ist.
prefetchScheduleName	Zeichenfolge	false	Der Name des Prefetch-Zeitplans, der mit diesem Werbeereignis verknüpft ist.
requestHeaders	Array vom Typ requestheaders	false	Die Header, die in der MediaTailor ADS-Anfrage enthalten waren. In der Regel enthalten die Protokolle diese, wenn eine Anforderung an den ADS fehlschlägt, um die Fehlerbehebung zu unterstützen.
requestId	Zeichenfolge	true	Die MediaTailor Anforderungs-ID, mit der Sie mehrere Protokolleinträge für dieselbe Anfrage korrelieren können.

Eigenschaft	Typ	Erforderlich	Beschreibung
<code>sessionId</code>	Zeichenfolge	true	Die eindeutige numerische Kennung, die MediaTailor der Spiellersitzung zugewiesen wurde. Alle Anforderungen, die ein Spieler für eine Sitzung stellt, haben dieselbe Sitzungs-ID. Beispiel: e039fd39-09f0-46b2-aca9-9871cc116cde .
<code>sessionType</code>	Zeichenfolge (zulässige Werte: [DASH, HLS])	true	Der Stream-Typ des Spielers.
<code>vastAd</code>	Objekt des Typs vastAd	false	Informationen zu einer einzelnen Anzeige, die aus der VAST-Antwort analysiert wurde.
<code>vastResponse</code>	Objekt des Typs vastResponse	false	Informationen über die VAST-Antwort, die vom ADS MediaTailor empfangen wurde.

Eigenschaft	Typ	Erforderlich	Beschreibung
<code>vodCreativeOffsets</code>	Objekt des Typs vodCreativeOffsets	false	Eine Karte, die die Zeitversätze im Manifest angibt, bei denen Avails auf der Grundlage der VMAP-Antwort eingefügt MediaTailor werden.
<code>vodVastResponseTimeOffset</code>	number	false	Der VMAP-spezifische Zeitversatz zum Einfügen der VOD-Anzeige.

adContent

In diesem Abschnitt werden die Eigenschaften der ADS-Protokolle adContent beschrieben.

Eigenschaften der ADS-Protokolle adContent

Eigenschaft	Typ	Erforderlich	Description
<code>adPlaylistUris</code>	Objekt des Typs adPlaylistUris	false	Die Zuordnung vom Ursprungsmanifest für eine Variante zum Anzeigenmanifest für die Variante. Für DASH enthält dies einen einzelnen Eintrag, da alle Varianten in einem einzigen DASH-Manifest dargestellt werden.

adPlaylistUri

In diesem Abschnitt werden die Eigenschaften der ADS-Protokolle beschrieben adPlaylistUri.

adPlaylistUri Eigenschaften von ADS-Protokollen

Eigenschaft	Typ	Erforderlich	Beschreibung
<any string>	Zeichenfolge	false	Die URL des Anzeigenmanifests für die spezifische Variante.

avail

In diesem Abschnitt werden die Eigenschaften der ADS-Protokolle avail beschrieben.

Eigenschaften der ADS-Protokolle avail

Eigenschaft	Typ	Erforderlich	Beschreibung
availId	Zeichenfolge	true	Die eindeutige Kennung für diese avail-Protokoll. Bei HLS ist dies die Mediensequenznummer, bei der der avail beginnt. Für DASH ist dies die Punkt-ID.
creativeAds	Array vom Typ creativeAd	true	Die Anzeigen, die in den Avail MediaTailor eingefügt wurden.
fillRate	number	true	Die Rate, mit der die Anzeigen die avail-Dauer erfüllen, von

Eigenschaft	Typ	Erforderlich	Beschreibung
			0,0 (für 0 %) bis 1,0 (für 100 %).
filledDuration	number	true	Die Summe der Dauer aller Anzeigen, die in avail eingefügt wurden.
numAds	number	true	Die Anzahl der Anzeigen, die in den Avail MediaTailor eingefügt wurden.
originAvailDuration	number	true	Die Dauer des avails, wie im Inhalts-Stream vom Ursprung angegeben (CUE_OUT oder SCTE).
skippedAds	Array vom Typ skippedAd	false	Die Anzeigen, die aus Gründen wie TRANSCODE_IN_PROGRESS und MediaTailor TRANSCODE_ERROR nicht eingefügt wurden. Eine Liste der Gründe, warum Werbung übersprungen wurde, finden Sie unter Fehlerbehebung beim Überspringen von Anzeigen.

Eigenschaft	Typ	Erforderlich	Beschreibung
slateAd	Objekt des Typs slateAd	true	Informationen zur Slate-Anzeige, mit MediaTailor der alle nicht ausgefüllten Segmente aufgefüllt werden.

creativeAd

In diesem Abschnitt werden die Eigenschaften der ADS-Protokolle creativeAd beschrieben.

Eigenschaften der ADS-Protokolle creativeAd

Eigenschaft	Typ	Erforderlich	Description
adContent	Objekt des Typs adContent	true	Informationen über den Inhalt der eingefügten Anzeige.
creativeUniqueId	Zeichenfolge	true	Die eindeutige Kennung für die Anzeige, die als Schlüssel für die Transcodierung verwendet wird. Dies ist das ID-Feld für das Werbemittel in der VAST-Antwort, sofern verfügbar. Andernfalls handelt es sich um die Mezzanine-URL der Anzeige.
trackingEvents	Objekt des Typs trackingEvents	true	Das Tracking-Beacon URLs für die verschiedenen

Eigenschaft	Typ	Erforderlich	Description
			Tracking-Ereignisse für die Anzeige. Die Schlüssel sind die Namen der Ereignisse, und die Werte sind eine Liste von Beacons URLs.
transcodeAdDuration	number	true	Die Dauer der Anzeige, berechnet aus der transcodierten Komponente.
uri	Zeichenfolge	true	Die URL der Mezzanine-Version der Anzeige, bei der es sich um die Eingabe für den Transcoder handelt.
vastDuration	number	true	Die Dauer der Anzeige, wie aus der VAST-Antwort analysiert.

requestheaders

In diesem Abschnitt werden die Eigenschaften der ADS-Protokolle requestheaders beschrieben.

Eigenschaften der ADS-Protokolle requestheaders

Eigenschaft	Typ	Erforderlich	Beschreibung
name	Zeichenfolge	true	Der Name des hinzuzufügenden Headers.

Eigenschaft	Typ	Erforderlich	Beschreibung
value	Zeichenfolge	true	Der Wert des Headers

skippedAd

In diesem Abschnitt werden die Eigenschaften der ADS-Protokolle skippedAd beschrieben.

Eigenschaften der ADS-Protokolle skippedAd

Eigenschaft	Typ	Erforderlich	Beschreibung
adMezzanineUrl	Zeichenfolge	true	Die Mezzanine-URL der übersprungenen Anzeige.
creativeUniqueId	Zeichenfolge	true	Die eindeutige Kennung für die Anzeige, die als Schlüssel für die Transcodierung verwendet wird. Dies ist das ID-Feld für das Werbemittel in der VAST-Antwort, sofern verfügbar. Andernfalls handelt es sich um die Mezzanine-URL der Anzeige.
skippedReason	Zeichenfolge	true	Der Code, der angibt, warum die Anzeige nicht eingefügt wurde. Beispiel: TRANSCODE_IN_PROGRESS . Eine Liste der Gründe, warum

Eigenschaft	Typ	Erforderlich	Beschreibung
			Werbung übersprungen wurde, finden Sie unter. Fehlerbehebung beim Überspringen von Anzeigen
transcodeAdDuration	number	false	Die Dauer der Anzeige, berechnet aus der transcodierten Komponente.
vastDuration	number	true	Die Dauer der Anzeige, wie aus der VAST-Antwort analysiert.

slateAd

In diesem Abschnitt werden die Eigenschaften der ADS-Protokolle slateAd beschrieben.

Eigenschaften der ADS-Protokolle slateAd

Eigenschaft	Typ	Erforderlich	Description
adContent	Objekt des Typs adContent	true	Informationen über den Inhalt der eingefügten Anzeige.
creativeUniqueId	Zeichenfolge	true	Die eindeutige Kennung für die Anzeige, die als Schlüssel für die Transcodierung verwendet wird. Dies ist das ID-Feld für das Werbemittel in der VAST-Antwort, sofern

Eigenschaft	Typ	Erforderlich	Description
			verfügbar. Andernfalls handelt es sich um die Mezzanine-URL der Anzeige.
<code>transcodeAdDuration</code>	number	true	Die Dauer der Anzeige, berechnet aus der transcodierten Komponente.
<code>uri</code>	Zeichenfolge	true	Die URL der Mezzanine-Version der Anzeige, bei der es sich um die Eingabe für den Transcoder handelt.

trackingEvents

In diesem Abschnitt werden die Eigenschaften der ADS-Protokolle trackingEvents beschrieben.

Eigenschaften der ADS-Protokolle trackingEvents

Eigenschaft	Typ	Erforderlich	Description
<code><any string></code>	Array vom Typ Zeichenfolge	false	Die Liste der Beacons URLs für das angegebene Tracking-Ereignis (Impression, abgeschlossen usw.)

vastAd

In diesem Abschnitt werden die Eigenschaften der ADS-Protokolle vastAd beschrieben.

Eigenschaften der ADS-Protokolle vastAd

Eigenschaft	Typ	Erforderlich	Beschreibung
adSystem	Zeichenfolge	true	Der Wert des AdSystem-Tags in der VAST-Antwort.
adTitle	Zeichenfolge	true	Die Mediendateien, die für die Anzeige in der VAST-Antwort verfügbar sind.
creativeAdId	Zeichenfolge	true	Der Wert des adId-Attributs des Creative-Tags in der VAST-Antwort.
creativeId	Zeichenfolge	true	Der Wert des id-Attributs des Creative-Tags in der VAST-Antwort.
duration	number	true	Die ungefähre Dauer der Anzeige, basierend auf dem duration-Tag im linear-Element der VAST-Antwort.
trackingEvents	Objekt des Typs trackingEvents	true	Das Tracking-Beacon URLs für die verschiedenen Tracking-Ereignisse für die Anzeige. Die Schlüssel sind die Namen der Ereignisse, und die Werte

Eigenschaft	Typ	Erforderlich	Beschreibung
			sind eine Liste von Beacons URLs.
vastAdId	Zeichenfolge	true	Der Wert des id-Attributs des Ad-Tags in der VAST-Antwort
vastAdTagUri	Zeichenfolge	false	Die VMAP-spezifische Umleitungs-URI für eine Anzeige.
vastMediaFiles	Array vom Typ vastMediaFile	true	Die Liste der verfügbaren Mediendateien für die Anzeige in der VAST-Antwort.

vastMediaFile

In diesem Abschnitt werden die Eigenschaften der ADS-Protokolle beschrieben vastMediaFile.

vastMediaFile Eigenschaften von ADS-Protokollen

Eigenschaft	Typ	Erforderlich	Beschreibung
apiFramework	Zeichenfolge	true	Das API-Framework, das für die Verwaltung der Mediendatei erforderlich ist. Beispiel: VPAID.
bitrate	number	true	Die Bitrate der Mediendatei.
delivery	Zeichenfolge	true	Das Protokoll, das für die Mediendatei verwendet wird, ist

Eigenschaft	Typ	Erforderlich	Beschreibung
			auf progressiv oder Streaming festgelegt.
height	number	true	Die Pixelhöhe der Mediendatei.
id	Zeichenfolge	true	Der Wert des id-Attributs des MediaFile -Tags.
type	Zeichenfolge	true	Der MIME-Typ der Mediendatei aus dem type-Attribut des MediaFile -Tags.
uri	Zeichenfolge	true	Die URL der Mezzanine-Version der Anzeige, bei der es sich um die Eingabe für den Transcoder handelt.
width	number	true	Die Pixelbreite der Mediendatei.

vastResponse

In diesem Abschnitt werden die Eigenschaften der ADS-Protokolle vastResponse beschrieben.

Eigenschaften der ADS-Protokolle vastResponse

Eigenschaft	Typ	Erforderlich	Description
errors	Array vom Typ Zeichenfolge	true	Der Fehler URLs wurde anhand der Error Tags in

Eigenschaft	Typ	Erforderlich	Description
			der VAST-Antwort analysiert.
<code>vastAds</code>	Array vom Typ vastAd	true	Die Anzeigen, die aus der VAST-Antwort analysiert wurden.
<code>version</code>	Zeichenfolge	true	Die VAST-Spezifikationsversion, die aus dem <code>version</code> -Attribut des VAST-Tags in der Antwort analysiert wird.

vodCreativeOffsets

In diesem Abschnitt werden die Eigenschaften der ADS-Protokolle beschrieben vodCreativeOffsets.

vodCreativeOffsets Eigenschaften von ADS-Protokollen

Eigenschaft	Typ	Erforderlich	Description
<code><any string></code>	Array vom Typ vodCreativeOffset	false	Eine Zuordnung von einem Zeitversatz im Manifest zu einer Liste von Anzeigen, die zu diesem Zeitpunkt eingefügt werden sollen.

vodCreativeOffset

In diesem Abschnitt werden die Eigenschaften der ADS-Protokolle beschrieben vodCreativeOffset.

vodCreativeOffset Eigenschaften von ADS-Protokollen

Eigenschaft	Typ	Erforderlich	Description
adContent	Objekt des Typs adContent	true	Informationen über den Inhalt der eingefügten Anzeige.
creativeUniqueId	Zeichenfolge	true	Die eindeutige Kennung für die Anzeige, die als Schlüssel für die Transcodierung verwendet wird. Dies ist das ID-Feld für das Werbemittel in der VAST-Antwort, sofern verfügbar. Andernfalls handelt es sich um die Mezzanine-URL der Anzeige.
trackingEvents	Objekt des Typs trackingEvents	true	Das Tracking-Beacon URLs für die verschiedenen Tracking-Ereignisse für die Anzeige. Die Schlüssel sind die Namen der Ereignisse, und die Werte sind eine Liste von Beacons URLs.
transcodedAdDuration	number	true	Die Dauer der Anzeige, berechnet aus der transcodierten Komponente.

Eigenschaft	Typ	Erforderlich	Description
uri	Zeichenfolge	true	Die URL der Mezzanine-Version der Anzeige, bei der es sich um die Eingabe für den Transcoder handelt.
vastDuration	number	true	Die Dauer der Anzeige, wie aus der VAST-Antwort analysiert.

JSON-Schema für das ADS-Protokoll

Im Folgenden ist das JSON-Schema für das AWS Elemental MediaTailor ADS-Protokoll aufgeführt.

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$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."
}

```

```

    }
  },

```

"__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" }
],

"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"
    ],
    "_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": {
    "$id": "#/properties/adContent",
    "type": ["object", "null"],
    "title": "adContent",
    "description": "Information about the content of the inserted ad.",
    "additionalProperties": false,
    "properties": {
      "adPlaylistUris": {
        "$id": "#/properties/adContent/adPlaylistUris",
        "type": "object",
        "title": "adPlaylistUris",
        "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/adPlaylistUris/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"
  ]
}
}
}
}
}
}
}
}

```

AWS Elemental MediaTailor Beschreibung und Ereignistypen der Manifestprotokolle

In den folgenden Abschnitten werden die Protokolle beschrieben, die ausgegeben MediaTailor werden, um Ereignisse mit dem Ursprungsserver zu beschreiben, wenn ein Manifest angefordert und empfangen wird. Dies sind ManifestService Protokolle.

Themen

- [ManifestService-Ereignisse](#)
- [Das Manifest protokolliert die Eigenschaften](#)

ManifestService-Ereignisse

Die folgenden Ereignisse werden bei MediaTailor Interaktionen mit dem Ursprung ausgelöst.

Protokoll	Description
CONFIG_SECURITY_ERROR	Die MediaTailor Konfiguration hat ein Sicherheitsproblem.
CONFIG_SYNTAX_ERROR	Der Ursprung und der Asset-Pfad führen zu einer falsch formatierten URL.
CONNECTION_ERROR	Die MediaTailor Verbindung zum Ursprung wurde abgelehnt oder ist fehlgeschlagen.
GENERATED_MANIFEST	MediaTailor hat ein Manifest generiert. Sie müssen den Debug-Modus aktiviert haben, um diese Protokolle empfangen zu können. Hinweise zum Debug-Protokollmodus, einschließlich seiner Aktivierung, finden Sie unter Debug-Protokolle generieren
HOST_DISALLOWED	MediaTailor lässt keine HTTP-Anfragen an diesen Host zu.
INCOMPATIBLE_HLS_VERSION	Das Manifest verwendet eine inkompatible HLS-Version. MediaTailor erfordert Version 3 oder höher.
INVALID_SINGLE_PERIOD_DASH_MANIFEST	Das DASH-Manifest mit einer Periode ist ungültig. MediaTailor durchläuft ein DASH-Manifest mit einer einzigen Periode.
IO_ERROR	MediaTailor ist bei der Kommunikation mit dem Ursprung auf einen IO-Fehler gestoßen.
LAST_PERIOD_MISSING_AUDIO	Im letzten Abschnitt des DASH-Manifests fehlt der gesamte Ton <code>AdaptationSets</code> aufgrund einer falschen Ausrichtung der ursprünglichen

Protokoll	Description
	Audio- oder Videodaten. Um Probleme bei der Wiedergabe zu vermeiden, sollten Sie die Veröffentlichung des letzten Zeitraums bis mindestens zur nächsten Anfrage verschieben.
LAST_PERIOD_MISSING_AUDIO_WARNING	Im letzten Abschnitt des DASH-Manifests fehlt <code>AdaptationSets</code> aufgrund einer falschen Ausrichtung der ursprünglichen Audio- oder Videodaten das gesamte Audio. Es wird entschieden, den letzten Zeitraum zu veröffentlichen (nicht zu verzögern). Fehlendes Audio kann zu Problemen bei der Wiedergabe führen.
MANIFEST_ERROR	Die MediaTailor Manifestanforderung ist fehlgeschlagen.
NO_MASTER_OR_MEDIA_PLAYLIST	Die ursprüngliche Antwort enthält keine primäre Playlist oder Medien-Playlist.
NO_MASTER_PLAYLIST	Die ursprüngliche Antwort enthält nicht die erwartete primäre Playlist.
NO_MEDIA_PLAYLIST	Die ursprüngliche Antwort enthält nicht die erwartete Medien-Playlist.
ORIGIN_MANIFEST	MediaTailor hat ein Ursprungsmanifest abgerufen. Sie müssen den Debug-Modus aktiviert haben, um diese Protokolle empfangen zu können. Hinweise zum Debug-Protokollmodus, einschließlich seiner Aktivierung, finden Sie unter Debug-Protokolle generieren .
PARSING_ERROR	Der Ursprung kann die Manifestanforderung nicht analysieren.

Protokoll	Description
SCTE35_PARSING_ERROR	MediaTailor kann das Signal Binary Element im Manifest nicht analysieren.
SESSION_INITIALIZED	Eine Sitzung wurde initialisiert. Sie müssen den Debug-Modus aktiviert haben, um diese Protokolle empfangen zu können. Hinweise zum Debug-Protokollmodus, einschließlich seiner Aktivierung, finden Sie unter. Debug-Protokolle generieren
TIMEOUT_ERROR	Bei der MediaTailor Manifestanforderung wurde das Zeitlimit überschritten.
TRACKING_RESPONSE	MediaTailor hat eine Tracking-Antwort zugestellt. Sie müssen den Debug-Modus aktiviert haben, um diese Protokolle empfangen zu können. Hinweise zum Debug-Protokollmodus, einschließlich seiner Aktivierung, finden Sie unter. Debug-Protokolle generieren
UNKNOWN_ERROR	MediaTailor ist auf einen unbekannten Fehler gestoßen.
UNKNOWN_HOST	Der Host ist unbekannt.
UNSUPPORTED_SINGLE_PERIOD_DASH_MANIFEST	Das DASH-Manifest mit einer Periode wird nicht unterstützt. MediaTailor durchläuft ein DASH-Manifest mit einer einzigen Periode.

Das Manifest protokolliert die Eigenschaften

In diesem Abschnitt werden die Eigenschaften der Manifestprotokolle beschrieben.

Eigenschaft	Typ	Erforderlich
awsAccountId	Zeichenfolge	true
eventTimestamp	Zeichenfolge	true
originId	Zeichenfolge	true
customerId	Zeichenfolge	false
eventType	Zeichenfolge	false
sessionId	Zeichenfolge	false
originRequestUrl	Zeichenfolge	false
mediaTailorPath	Zeichenfolge	false
requestId	Zeichenfolge	false
responseBody	Zeichenfolge	false
sessionType	Zeichenfolge (zulässige Werte: [DASH, HLS])	false
requestNextToken	Zeichenfolge	false
eventDescription	Zeichenfolge	false
assetPath	Zeichenfolge	false
originFullUrl	Zeichenfolge	false
originPrefixUrl	Zeichenfolge	false
additionalInfo	Zeichenfolge	false
cause	Zeichenfolge	false
response	Zeichenfolge	false

Eigenschaft	Typ	Erforderlich
httpCode	Zeichenfolge	false
errorMessage	Zeichenfolge	false
adAdsResponse	Zeichenfolge	false
adAdsRawResponse	Zeichenfolge	false
adAdsRequest	Zeichenfolge	false
adNumNewAvails	Zeichenfolge	false
generatedMediaPlay list	Zeichenfolge	false
requestStartTime	Zeichenfolge	false
requestEndTime	Zeichenfolge	false
requestStartTimeEp ochMillis	Zeichenfolge	false
requestEndTimeEpoc hMillis	Zeichenfolge	false

AWS Elemental MediaTailor Beschreibung und Ereignistypen der Transcodierungsprotokolle

In den folgenden Abschnitten werden die Logs beschrieben, die ausgegeben werden MediaTailor , um Ereignisse mit dem Transcode-Dienst zu beschreiben, wenn Werbematerial auf das Anzeigen-Stitching vorbereitet wird. Dabei handelt es sich um Protokolle. TranscodeService

Themen

- [TranscodeService-Ereignisse](#)
- [Eigenschaften von Transcode-Protokollen](#)

TranscodeService-Ereignisse

Die folgenden Ereignisse werden bei MediaTailor Interaktionen während der Transcodierung von Anzeigen ausgelöst.

Protokoll	Description
IMPORT_ERROR	MediaTailor ist während eines Importauftrags (für vorkonditionierte Werbung) auf einen internen Fehler gestoßen. Es wurde ein leerer Satz von Anzeigen verwendet.
INITIALIZED	MediaTailor hat entweder einen Transcodierungsauftrag oder einen Importauftrag (für vorkonditionierte Anzeigen) initialisiert.
INTERNAL_ERROR	MediaTailor ist auf einen internen Fehler gestoßen. Es wurde ein leerer Satz von Anzeigen verwendet.
MISSING_VARIANTS	MediaTailor konnte die Anzeige aufgrund fehlender Varianten nicht transkodieren. Es wurde ein leerer Satz von Anzeigen verwendet.
PROFILE_NOT_FOUND	MediaTailor konnte die Anzeige nicht transkodieren, da ein zu transkodierendes Profil fehlt. Es wurde ein leerer Satz von Anzeigen verwendet.
TRANSCODE_COMPLETED	Die Videotranskodierung ist abgeschlossen. Die Anzeige kann zum Einfügen von Anzeigen verwendet werden.
TRANSCODE_ERROR	MediaTailor ist während eines Transcodierungsauftrags auf einen internen Fehler gestoßen. Es wurde ein leerer Satz von Anzeigen verwendet.

Protokoll	Description
TRANSCODE_IN_PROGRESS	Die Videotranskodierung ist im Gange. Das transkodierte Video ist nicht bereit. Es wird ein leerer Satz von Anzeigen verwendet.

Eigenschaften von Transcode-Protokollen

In diesem Abschnitt werden die Eigenschaften der Transcodierungsprotokolle beschrieben.

Eigenschaft	Typ	Erforderlich	Beschreibung
awsAccountId	Zeichenfolge	true	Die AWS Konto-ID für die MediaTailor Konfiguration, die für die Sitzung verwendet wurde.
eventTimestamp	Zeichenfolge	true	Das Datum und die Uhrzeit des Ereignisses.
originId	Zeichenfolge	true	Der Konfigurationsname aus der MediaTailor Konfiguration. Dies unterscheidet sich von der Quelle der Videoinhalte, die ebenfalls Teil der Konfiguration ist.
eventType	Zeichenfolge	false	Der Code für das Ereignis, das diese Protokollnachricht ausgelöst hat. Beispiel: TRANSCODE_ERROR .

Eigenschaft	Typ	Erforderlich	Beschreibung
<code>eventDescription</code>	Zeichenfolge	false	Eine kurze Beschreibung des Ereignisses, das diese Protokollnachricht ausgelöst hat. Sie wird vom MediaTailor Dienst bereitgestellt. Standardmäßig ist dieses Feld leer.
<code>sessionId</code>	Zeichenfolge	false	Die eindeutige numerische Kennung, die MediaTailor der Spielsitzung zugewiesen wurde. Alle Anforderungen, die ein Spieler für eine Sitzung stellt, haben dieselbe Sitzungs-ID. Beispiel: e039fd39-09f0-46b2-aca9-9871cc116cde .
<code>creativeUniqueId</code>	Zeichenfolge	false	Die eindeutige Kennung für das Werbematerial, das transkodiert wird.
<code>profileName</code>	Zeichenfolge	false	
<code>adUri</code>	Zeichenfolge	false	Die URI für das Werbemittel.

Eigenschaft	Typ	Erforderlich	Beschreibung
transcode RequestId	Zeichenfolge	false	Die eindeutige Kennung für diese Transcodierungsanforderung.
cacheStatus	Zeichenfolge	false	Gibt an, ob die MediaTailor transkodierte Anzeige zwischengespeichert wurde.

Versendete Protokolle werden zum Senden von Protokollen verwendet

AWS Elemental MediaTailor

Sie können Vended-Logs verwenden, um flexibler zu MediaTailor sein und mehr Kontrolle darüber zu haben, wohin Logs aus Ihrer Wiedergabekonfiguration gesendet werden sollen.

MediaTailor Sendet bei verkauften Protokollen alle Protokollaktivitäten, die mit einer Konfiguration verknüpft sind, an Amazon CloudWatch Logs. CloudWatch Logs sendet dann den Prozentsatz der von Ihnen angegebenen Logs an das von Ihnen gewählte Ziel. Unterstützte Ziele sind eine Amazon CloudWatch Logs-Protokollgruppe, ein Amazon S3 S3-Bucket oder ein Amazon Data Firehose-Stream.

Da verkaufte Protokolle zu Mengenrabatten erhältlich sind, können Sie im Vergleich zum direkten Senden von Protokollen an Logs Kosteneinsparungen feststellen. CloudWatch Die Preise finden Sie unter Verkaufte Logs auf der Registerkarte Logs bei [Amazon CloudWatch Pricing](#).

Um Vended Logs zu verwenden, müssen Sie wie folgt vorgehen:

1. [Berechtigungen hinzufügen](#).
2. [Erstellen Sie Ziele für die Protokollzustellung](#).
3. [Konfigurieren Sie die Protokollzustellung in CloudWatch Logs](#).
4. [Aktivieren Sie Vended-Logs in MediaTailor](#).

Weitere Informationen zu Vended-Logs finden Sie unter [Aktivieren der Protokollierung von AWS Diensten](#) im CloudWatch Logs-Benutzerhandbuch. MediaTailor unterstützt Version 2 der verkauften Logs.

Schritt 1: Fügen Sie Berechtigungen für die MediaTailor Protokollzustellung hinzu

Die Person, die Versandprotokolle einrichtet, muss über die erforderlichen Berechtigungen verfügen, um das Zustellungsziel zu erstellen, die Protokollzustellung zu konfigurieren und die Anmeldung per Versand zu aktivieren. MediaTailor Verwenden Sie die folgenden Richtlinien, um sicherzustellen, dass Sie über die entsprechenden Berechtigungen zum Einrichten von Versandprotokollen verfügen.

Richtlinien für CloudWatch Protokolle und Lieferziele

Die folgenden Abschnitte im Amazon CloudWatch Logs-Benutzerhandbuch enthalten die Richtlinien, die es Ihnen ermöglichen, mit Protokollen in CloudWatch Logs und Ihren Lieferzielen zu arbeiten. Wenn Sie Protokolle an mehrere Standorte senden, können Sie die Richtlinienenerklärungen zu einer Richtlinie zusammenfassen, anstatt mehrere Richtlinien zu erstellen.

- [Protokolle, die an CloudWatch Logs gesendet wurden](#)
- [An Amazon S3 gesendete Protokolle](#)
- [An Firehose gesendete Logs](#)

Richtlinie für die Einrichtung über die Konsole

Wenn Sie den Versand von Verkaufsprotokollen über die Konsole statt über die API oder einrichten AWS CLI, müssen Sie in Ihrer Richtlinie über die folgenden zusätzlichen Berechtigungen verfügen.

Die Richtlinie für verkaufte Logins MediaTailor

In Ihrer Richtlinie müssen Sie über die folgenden Berechtigungen verfügen, um die Zustellung von ausgelieferten Protokollen zu erstellen MediaTailor, anzuzeigen oder zu ändern.

Informationen zum Hinzufügen von Berechtigungen und zum Arbeiten mit Richtlinien finden Sie unter [Identity and Access Management für AWS Elemental MediaTailor](#).

Schritt 2: Lieferziele für MediaTailor Protokolle erstellen

Erstellen Sie die Ressourcen, an die Ihre Protokolle gesendet werden sollen. Notieren Sie den ARN der Ressource, um ihn bei der Konfiguration der Protokollzustellung in einem späteren Schritt zu verwenden.

CloudWatch Protokolliert das Ziel der Protokollgruppenzustellung

Verwenden Sie eine der folgenden Optionen, um Hilfe bei der Erstellung einer Protokollgruppe zu erhalten.

- Informationen zur Konsole finden Sie unter [Erstellen einer Protokollgruppe in CloudWatch Logs](#) im Amazon CloudWatch Logs-Benutzerhandbuch.
- Informationen zur API finden Sie [CreateLogGroup](#) in der Amazon CloudWatch Logs API-Referenz.
- Informationen zu SDKs und CLI finden [Sie unter Verwendung CreateLogGroup mit einem AWS SDK oder AWS CLI](#) im Amazon CloudWatch Logs-Benutzerhandbuch.

Lieferziel für Amazon S3 S3-Buckets

Verwenden Sie eine der folgenden Optionen, um Hilfe bei der Erstellung eines Buckets zu erhalten.

- Informationen zur Konsole und zur CLI finden Sie unter [Create a Bucket](#) im Amazon Simple Storage Service-Benutzerhandbuch. SDKs
- Informationen zur API finden Sie [CreateBucket](#) in der Amazon Simple Storage Service API-Referenz.

Firehose Stream-Lieferziel

Hilfe beim Erstellen eines Streams finden Sie unter [Create a Firehose Stream from console](#) im Amazon Data Firehose Developer Guide.

Schritt 3: Aktivieren Sie Vended-Logs für die Wiedergabekonfiguration MediaTailor

Erstellen oder aktualisieren Sie die Wiedergabekonfiguration, mit der Protokolle an das Lieferziel gesendet werden, das Sie im vorherigen Schritt erstellt haben. Notieren Sie sich den Namen der Konfiguration, um ihn bei der Konfiguration der Protokollzustellung in einem späteren Schritt zu verwenden.

- Um die Übertragung von Protokollen über die Konsole zu aktivieren, verwenden [Konfiguration erstellen](#) oder [Konfigurationseinstellungen bearbeiten](#) bearbeiten Sie eine Konfiguration, um auf die Protokollierungseinstellungen zuzugreifen. Wählen Sie für Logging-Strategien die Option Vended Logs aus.
- Um Vended Logs über die API zu aktivieren, müssen Sie über eine bestehende Konfiguration verfügen. Wird verwendet `ConfigureLogsForPlaybackConfiguration`, um die Protokollierungsstrategie Vended Logs hinzuzufügen.

Wenn Sie die bisherige MediaTailor Protokollierungsstrategie verwenden, bei der Logs direkt an Logs gesendet werden, und zu ausverkauften CloudWatch Logs migrieren möchten, finden Sie weitere Informationen unter [Migration der Protokollierungsstrategie](#).

 Important

Wenn Sie die Protokollstrategie von Legacy CloudWatch auf Vended Logs ändern, MediaTailor wird diese Änderung vorgenommen, sobald Sie die Aktualisierungen speichern. Sie erhalten keine Logs mehr, bis Sie die vollständige Konfiguration von Vended Logging abgeschlossen haben.

Schritt 4: Konfigurieren Sie die Protokollzustellung in CloudWatch Logs

In CloudWatch Logs müssen Sie drei Elemente erstellen, die die einzelnen Teile der Protokollzustellung darstellen. Diese Elemente werden in der Amazon CloudWatch Logs API-Referenz ausführlich beschrieben. [CreateDelivery](#) Die allgemeinen Schritte zur Konfiguration der Lieferung mit der CloudWatch Logs-API lauten wie folgt.

So konfigurieren Sie die Protokollzustellung in CloudWatch Logs (API)

1. Wird verwendet [PutDeliverySource](#), um die Quelle der Protokolle hinzuzufügen.

A `DeliverySource` steht für die Wiedergabekonfiguration, die die Protokolle generiert. Sie benötigen den Namen der Wiedergabekonfiguration, um die zu erstellen `DeliverySource`.

2. Wird verwendet [PutDeliveryDestination](#), um das Ziel hinzuzufügen, in das die Protokolle geschrieben werden.

A `DeliveryDestination` steht für das Lieferziel. Sie benötigen den ARN der Protokollgruppe, des Buckets oder des Streams, um den zu erstellen `DeliveryDestination`.

3. Verwenden Sie [PutDeliveryDestinationPolicy](#) diese Option, wenn Sie Logs kontenübergreifend bereitstellen.

Wenn sich das Lieferziel in einem anderen Konto als in der Wiedergabekonfiguration befindet, benötigen Sie ein `DeliveryDestinationPolicy`. Diese Richtlinie ermöglicht CloudWatch Logs die Übermittlung von Protokollen an die `DeliveryDestination`.

4. Wird verwendet [CreateDelivery](#), um das `DeliverySource` mit dem zu verknüpfen `DeliveryDestination`.

A `Delivery` steht für die Verbindung zwischen `DeliverySource` und `DeliveryDestination`.

Migrieren Sie Ihre AWS Elemental MediaTailor Protokollierungsstrategie

Wenn Sie die Protokollstrategie von Legacy CloudWatch auf Vended Logs ändern, MediaTailor wird diese Änderung vorgenommen, sobald Sie die Updates speichern. Gehen Sie wie folgt vor, um Unterbrechungen in Ihrem Logging-Workflow zu vermeiden, um Ihre Protokollierungsstrategie zu migrieren.

1. Führen Sie die unter [Verwendung von verkauften Protokollen](#) beschriebenen Schritt aus. Wählen Sie für [Aktivieren Sie Vended-Logs in MediaTailor](#) beide Protokollierungsstrategien (Vending Logs und Legacy CloudWatch) aus.

MediaTailor sendet Protokolle sowohl über verkaufte Logs als auch direkt an CloudWatch Logs.

2. Nehmen Sie die erforderlichen Änderungen in Ihrem Arbeitsablauf vor, die von Ihrer Protokollierungsstrategie und Ihrem Lieferziel abhängen.
3. Besuchen Sie Legacy erneut [Aktivieren Sie Vended-Logs in MediaTailor](#) und entfernen Sie es CloudWatch aus den Protokollierungsstrategien.

AWS Elemental MediaTailor Logs direkt in Amazon CloudWatch Logs schreiben

MediaTailor erstellt Protokolle, die detaillierte Informationen über Sitzungsaktivitäten und Interaktionen mit dem Anzeigenentscheidungsserver enthalten, und schreibt sie an Amazon CloudWatch. Die Protokolle enthalten eine sequentielle Beschreibung der Aktivitäten, die während der Sitzung stattfinden.

MediaTailor kann auch verkaufte Holzscheite verwenden, um Flexibilität bei der Holzlieferung und Mengenrabatte zu erzielen. Informationen zu verkauften Protokollen finden Sie unter. [Verwendung von verkauften Protokollen](#)

Themen

- [Berechtigungen für Amazon CloudWatch Logs](#)
- [Protokoll „As Run“ für AWS Elemental MediaTailor Channel Assembly](#)
- [AWS Elemental MediaTailor ADS-Protokollanalyse in Amazon CloudWatch Logs Insights](#)

Berechtigungen für Amazon CloudWatch Logs

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

Um den MediaTailor Zugriff zu ermöglichen CloudWatch

1. Öffnen Sie unter <https://console.aws.amazon.com/iam/> die IAM-Konsole.
2. Klicken Sie im Navigationsbereich der IAM-Konsole auf Rollen, und wählen Sie dann Rolle erstellen.
3. Wählen Sie den Rollentyp „Anderes AWS Konto“.
4. Geben Sie als Konto-ID Ihre AWS Konto-ID ein.
5. Wählen Sie Require external ID (Externe ID fordern) und geben Sie **Midas** ein. Mit dieser Option wird automatisch eine Bedingung zur Vertrauensrichtlinie hinzugefügt, die es dem Service nur dann ermöglicht, die Rolle anzunehmen, wenn in der Anforderung die richtige `sts:ExternalId` enthalten ist.
6. Wählen Sie Weiter: Berechtigungen aus.
7. Fügen Sie eine Berechtigungsrichtlinie hinzu, in der die Aktionen angegeben werden, die von dieser Rolle ausgeführt werden dürfen. Wählen Sie eine der folgenden Optionen aus und wählen Sie dann Next: Review (Nächster Schritt: Prüfen):
 - CloudWatchLogsFullAccessum vollen Zugriff auf Amazon CloudWatch Logs zu gewähren
 - CloudWatchFullAccessum vollen Zugriff auf Amazon zu gewähren CloudWatch
8. Geben Sie für Role name (Rollenname) den Namen **MediaTailorLogger** ein und klicken Sie auf Create role (Rolle erstellen).

9. Wählen Sie auf der Seite Roles (Rollen) die von Ihnen soeben erstellte Rolle aus.
10. Bearbeiten Sie die Vertrauensstellung, um den Prinzipal zu aktualisieren:
 1. Wählen Sie auf der Seite Summary (Übersicht) der Rolle die Registerkarte Trust relationship (Vertrauensstellung) aus.
 2. Wählen Sie Vertrauensstellung bearbeiten aus.
 3. Ändern Sie im Richtliniendokument den Principal in den MediaTailor Service. Das sollte wie folgt aussehen:

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

Die gesamte Richtlinie sollte folgendermaßen lauten:

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

Protokoll „As Run“ für AWS Elemental MediaTailor Channel Assembly

Das As Run-Protokoll in der CloudWatch MediaTailor/Channel/AsRunLog Protokollgruppe enthält Informationen zu Programmen und Werbeunterbrechungen, während sie abgespielt werden.

Wenn Sie einen Channel erstellen, ist das As Run-Protokoll standardmäßig deaktiviert. Mit der Konsole oder der AWS Command Line Interface (AWS CLI) können Sie den Protokollstatus „As Run“ für jeden Kanal in Ihrem Konto aktivieren und deaktivieren.

Wenn Sie das As-Run-Protokoll aktivieren, MediaTailor wird automatisch eine dienstbezogene Rolle erstellt, mit der Sie das As-Run-Protokoll in Ihrem CloudWatch Logs-Konto schreiben und verwalten können MediaTailor . Weitere Informationen zu serviceverknüpften Rollen finden Sie unter [Verwenden von serviceverknüpften Rollen für MediaTailor](#).

Note

Das As Run Log unterstützt derzeit nur das Standardprogramm. Derzeit werden die durch Programmregeln erstellten AlternateMedia nicht unterstützt. Dies bedeutet, dass das As Run Log für AlternateMedia derzeit nicht generiert wird.

Themen

- [Das As Run-Protokoll aktivieren](#)
- [Das As Run-Protokoll deaktivieren](#)

Das As Run-Protokoll aktivieren

Um das As Run-Protokoll zu aktivieren, geben Sie den Kanalnamen an und aktivieren Sie den Protokolltyp As Run für diesen Kanal.

Console

Um das As Run-Protokoll beim Erstellen eines Kanals zu aktivieren

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die MediaTailor Konsole unter <https://console.aws.amazon.com/mediatailor/>.
2. Wählen Sie im Navigationsbereich Channel-Assembly > Channels aus.
3. Wählen Sie in der Navigationsleiste die Option Kanal erstellen aus.
4. Konfigurieren Sie Ihren Kanal in den Bereichen Kanaldetails festlegen, Ausgänge konfigurieren und Zugriffskontrolle wie gewünscht.
5. Wählen Sie im Bereich Zugriffskontrolle die Option Weiter aus.
6. Wählen Sie im Bereich Protokollierung unter Protokolltypen die Option Als Ausführung aktivieren aus, um das Protokoll Als Ausführung zu aktivieren.

Um das As-Run-Protokoll bei der Aktualisierung eines Kanals zu aktivieren

Note

Wenn der Kanal gerade läuft, müssen Sie ihn zuerst beenden, bevor Sie ihn aktualisieren können. Nachdem Sie den Kanal gestoppt haben, können Sie „Aktionen“ > „Bearbeiten“ wählen, um mit der Aktualisierung des Kanals zu beginnen.

1. Melde dich bei an AWS-Managementkonsole und öffne die MediaTailor Konsole unter <https://console.aws.amazon.com/mediatailor/>.
2. Wählen Sie im Navigationsbereich Channel-Assembly > Channels aus.
3. Wählen Sie den Kanal aus, für den Sie das As Run-Protokoll aktualisieren möchten.
4. Wählen Sie Actions (Aktionen) und Edit (Bearbeiten).

5. Aktualisieren Sie in den Bereichen Kanaldetails festlegen, Ausgänge konfigurieren und Zugriffskontrolle Ihre Kanalkonfiguration wie gewünscht.
6. Wählen Sie im Bereich „Zugriffskontrolle“ die Option Weiter aus.
7. Wählen Sie im Bereich Protokollierung unter Protokolltypen die Option Als Ausführung aktivieren aus, um das Protokoll Als Ausführung zu aktivieren.

Um das As-Run-Protokoll auf der Registerkarte Protokollierung zu aktivieren

 Note

Wenn der Channel gerade läuft, müssen Sie die Registerkarte Logging verwenden, anstatt Aktionen > Bearbeiten zu wählen, um das As Run-Protokoll zu aktivieren.

1. Melden Sie sich bei an AWS-Managementkonsole und öffnen Sie die MediaTailor Konsole unter <https://console.aws.amazon.com/mediatailor/>.
2. Wählen Sie im Navigationsbereich Channel-Assembly > Channels aus.
3. Wählen Sie den Kanal aus, für den Sie das As Run-Protokoll aktivieren möchten.
4. Wählen Sie in der Navigationsleiste unter dem Namen des Kanals die Option Logging aus.
5. Wählen Sie unter Protokollierung > Protokolltypen die Option Als ausgeführt aus, um das Protokoll Als ausgeführt zu aktivieren.

AWS Command Line Interface (AWS CLI)

Um das As Run-Protokoll zu aktivieren

Führen Sie den [configure-logs-for-channel](#) Befehl aus und geben Sie die entsprechenden Werte für die erforderlichen Parameter an.

Dieses Beispiel ist für Linux, macOS oder Unix formatiert und verwendet den umgekehrten Schrägstrich (\) zur Verbesserung der Lesbarkeit.

```
$ aws mediatailor configure-logs-for-channel \  
--channel-name MyChannel \  
--log-types AS_RUN
```

Dieses Beispiel ist für Microsoft Windows formatiert und verwendet das Zeilenfortsetzungszeichen Caret (^), um die Lesbarkeit zu verbessern.

```
C:\> aws mediatailor configure-logs-for-channel ^  
--channel-name MyChannel ^  
--log-types AS_RUN
```

Wobei Folgendes gilt:

- *MyChannel* ist der Name des Kanals, der Ihnen gehört und für den Sie das As Run-Protokoll aktivieren möchten.

Wird der Befehl erfolgreich ausgeführt, erhalten Sie eine Ausgabe ähnlich der folgenden:

```
{  
  "ChannelName": "MyChannel",  
  "LogTypes": [  
    "AS_RUN"  
  ]  
}
```

Das As Run-Protokoll deaktivieren

Um das As-Run-Protokoll für einen Kanal zu deaktivieren, für den es aktiviert ist, geben Sie den Kanalnamen an und deaktivieren Sie den Protokolltyp As Run für diesen Kanal.

Console

Um das As Run-Protokoll bei der Aktualisierung eines Kanals zu deaktivieren

Note

Wenn der Kanal gerade läuft, müssen Sie ihn zuerst beenden, bevor Sie ihn aktualisieren können. Nachdem Sie den Kanal gestoppt haben, können Sie „Aktionen“ > „Bearbeiten“ wählen, um mit der Aktualisierung des Kanals zu beginnen.

1. Melde dich bei an AWS-Managementkonsole und öffne die MediaTailor Konsole unter <https://console.aws.amazon.com/mediatailor/>.

2. Wählen Sie im Navigationsbereich Channel-Assembly > Channels aus.
3. Wählen Sie den Kanal aus, für den Sie das As Run-Protokoll aktualisieren möchten.
4. Wählen Sie Actions (Aktionen) und Edit (Bearbeiten).
5. Aktualisieren Sie in den Bereichen Kanaldetails festlegen, Ausgänge konfigurieren und Zugriffskontrolle Ihre Kanalkonfiguration wie gewünscht.
6. Wählen Sie im Bereich „Zugriffskontrolle“ die Option Weiter aus.
7. Deaktivieren Sie im Bereich Protokollierung unter Protokolltypen die Option Als Ausführung aktivieren, um das Protokoll Als Ausführung zu deaktivieren.

Um das As-Run-Protokoll auf der Registerkarte Protokollierung zu deaktivieren

 Note

Wenn der Channel gerade läuft, müssen Sie die Registerkarte Logging verwenden, anstatt Aktionen > Bearbeiten zu wählen, um das As Run-Protokoll zu deaktivieren.

1. Melden Sie sich bei an AWS-Managementkonsole und öffnen Sie die MediaTailor Konsole unter <https://console.aws.amazon.com/mediatailor/>.
2. Wählen Sie im Navigationsbereich Channel-Assembly > Channels aus.
3. Wählen Sie den Kanal aus, für den Sie das As Run-Protokoll deaktivieren möchten.
4. Wählen Sie in der Navigationsleiste unter dem Namen des Kanals die Option Logging aus.
5. Deaktivieren Sie unter Protokollierung > Protokolltypen die Option Als ausgeführt, um das Protokoll Als ausgeführt zu deaktivieren.

AWS Command Line Interface (AWS CLI)

Um das As-Run-Protokoll zu deaktivieren

Führen Sie den [configure-logs-for-channel](#) Befehl aus und geben Sie die entsprechenden Werte für die erforderlichen Parameter an.

Dieses Beispiel ist für Linux, macOS oder Unix formatiert und verwendet den umgekehrten Schrägstrich (\) zur Verbesserung der Lesbarkeit.


```
$ aws mediatailor configure-logs-for-channel \  
--channel-name MyChannel \  
--log-types
```

Dieses Beispiel ist für Microsoft Windows formatiert und verwendet das Zeilenfortsetzungszeichen Caret (^), um die Lesbarkeit zu verbessern.

```
C:\> aws mediatailor configure-logs-for-channel ^  
--channel-name MyChannel ^  
--log-types
```

Wobei Folgendes gilt:

- *MyChannel* ist der Name des Kanals, dessen Eigentümer Sie sind und für den Sie das As Run-Protokoll deaktivieren möchten.

Wird der Befehl erfolgreich ausgeführt, erhalten Sie eine Ausgabe ähnlich der folgenden:

```
{  
  "ChannelName": "MyChannel",  
  "LogTypes": []  
}
```

AWS Elemental MediaTailor ADS-Protokollanalyse in Amazon CloudWatch Logs Insights

Mit Amazon CloudWatch Logs Insights können Sie AWS Elemental MediaTailor ADS-Protokolle (Ad Decision Server) anzeigen und abfragen. MediaTailor sendet Ereignisprotokolle CloudWatch für normale Verarbeitungs- und Fehlerbedingungen an. Die Protokolle entsprechen einem JSON-Schema. Mit CloudWatch Logs Insights können Sie Protokolle nach Zeitrahmen auswählen und dann Abfragen für sie ausführen.

Allgemeine Informationen finden Sie unter [Analysieren von Protokolldaten mit CloudWatch Logs Insights](#).

 Note

Um auf die Protokolle zugreifen zu können, benötigen Sie Berechtigungen für den Zugriff auf Amazon CloudWatch. Detaillierte Anweisungen finden Sie unter [Berechtigungen für Amazon CloudWatch Logs](#).

Um ADS-Protokolle über die CloudWatch Konsole anzuzeigen und abzufragen

1. Öffnen Sie die CloudWatch Konsole unter <https://console.aws.amazon.com/cloudwatch/>.
2. Wählen Sie im Navigationsbereich unter Logs, (Protokolle) die Option Insights(Einblicke) aus.
3. Geben Sie in der Suchleiste ein **AdDec**, und wählen Sie dann in der Dropdownliste `awsMediaTailor/AdDecisionServerInteractions`.
4. (Optional) Passen Sie den Zeitraum an, den Sie untersuchen möchten.
5. (Optional) Ändern Sie die Abfrage im Dialogfeld. Allgemeine Hinweise finden Sie unter [CloudWatch Logs Insights-Abfragesyntax](#). Beispiele für Abfragen für MediaTailor ADS finden Sie unter [Abfragen der ADS-Protokolle](#).
6. Wählen Sie Abfrage ausführen. Die Abfrage kann einige Sekunden dauern. Während dieser Zeit erscheint Cancel (Abbrechen) anstelle von Run query (Abfrage ausführen).
7. (Optional) Um die Ergebnisse als CSV-Datei zu exportieren, wählen Sie Actions (Aktionen) und dann Download query results (CSV) (Abfrageergebnisse herunterladen) aus.

 Note

Die Konsole begrenzt die Anzahl der Datensätze, die sie in Abfrageergebnissen zurückgibt und die sie exportiert. Verwenden Sie daher für Massendaten die API, das AWS Command Line Interface (AWS CLI) oder ein SDK.

Themen

- [Abfragen der ADS-Protokolle](#)

Abfragen der ADS-Protokolle

CloudWatch Logs Insights bietet eine Vielzahl von Optionen zum Abfragen Ihrer Logs. Ausführliche Informationen zur Abfragesyntax finden Sie unter [CloudWatch Logs Insights-Abfragesyntax](#). In diesem Abschnitt finden Sie Beispiele für gängige Abfragen für den Einstieg in Ihre ADS-Protokollabfragen. Alle Abfragen werden über die Protokolle für die aktuelle Zeitbereichseinstellung ausgeführt.

Die folgende Abfrage ruft alle Informationen aus den ADS-Protokollen ab.

```
fields @timestamp, eventType, sessionId, requestId, @message
| sort sessionId, @timestamp asc
```

Die folgende Abfrage ruft alle Anforderungen an den ADS ab. Diese Abfrage zeigt eine Möglichkeit, den Inhalt des Anforderungsheaders für MediaTailor Logs abzurufen.

```
fields @timestamp, adsRequestUrl, requestHeaders.0.value as @userAgent,
requestHeaders.1.value as @xForwardedFor, sessionId, requestId
| filter eventType = "MAKING_ADS_REQUEST"
| sort @timestamp asc
```

Die folgende Abfrage ruft die Anzeigen ab, die für eine bestimmte Sitzung MediaTailor eingefügt wurden.

```
fields @timestamp, sessionId, requestId, @message
| filter eventType = "FILLED_AVAIL"
| sort @timestamp asc
```

Die folgende Abfrage ruft das Tracking ab URLs , das im MediaTailor Namen des Spielers aufgerufen wurde.

```
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
```

Die folgende Abfrage ruft Informationen für eine bestimmte Wiedergabebesitzung ab, indem die Ergebnisse nach sessionId gefiltert werden.

```
fields @timestamp, eventType, sessionId, requestId, @message
| filter sessionId = "0aaf6507-c6f9-4884-bfe7-f2f841cb8195"
| sort @timestamp asc
```

Die folgende Abfrage ruft Informationen für eine einzelne Anforderung ab, indem die Ergebnisse nach `requestId` gefiltert werden.

```
fields @timestamp, eventType, sessionId, requestId, @message
| filter requestId = "f5d3cf39-6258-4cf1-b3f6-a34ff8bf641d"
| sort @timestamp asc
```

Die folgende Abfrage ruft die Anzahl der Protokolleinträge für jeden Ereignistyp ab, der protokolliert wurde.

```
fields eventType
| stats count() as @eventCount by eventType
```

Die folgende Abfrage ruft die Avail-ID und die Liste übersprungener Anzeigen für alle Avails ab, für die Anzeigen übersprungen wurden.

```
fields avail.availId
| parse @message '"skippedAds":[*]' as @skippedAdsList
| filter ispresent(@skippedAdsList)
```

Steuerung des Volumens der Logs AWS Elemental MediaTailor

MediaTailor Sitzungsprotokolle zum Einfügen von Anzeigen sind manchmal ausführlich. Um die Protokollkosten zu senken, können Sie den Prozentsatz der Sitzungsprotokolle definieren, die MediaTailor an Amazon CloudWatch Logs gesendet werden. Wenn Ihre Wiedergabekonfiguration beispielsweise 1000 Sitzungen zum Einfügen von Werbeanzeigen umfasst und Sie einen aktivierten Prozentwert von festlegen60, werden Protokolle für 600 der Sitzungen an CloudWatch Logs MediaTailor gesendet. MediaTailor entscheidet nach dem Zufallsprinzip, für welche der Sitzungen Protokolle gesendet werden sollen. Wenn Sie Protokolle für eine bestimmte Sitzung anzeigen möchten, können Sie den [Debug-Protokollmodus](#) verwenden.

Wenn Sie einen Prozentsatz für die Protokollierung festlegen, MediaTailor wird automatisch eine mit dem Dienst verknüpfte Rolle erstellt, MediaTailor die die zum Schreiben von CloudWatch Protokollen

in Ihr Konto erforderlichen Berechtigungen gewährt. Informationen zur MediaTailor Verwendung von dienstbezogenen Rollen finden Sie unter. [Verwenden von serviceverknüpften Rollen für MediaTailor](#)

Eine Protokollkonfiguration erstellen

Um den Prozentsatz der Sitzungsprotokolle zu kontrollieren, die in CloudWatch Protokolle MediaTailor geschrieben werden, erstellen Sie eine Protokollkonfiguration für Ihre Wiedergabekonfiguration. Wenn Sie eine Protokollkonfiguration erstellen, geben Sie einen Namen für die Wiedergabekonfiguration und einen aktivierten Prozentwert an.

Console

Um eine Protokollkonfiguration für eine bestehende Wiedergabekonfiguration zu erstellen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die MediaTailor Konsole unter <https://console.aws.amazon.com/mediatailor/>.
2. Wählen Sie im Bereich Wiedergabekonfiguration die Wiedergabekonfiguration aus, für die Sie die Protokollkonfiguration festlegen möchten.
3. Wählen Sie Bearbeiten aus.
4. Geben Sie unter Protokollkonfiguration einen Wert für die Aktivierung in Prozent an.

Um eine Protokollkonfiguration für eine neue Wiedergabekonfiguration zu erstellen

- Folgen Sie dem Verfahren unter [Konfiguration protokollieren](#).

AWS Command Line Interface (AWS CLI)

Um eine Protokollkonfiguration für eine bestehende Wiedergabekonfiguration zu erstellen

Um eine Protokollkonfiguration mit dem zu erstellen AWS CLI, führen Sie den Befehl [configure-logs-for-playback-configuration](#) aus und geben Sie die entsprechenden Werte für die erforderlichen Parameter an.

Dieses Beispiel ist für Linux, macOS oder Unix formatiert und verwendet den umgekehrten Schrägstrich (\) zur Verbesserung der Lesbarkeit.

```
$ aws mediatailor configure-logs-for-playback-configuration \
--percent-enabled 10 \
```

```
--playback-configuration-name MyPlaybackConfiguration
```

Dieses Beispiel ist für Microsoft Windows formatiert und verwendet das Zeilenfortsetzungszeichen Caret (^), um die Lesbarkeit zu verbessern.

```
C:\> aws mediatailor configure-logs-for-playback-configuration ^  
--percent-enabled 10 ^  
--playback-configuration-name MyPlaybackConfiguration
```

Wobei Folgendes gilt:

- *percent-enabled* ist der Prozentsatz der Sitzungsprotokolle zur Wiedergabekonfiguration, der an Logs gesendet wird. MediaTailor CloudWatch
- *playback-configuration-name* ist der Name der Wiedergabekonfiguration, für die die Protokollkonfigurationseinstellungen festgelegt werden sollen.

Wird der Befehl erfolgreich ausgeführt, erhalten Sie eine Ausgabe ähnlich der folgenden:

```
{  
  "PercentEnabled": 10,  
  "PlaybackConfigurationName": "MyPlaybackConfiguration"  
}
```

Um eine Protokollkonfiguration für eine neue Wiedergabekonfiguration zu erstellen

- Verwenden Sie die `configure-logs-for-playback-configuration` Option für den [put-playback-configuration](#) Befehl.

Deaktivierung einer Protokollkonfiguration

Nachdem Sie eine Protokollkonfiguration erstellt haben, können Sie sie nicht löschen — Sie können sie nur deaktivieren. Um die Protokollkonfiguration zu deaktivieren, setzen Sie in der MediaTailor Konsole oder API den Wert für die Aktivierung in Prozent auf 0. Dadurch wird die gesamte Sitzungsprotokollierung für diese Wiedergabekonfiguration deaktiviert.

Wenn Sie die dienstbezogene Rolle löschen möchten, die für die Protokollkonfiguration (en) in Ihrem Konto MediaTailor verwendet wird, müssen Sie zunächst alle Protokollkonfigurationen

deaktivieren. Informationen zum Löschen der dienstverknüpften Rolle finden Sie unter. [Verwenden von serviceverknüpften Rollen für MediaTailor](#)

Console

So deaktivieren Sie die Protokollkonfiguration für eine Wiedergabekonfiguration

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die MediaTailor Konsole unter <https://console.aws.amazon.com/mediatailor/>.
2. Wählen Sie im Bereich Wiedergabekonfiguration die Wiedergabekonfiguration aus, für die Sie die Protokollkonfiguration deaktivieren möchten.
3. Wählen Sie Bearbeiten aus.
4. Stellen Sie unter Protokollkonfiguration den Wert für Prozent aktiviert auf ein 0. Dadurch wird die gesamte Sitzungsprotokollierung für diese Wiedergabekonfiguration deaktiviert.
5. Wählen Sie Speichern aus.

AWS Command Line Interface (AWS CLI)

Um eine Protokollkonfiguration zu deaktivieren

- Stellen Sie den `percent-enabled` Wert 0 mit dem Befehl [configure-logs-for-playback-configuration](#) auf ein.

Filterung von AWS Elemental MediaTailor Protokollen und Ereignissen

Protokolle, die von einer Wiedergabekonfiguration in ausgegeben werden, MediaTailor enthalten Informationen über eine Vielzahl von Aktivitäten, die während der Wiedergabesitzung stattfinden. Diese Aktivitäten werden im Ereignistyp der Protokolle identifiziert. Viele Ereignisse werden standardmäßig protokolliert. Um die Kosten für Logs in Amazon unter Kontrolle zu halten CloudWatch, können Sie angeben, welche Logs ausgegeben MediaTailor werden.

MediaTailor bietet Ihnen die Kontrolle über die Protokollfilterung, sodass Sie Folgendes tun können:

- Geben Sie die Protokollereignisse an, die Sie aus den Protokollen ausschließen möchten
- Aktivieren Sie die Protokollierung von Rohantworten vom AD Decision Server (ADS)

Sie können diese Einstellungen für die Protokollfilterung unabhängig für jede Wiedergabesitzung oder als Standard für alle Wiedergabesitzungen für eine Wiedergabekonfiguration festlegen.

- Um Protokolle pro Sitzung zu filtern, fügen Sie Abfrageparameter in die Initialisierungsanfrage für die Wiedergabe-Sitzung ein.
- Um Protokolle auf Basis der Konfiguration pro Wiedergabe zu filtern, verwenden Sie die MediaTailor Konsole oder die API, um Ihre Präferenzen in den Einstellungen für die Wiedergabekonfiguration anzugeben.

Die folgenden Abschnitte enthalten Anweisungen zur Aktivierung der Protokollfilterung für Sitzungen und Wiedergabekonfigurationen.

Protokollfilter pro Sitzung

Um für jede Sitzung eine benutzerdefinierte Ebene der Protokolldetails zu definieren, fügen Sie die folgenden Parameter an Ihre erste server- oder clientseitige Wiedergabe-Sitzungsanforderung an. Fügen Sie den Parametern Werte hinzu, um die Ereignisse darzustellen, die Sie ein- oder ausschließen möchten, und zwar in einem kommasetrennten Format:

- `aws.adsInteractionLogPublishOptInEventTypesum` Protokolle für bestimmte Interaktionen mit dem AD Decision Server (ADS) zu erhalten.
- `aws.adsInteractionLogExcludeEventTypesum` den Empfang von Protokollen für bestimmte ADS-Interaktionen zu beenden.
- `aws.manifestServiceLogExcludeEventTypesum` den Empfang von Protokollen für bestimmte Manifest-Dienstinteraktionen zu beenden.

Eine Liste der Protokoll- und MediaTailor Ereignistypen, die ausgesendet werden, finden Sie unter [Manifest-ProtokolleADS-Protokolle](#), und [Protokolle transkodieren](#).

Wenn Sie keine Abfrageparameter für die Protokollfilterung übergeben, werden alle Protokolle an Ihr Lieferziel MediaTailor geschrieben.

Example serverseitige Sitzungsinitialisierung mit Protokollfiltern

Um `PARSING_ERROR` Ereignisse aus Ihren Manifestprotokollen und `MAKING_ADS_REQUEST` aus den ADS-Protokollen auszuschließen `GENERATED_MANIFEST`, würde die Anfrage zur Sitzungsinitialisierung wie folgt aussehen:


```
GET <mediatailorURL>/v1/master/<hashed-account-id>/<origin-id>/index.m3u8?
aws.logMode=DEBUG&aws.manifestServiceLogExcludeEventTypes=GENERATED_MANIFEST,PARSING_ERROR&aws.
```

Um Rohprotokolle aus Ihrem ADS zu aktivieren, geben Sie den RAW_ADS_RESPONSE Wert für den AdsInteractionPublishOptInEventType Parameter an:

```
GET <mediatailorURL>/v1/master/<hashed-account-id>/<origin-id>/index.m3u8?
aws.adsInteractionPublishOptInEventType=RAW_ADS_RESPONSE
```

Example clientseitige Sitzungsinitialisierung mit Protokollfiltern

Um Protokollereignisse während der clientseitigen Sitzungsinitialisierung auszuschließen, fügen Sie Parameter in die POST-Anfrage Ihres Clients ein `availSuppression` und protokollieren Sie sie. MediaTailor Weitere Hinweise zum Erstellen einer clientseitigen Wiedergabesitzungsanforderung finden Sie unter [Kundenseitiges Anzeigen-Tracking](#). Im folgenden Beispiel werden PARSING_ERROR Ereignisse aus Ihren Manifestprotokollen CONFIG_SECURITY_ERROR und aus den ADS-Protokollen ausgeschlossen. MAKING_ADS_REQUEST

```
POST parent.m3u8
{
  "adsInteractionLog": {
    ...
    "excludeEventTypes": [
      "MAKING_ADS_REQUEST"
    ]
  },
  "manifestServiceLog": {
    ...
    "excludeEventTypes": [
      "GENERATED_MANIFEST",
      "PARSING_ERROR"
    ]
  },
  "logMode": "DEBUG"
}
```

Um Rohprotokolle aus Ihrem ADS zu aktivieren, geben Sie den RAW_ADS_RESPONSE Wert für den publishOptInEventTypes Parameter an:

```
POST parent.m3u8
```

```
{
  "adsInteractionLog": {
    "publishOptInEventTypes": ["RAW_ADS_RESPONSE"],
    "excludeEventTypes": [
      "MAKING_ADS_REQUEST"
    ]
  },
  "manifestServiceLog": {
    ...
    "excludeEventTypes": [
      "GENERATED_MANIFEST",
      "PARSING_ERROR"
    ]
  },
  "logMode": "DEBUG"
}
```

Protokollfilter für die Konfiguration pro Wiedergabe

Verwenden Sie die Einstellungen der Wiedergabekonfiguration, um die Protokoll-Ereignistypen zu definieren, die MediaTailor standardmäßig von dieser Wiedergabekonfiguration ausgegeben werden. MediaTailor verwendet diese Standardeinstellungen für die Protokollfilterung für alle Sitzungen, die keine Filterabfrageparameter in der Sitzungsstartanforderung enthalten.

Sie können wählen, ob Sie Folgendes tun möchten:

- Empfangen Sie Protokolle für bestimmte Interaktionen mit dem Ad Decision Server (ADS).
- Schließen Sie Protokolle für bestimmte ADS-Interaktionen aus.
- Schließen Sie Protokolle für bestimmte manifeste Dienstinteraktionen aus.

Informationen zum Einstellen dieser Einstellungen von der MediaTailor Konsole aus finden Sie unter [Konfiguration erstellen](#). Informationen zur MediaTailor API finden Sie [PutPlaybackConfiguration](#) in der AWS Elemental MediaTailor API-Referenz.

Eine Liste der Protokoll- und MediaTailor Ereignistypen, die ausgegeben werden, finden Sie unter [Manifest-Protokolle](#), [ADS-Protokolle](#), und [Protokolle transkodieren](#).

AWS Elemental MediaTailor Debug-Protokolle werden generiert

Verwenden Sie Debug-Protokolle, um Probleme bei der Wiedergabesitzung beim Einfügen von MediaTailor Anzeigen zu beheben. Um Debug-Logs zu generieren, stellen Sie in der Anfrage des

Players den Protokollmodus auf Debuggen ein. MediaTailor Stellen Sie für serverseitige Berichte den Protokollmodus in der Wiedergabeanforderung ein. Für clientseitige Berichte legen Sie den Protokollmodus in der Anforderung zur Sitzungsinitialisierung fest.

Wenn der Protokollmodus auf Debug gesetzt ist, werden alle Protokollereignistypen in das Verzeichnis Logs MediaTailor geschrieben. CloudWatch Die Protokolle enthalten Informationen zu den folgenden Ereignissen. Eine vollständige Liste der in den Debug-Protokollen erzeugten Daten finden Sie unter [Debug-Protokollfelder](#).

- Origin-Interaktion — Details zu MediaTailor Interaktionen mit dem Ursprungsserver. Zum Beispiel die Antwort auf das Ursprungs-Manifest, der Manifesttyp und die Quell-URL.
- Generiertes Manifest — Details zur Antwort auf die Playback-Sitzung von MediaTailor. Zum Beispiel das Manifest, das MediaTailor generiert wird.
- Sitzung initialisiert — Details zur Sitzungsinitialisierung, z. B. die Sitzungs-ID.

Informationen zum Anpassen der Protokollereignistypen, die Sie pro Sitzung erhalten, finden Sie unter [Filterung von Protokollen und Ereignissen](#)

Voraussetzungen

Um den Protokollmodus auf Debug einzustellen, müssen Sie zunächst die MediaTailor Erlaubnis zum Senden von Protokollen an erteilen CloudWatch, falls Sie dies noch nicht getan haben. Sobald Sie die Zugriffsberechtigung MediaTailor erteilt haben CloudWatch, können Sie den Debug-Protokollmodus aktivieren. Informationen zum Erteilen von MediaTailor Zugriffsberechtigungen CloudWatch finden Sie unter [Berechtigungen für Amazon einrichten CloudWatch](#).

Wie stelle ich den Protokollmodus auf Debug ein

In diesem Abschnitt wird erklärt, wie Sie den Protokollmodus für serverseitige und clientseitige Berichte auf Debug einstellen.

Serverseitige Berichterstattung

Fügen Sie für serverseitige Berichte den `?aws.logMode=DEBUG` Abfrageparameter und den Wert in die Wiedergabeanforderung Ihres GET HTTP Players an den HLS- oder DASH-Endpunkt ein. MediaTailor [Allgemeine Informationen zur serverseitigen Berichterstattung finden Sie unter Serverseitige Berichterstattung](#).

 Important

Bei DEBUG-Wert ist die Groß- und Kleinschreibung zu beachten.

Eine Wiedergabeanforderung, die Folgendes beinhaltet, ?aws.logMode=DEBUG sieht wie folgt aus:

Example Wiedergabeanforderung an einen HLS-Endpunkt

```
GET <mediatailorURL>/v1/master/<hashed-account-id>/<origin-id>/<asset-id>?  
aws.logMode=DEBUG
```

Nachdem Sie den Protokollmodus auf Debug gesetzt haben, empfehlen wir Ihnen, zu überprüfen, ob die Debug-Protokollierungssitzung aktiv ist. Um zu überprüfen, ob die Debug-Sitzung aktiv ist, überprüfen Sie, ob CloudWatch Protokolle für die Sitzungs-ID vorhanden sind. Die Sitzungs-ID ist im Wiedergabeendpunkt enthalten, der dies MediaTailor bereitstellt. Weitere Informationen finden Sie unter [Verify that the debug log mode is active for your playback session](#).

Clientseitige Berichterstattung

Für die clientseitige Berichterstattung fügen Sie den logMode Schlüssel und den DEBUG Wert in den Text der Anfrage zur POST HTTP Sitzungsinitialisierung Ihres Clients in den /v1/session-Endpunkt ein. MediaTailor [Allgemeine Informationen zur clientseitigen Berichterstattung finden Sie unter Clientseitige Berichterstattung](#).

 Important

Bei DEBUG-Wert ist die Groß- und Kleinschreibung zu beachten.

Nachdem Sie den Protokollmodus auf Debug gesetzt haben, empfehlen wir Ihnen, zu überprüfen, ob die Debugsitzung aktiv ist. Um zu überprüfen, ob die Debug-Sitzung aktiv ist, vergewissern Sie sich, dass in den Protokollen ein SESSION_INITIALIZED Ereignis mit der Sitzungs-ID verknüpft ist. CloudWatch Die Sitzungs-ID ist im Wiedergabeendpunkt enthalten, der Folgendes MediaTailor bereitstellt. Weitere Informationen finden Sie unter [Verify that the debug log mode is active for your playback session](#).

Maximale Anzahl aktiver Debug-Sitzungen

Sie können maximal 10 aktive Debug-Protokollsitzungen haben. Wenn Ihr Player seine Anfrage zur Sitzungsinitialisierung oder Wiedergabe an sendet MediaTailor, wird MediaTailor überprüft, ob das Limit erreicht wurde. Ist dies der Fall, wird MediaTailor überprüft, ob es veraltete Sessions gibt. Eine Sitzung ist veraltet, wenn innerhalb eines bestimmten Zeitraums nicht darauf zugegriffen wurde. Bei Live-Streams beträgt dieser Zeitraum 10 Minuten, bei VOD-Streams 30 Minuten.

Wenn das maximale Limit für aktive Debug-Protokollsitzungen erreicht wurde, werden Debug-Protokolle für Ihre Sitzung nicht in die CloudWatch Protokolle geschrieben. Wenn Sie in den Protokollen für Ihre Sitzung keine CloudWatch Debug-Protokolle sehen, haben Sie dieses Limit möglicherweise erreicht. Informationen darüber, ob das Limit erreicht wurde, finden Sie unter [Verify that the debug log mode is active for your playback session](#).

Debuggen von Protokollfeldern

In der folgenden Tabelle sind die Debug-Logfelder aufgeführt, in die MediaTailor geschrieben wird. CloudWatch

Feld	Description
awsAccountId	Ihre AWS-Konto ID.
customerId	Ihre MediaTailor Kundennummer.
eventTimestamp	Der ISO 8601-Zeitstempel, der dem Debug-Log-Ereignis zugeordnet ist.
eventType	Der Typ des Debug-Protokollereignisses. Werte: • ORIGIN_INTERACTION — Details zu MediaTailor Interaktionen mit dem Ursprungsserver. Zum Beispiel die Antwort auf das Ursprungs-Manifest, der Manifesttyp und die Quell-URL. • GENERATED_MANIFEST — Details zur Antwort auf die Wiedergabesitzung von MediaTailor. Zum Beispiel das Manifest, das MediaTailor generiert wird.

Feld	Description
	• <code>SESSION_INITIALIZED</code> — Details zur Sitzungsinitialisierung, z. B. die Sitzungs-ID.
<code>originRequestUrl</code>	Die URL Ihres Ursprungsservers, der für diese Anfrage abgerufen wird.
<code>mediaTailorPath</code>	Der MediaTailor Endpunkt, der aufgerufen wurde, einschließlich aller Parameter, die MediaTailor in der ersten Manifestanforderung übergeben wurden.
<code>requestId</code>	Die ID einer bestimmten HTTP-Anfrage an MediaTailor.
<code>responseBody</code>	Das Manifest im Antworttext von MediaTailor. Dies ist entweder das rohe Ursprungsmanifest oder das von generierte Manifest MediaTailor.
<code>sessionId</code>	Die ID der Wiedergabe-Sitzung.
<code>sessionType</code>	Die Art der Wiedergabesitzung. Werte: HLS, DASH

Lesen Sie die Debug-Protokolle

MediaTailor schreibt die Debug-Logs in Amazon CloudWatch Logs. Es fallen typische CloudWatch Logs-Gebühren an. Verwenden Sie CloudWatch Insights, um die Debug-Logs zu lesen.

Informationen zur Verwendung von CloudWatch Logs Insights finden Sie unter [Analysieren von Protokolldaten mit CloudWatch Logs Insights](#) im AWS CloudWatch Logs-Benutzerhandbuch.

Note

Es kann einige Minuten dauern, bis die Debug-Protokolle angezeigt werden. CloudWatch Wenn Sie die Protokolle nicht sehen, warten Sie ein paar Minuten und versuchen Sie es erneut. Wenn Sie die Protokolle immer noch nicht sehen, haben Sie möglicherweise die maximale Anzahl aktiver Debug-Protokollsitzungen erreicht. Um zu überprüfen, ob dies der Fall ist, führen Sie eine CloudWatch Abfrage aus, um festzustellen, ob für Ihre Playback-

Sitzung eine Debug-Sitzung initialisiert wurde. Weitere Informationen finden Sie unter [Verify that the debug log mode is active for your playback session](#).

Beispiele

Dieser Abschnitt enthält Beispielabfragen, mit denen Sie MediaTailor Debug-Protokolldaten lesen können.

Example 1: Stellen Sie sicher, dass der Debug-Protokollmodus für Ihre Wiedergabesitzung aktiv ist

```
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: Sehen Sie sich die Antworten von Ihrem Absender an

```
fields @timestamp, responseBody, @message, mediaTailorPath
| filter eventType = "ORIGIN_MANIFEST" and sessionId = "32002de2-837c-4e3e-9660-f3075e8dfd90"
```

Example 3: Sehen Sie sich das Manifest an, das von MediaTailor für eine bestimmte Sitzung generiert wurde

```
fields @timestamp, responseBody, @message
| filter mediaTailorPath like "/v1/master/" and eventType = "GENERATED_MANIFEST" and
sessionId = "32002de2-837c-4e3e-9660-f3075e8dfd90"
```

Example 4: Alle Ereignisse für eine bestimmte Zeit anzeigen **requestId**

Verwenden Sie diese Abfrage, um das Ursprungsmanifest und das von generierte Manifest anzuzeigen MediaTailor.

```
fields @timestamp, responseBody, @message, mediaTailorPath
| filter requestId = "e5ba82a5-f8ac-4efb-88a0-55bed21c45b4"
```

Überwachung AWS Elemental MediaTailor mit CloudWatch

Amazon-Metriken

Sie können AWS Elemental MediaTailor Metriken überwachen mit CloudWatch. CloudWatch sammelt Rohdaten über die Leistung des Dienstes und verarbeitet diese Daten zu lesbaren Metriken, die nahezu in Echtzeit verfügbar sind. Diese Statistiken werden 15 Monate gespeichert, damit Sie auf Verlaufsinformationen zugreifen können und einen besseren Überblick darüber erhalten, wie Ihre Webanwendung oder der Service ausgeführt werden. Sie können auch Alarme einrichten, die auf bestimmte Grenzwerte achten und Benachrichtigungen senden oder Aktivitäten auslösen, wenn diese Grenzwerte erreicht werden. Weitere Informationen finden Sie im [CloudWatch Amazon-Benutzerhandbuch](#).

Metriken können nützlich sein, wenn Sie veraltete Manifeste untersuchen. Weitere Informationen finden Sie unter [Verwendung von Metriken zur Diagnose veralteter Manifeste](#).

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

So zeigen Sie Metriken mit der Konsole an CloudWatch

1. Öffnen Sie die CloudWatch Konsole unter <https://console.aws.amazon.com/cloudwatch/>.
2. Wählen Sie im Navigationsbereich Metriken aus.
3. Wählen Sie unter Alle Metriken den MediaTailorNamespace aus.
4. Wählen Sie zur Ansicht der Metriken die Metrikdimension aus (z. B. originID).
5. Geben Sie den Zeitraum an, den Sie anzeigen möchten.

Um Metriken mit dem AWS Command Line Interface (AWS CLI) anzuzeigen

- Geben Sie in einer Eingabeaufforderung den folgenden Befehl ein:

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

AWS Elemental MediaTailor CloudWatch Metriken

Der AWS Elemental MediaTailor Namespace umfasst die folgenden Metriken. Diese Metriken werden standardmäßig in Ihrem Konto veröffentlicht.

Metriken von Channel Assembly (CA)

In der folgenden Tabelle sind alle Metriken nach Kanal oder nach Kanalausgabe verfügbar.

Metrik	Description
<code>4xxErrorCount</code>	Die Anzahl der 4xx Fehler.
<code>5xxErrorCount</code>	Die Anzahl der 5xx Fehler.
<code>RequestCount</code>	Die Gesamtanzahl der -Anforderungen. Die Anzahl der Transaktionen hängt weitgehend davon ab, wie oft Spieler aktualisierte Manifeste anfordern und wie viele Spieler sie haben. Jede Player-Anforderung zählt als eine Transaktion.
<code>TotalTime</code>	Die Zeit, die der Anwendungsserver für die Bearbeitung der Anfrage benötigte, einschließlich der Zeit, die für den Empfang und das Schreiben von Bytes vom Client und vom Netzwerk benötigt wurde.

Metriken zum serverseitigen Einfügen von Werbung (SSAI)

In der folgenden Tabelle sind Metriken zum Einfügen von serverseitiger Werbung aufgeführt.

Metrik	Description
<code>AdDecisionServer.Ads</code>	Die Anzahl der Anzeigen, die in den Antworten des Ad Decision Servers (ADS) innerhalb des von Ihnen CloudWatch angegebenen Zeitraums enthalten waren.
<code>AdDecisionServer.Duration</code>	Die Gesamtdauer in Millisekunden aller Anzeigen, die innerhalb des von Ihnen angegebenen CloudWatch Zeitraums vom ADS MediaTailor empfangen wurden. Diese Dauer

Metrik	Description
	kann länger als die von Ihnen angegebene Dauer <code>Avail.Duration</code> sein.
<code>AdDecisionServer.Errors</code>	Die Anzahl der Antworten mit 200-Statuscode, die nicht über HTTP erfolgen, der leeren Antworten und der Antworten mit Timeout, die innerhalb des von Ihnen angegebenen CloudWatch Zeitraums vom ADS MediaTailor empfangen wurden.
<code>AdDecisionServer.FillRate</code>	Der einfache Durchschnitt der Raten, mit denen die Antworten vom ADS die entsprechenden einzelnen Ad-Avails für den von Ihnen angegebenen Zeitraum gefüllt haben. Um den gewichteten Durchschnittswert zu erhalten, berechnen Sie die <code>AdDecisionServer.Duration</code> als Prozentsatz der <code>Avail.Duration</code>. Weitere Informationen über einfache und gewichtete Durchschnittswerte finden Sie unter Einfache und gewichtete Durchschnittswerte.
<code>AdDecisionServer.Latency</code>	Die Antwortzeit in Millisekunden für Anfragen an das ADS. MediaTailor
<code>AdDecisionServer.Timeouts</code>	Die Anzahl der Anfragen an das ADS, bei denen das Timeout in dem von Ihnen angegebenen CloudWatch Zeitraum überschritten wurde.

Metrik	Description
AdNotReady	Gibt an, wie oft der ADS in dem von Ihnen angegebenen Zeitraum auf eine Anzeige verwiesen hat, die von dem internen Transcode r-Service noch nicht transcodiert worden war. Ein hoher Wert für diese Metrik kann zu einem niedrigen Gesamtwert für Avail.FillRate betragen.
AdsBilled	Die Anzahl der Anzeigen, für die Kunden MediaTailor Rechnungen auf der Grundlage der Insertion in Rechnung gestellt werden.
Avail.Duration	Die geplante Gesamtzahl der Millisekunden, an denen die Anzeige innerhalb des Zeitraums verfügbar ist. CloudWatch Die geplante Gesamtzahl basiert auf der Dauer der Anzeigenverfügbarkeit im Ursprungsmanifest.
Avail.FilledDuration	Die geplante Anzahl von Millisekunden an Werbeanzeigen, die innerhalb des Zeitraums mit MediaTailor Werbeanzeigen gefüllt werden. CloudWatch

Metrik	Description
<code>Avail.FillRate</code>	Der geplante einfache Durchschnitt der Raten, zu denen einzelne Anzeigen innerhalb des Zeitraums geschaltet MediaTailor werden. CloudWatch Um den gewichteten Durchschnittswert zu erhalten, berechnen Sie die <code>Avail.FilledDuration</code> als Prozentsatz der <code>Avail.Duration</code>. Weitere Informationen über einfache und gewichtete Durchschnittswerte finden Sie unter Einfache und gewichtete Durchschnittswerte. Welchen Wert MediaTailor maximal für <code>Avail.FillRate</code> erzielen kann, ist von dem Wert für <code>AdDecisionServer.FillRate</code> abhängig. Wenn der Wert für <code>Avail.FillRate</code> niedrig ist, vergleichen Sie ihn mit dem Wert für <code>AdDecisionServer.FillRate</code>. Wenn die <code>AdDecisionServer.FillRate</code> niedrig ist, gibt Ihr ADS möglicherweise nicht genügend Werbung für die Avail-Dauer zurück.
<code>Avail.Impression</code>	Die Anzahl der Anzeigen mit Impressions-Tracking-Ereignissen, die beim MediaTailor serverseitigen Beaconsing beobachtet werden (nicht die Anzahl der Impressionen).
<code>Avail.ObservedDuration</code>	Die beobachtete Gesamtzahl der Millisekunden an Anzeigenzugriffen innerhalb des Zeitraums. CloudWatch <code>Avail.ObservedDuration</code> wird am Ende der Werbeanzeige ausgegeben und basiert auf der Dauer der Segmente, über die während der Verfügbarkeit der Anzeige im Manifest berichtet wurde.

Metrik	Description
<code>Avail.ObservedFilledDuration</code>	Die beobachtete Anzahl von Millisekunden der Werbeanzeige, die innerhalb des Zeitraums mit Werbeanzeigen MediaTailor gefüllt wurde. CloudWatch
<code>Avail.ObservedFillRate</code>	Der beobachtete einfache Durchschnitt der Raten, MediaTailor mit denen gefüllte einzelne Anzeigen innerhalb des Zeitraums geschaltet wurden. CloudWatch Wird nur für HLS-Manifeste beim ersten CUE-IN Tag ausgegeben. Wenn kein CUE-IN Tag vorhanden ist, wird diese Metrik MediaTailor nicht ausgegeben.
<code>Avail.ObservedSlateDuration</code>	Die beobachtete Gesamtzahl der Millisekunden von Schiefer, die innerhalb des Zeitraums eingefügt wurden. CloudWatch
<code>GetManifest.Age</code>	Das Gesamtalter des Manifests in Millisekunden. Gemessen von dem Zeitpunkt, zu dem der Ursprung das Manifest erstellt, bis zu dem Zeitpunkt, zu dem das personalisierte Manifest MediaTailor gesendet wird. Weitere Informationen zu Metriken zur Messung des Alters von Manifesten finden Sie unter Verwendung von Metriken zur Diagnose veralteter Manifeste.
<code>GetManifest.Errors</code>	Die Anzahl der Fehler, die MediaTailor beim Generieren von Manifesten in dem von Ihnen angegebenen CloudWatch Zeitraum aufgetreten sind.

Metrik	Description
GetManifest.Latency	Die MediaTailor Antwortzeit in Millisekunden für die Anforderung zur Generierung von Manifesten. Weitere Informationen zu Metriken zur Messung des Alters von Manifesten finden Sie unter. Verwendung von Metriken zur Diagnose veralteter Manifeste
GetManifest.MediaTailorAge	Die Zeitspanne in Millisekunden, MediaTailor in der das Manifest gespeichert wurde. Gemessen vom MediaTailor Empfang einer ursprünglichen Antwort bis zum Versand des personalisierten MediaTailor Manifests. Weitere Informationen zu Metriken zur Messung des Alters von Manifesten finden Sie unter Verwendung von Metriken zur Diagnose veralteter Manifeste.
Origin.Age	Die Zeitspanne in Millisekunden, für die der Ursprung das Manifest hat. Gemessen von dem Zeitpunkt, zu dem der Ursprung das Manifest erstellt, bis zu dem Zeitpunkt, an dem die ursprüngliche MediaTailor Anfrage gesendet wird. Alle <code>origin.*</code> Metriken werden für Anfragen ausgegeben, die direkt vom Ursprung aus erfüllt werden. Sie werden nicht für zwischeng gespeicherte Ausgangsantworten ausgegeben. Weitere Informationen zu Metriken zur Messung des Manifestalters finden Sie unter Verwendung von Metriken zur Diagnose veralteter Manifeste.

Metrik	Description
<code>Origin.Errors</code>	Die Anzahl der 200-Statuscode-Antworten ohne HTTP-Status und der Antworten mit Timeout, die in dem von Ihnen angegebenen CloudWatch Zeitraum vom Ursprungsserver MediaTailor empfangen wurden. Alle <code>origin.*</code> Messwerte werden für Anfragen ausgegeben, die direkt vom Ursprung aus erfüllt werden. Sie werden nicht für zwischeng espeicherte Ausgangsantworten ausgegeben.
<code>Origin.ManifestFileSizeBytes</code>	Die Dateigröße des Ursprungs-Manifests in Byte für HLS und DASH. In der Regel wird diese Metrik in Verbindung mit <code>Origin.ManifestFileSizeTooLarge</code> verwendet. Alle <code>origin.*</code> Metriken werden für Anfragen ausgegeben, die direkt vom Ursprung aus erfüllt werden. Sie werden nicht für zwischeng espeicherte Ausgangsantworten ausgegeben.
<code>Origin.ManifestFileSizeTooLarge</code>	Die Anzahl der Antworten vom Ursprung, deren Manifestgröße die konfigurierte Menge übersteigt. In der Regel wird diese Metrik in Verbindung mit <code>Origin.ManifestFileSizeBytes</code> verwendet. Alle <code>origin.*</code> Metriken werden für Anfragen ausgegeben, die direkt vom Ursprung aus erfüllt werden. Sie werden nicht für zwischeng espeicherte Ausgangsantworten ausgegeben.

Metrik	Description
<code>Origin.Timeouts</code>	Die Anzahl der Anfragen an den Ursprungs server, bei denen das Timeout in dem von Ihnen angegebenen CloudWatch Zeitraum überschritten wurde. Alle <code>origin.*</code> Metriken werden für Anfragen ausgegeben, die direkt vom Ursprung aus erfüllt werden. Sie werden nicht für zwischeng espeicherte Ausgangsantworten ausgegeben.
<code>Requests</code>	Die Anzahl gleichzeitiger Transaktionen pro Sekunde für alle Anforderungstypen. Die Anzahl der Transaktionen hängt hauptsäch lich von der Anzahl der Spieler ab und davon, wie oft die Spieler aktualisierte Manifeste anfordern. Jede Player-Anforderung zählt als eine Transaktion.
<code>SkippedReason.DurationExceeded</code>	Die Anzahl der Anzeigen, die nicht in eine Avail eingefügt wurden, weil die ADS eine Anzeigendauer zurückgab, die über der angegebenen Verfügbarkeitsdauer lag. Ein hoher Wert für diese Kennzahl kann zu einer Diskrepanz zwischen der Metrik und beitragen . <code>Avail.Ads</code> <code>AdDecisionServer.Ads</code> Weitere Informationen zu den Gründen, warum Werbung übersprungen wurde, finden Sie unter. Fehlerbehebung beim Überspringen von Anzeigen
<code>SkippedReason.EarlyCueIn</code>	Die Anzahl der Anzeigen, die aufgrund eines vorzeitigen Fehlers übersprungen wurden. CUE-IN
<code>SkippedReason.ImportError</code>	Die Anzahl der Anzeigen, die aufgrund eines Fehlers im Importjob übersprungen wurden.

Metrik	Description
<code>SkippedReason.ImportInProgress</code>	Die Anzahl der Anzeigen, die aufgrund eines vorhandenen aktiven Importauftrags übersprungen wurden.
<code>SkippedReason.InternalError</code>	Die Anzahl der Anzeigen, die aufgrund eines MediaTailor internen Fehlers übersprungen wurden.
<code>SkippedReason.NewCreative</code>	Die Anzahl der Anzeigen, die nicht in ein Avail eingefügt wurden, weil das Asset zum ersten Mal von einem Kunden angefordert wurde. Ein hoher Wert für diese Kennzahl kann vorübergehend zu einem niedrigen Gesamtwert führen <code>Avail.FillRate</code> , bis die Assets erfolgreich transkodiert werden können.
<code>SkippedReason.NoVariantMatch</code>	Die Anzahl der Anzeigen, die übersprungen wurden, weil es keine Variantenübereinstimmung zwischen der Anzeige und dem Inhalt gab.
<code>SkippedReason.PersonalizationThresholdExceeded</code>	Die Dauer von Anzeigen, die in dieser Konfiguration den für die Personalisierung festgelegten Schwellenwert überschreiten.
<code>SkippedReason.ProfileNotFound</code>	Die Anzahl der Anzeigen, die übersprungen wurden, weil das Transcodierungsprofil nicht gefunden wurde.
<code>SkippedReason.TranscodeError</code>	Die Anzahl der Anzeigen, die aufgrund eines Transcodierungsfehlers übersprungen wurden.

Metrik	Description
<code>SkippedReason.TranscodeInProgress</code>	Die Anzahl der Anzeigen, die nicht in einen Avail eingefügt wurden, weil die Anzeige noch nicht transkodiert wurde. Ein hoher Wert für diese Kennzahl kann vorübergehend zu einem niedrigen Gesamtwert führen <code>Avail.FillRate</code> , bis die Inhalte erfolgreich transkodiert werden können.
<code>GetAssets.Requests</code>	Die Anzahl der Asset List-Anfragen, die innerhalb des Zeitraums für HLS Interstitials-Sitzungen eingegangen sind. CloudWatch Verwenden Sie diese Kennzahl, um das Volumen verspäteter Bindungen und Entscheidungen zu überwachen und den Umfang der Nutzung von HLS Interstitials zu ermitteln.
<code>GetAssets.Latency</code>	Die Antwortzeit für Anfragen zur Inventarliste in Millisekunden für HLS Interstitials-Sitzungen. Überwachen Sie diese Kennzahl, um eine optimale Leistung bei der Entscheidungsfindung für Werbeanzeigen sicherzustellen und potenzielle Engpässe im Arbeitsablauf bei der späten Bindung zu identifizieren.

Note

Bei HLS Interstitials-Sitzungen verhalten sich einige Kennzahlen aufgrund der spätverbindlichen Natur der Anzeigenentscheidungen unterschiedlich:

- `Avail.ObservedFilledDurationTreffer`, `Avail.FilledDuration` da MediaTailor das tatsächliche Wiedergabeverhalten auf der Clientseite nicht beobachtet werden kann.
- `Avail.ObservedSlateDuration` meldet die geplante Slate-Dauer anhand der Antworten auf die Asset-Liste und nicht anhand der beobachteten Wiedergabe.

- Metriken mit dem Präfix „Observed“ liefern geschätzte Werte für HLS Interstitials-Sitzungen.

Einfache und gewichtete Durchschnittswerte

Sie können den einfachen Durchschnitt und den gewichteten Durchschnitt für die Antworten des ADS auf Anzeigenanfragen von MediaTailor und für die Anzahl der MediaTailor verfügbaren Anzeigen abrufen:

- Die einfachen Durchschnittswerte werden in `AdDecisionServer.FillRate` und `Avail.FillRate` angegeben. Dabei handelt es sich um die Durchschnittswerte der Füllraten in Prozent für einzelne Avails für den Zeitraum. Bei den einfachen Durchschnittswerten werden keine Unterschiede in der Dauer der einzelnen Avails berücksichtigt.
- Die gewichteten Durchschnittswerte beziehen sich auf die prozentuale Füllrate für die Summe aller Werte der Avail-Dauer. Diese Werte werden als $(\text{AdDecisionServer.Duration} * 100) / \text{Avail.Duration}$ und $(\text{Avail.FilledDuration} * 100) / \text{Avail.Duration}$ berechnet. Diese Durchschnittswerte reflektieren die Unterschiede in der Dauer der einzelnen Ad-Avails, sodass solche mit längerer Dauer stärker gewichtet werden.

Für einen Zeitraum mit nur einem einzigen Ad-Avail entspricht der mit `AdDecisionServer.FillRate` angegebene einfache Durchschnitt dem mit $(\text{AdDecisionServer.Duration} * 100) / \text{Avail.Duration}$ angegebenen gewichteten Durchschnitt. Der mit `Avail.FillRate` angegebene einfache Durchschnitt entspricht dem mit $(\text{Avail.FilledDuration} * 100) / \text{Avail.Duration}$ angegebenen gewichteten Durchschnitt.

Beispiel

Angenommen, die von Ihnen angegebene Zeitspanne verfügt über die folgenden zwei Ad-Avails:

- Die Dauer des ersten Ad-Avail beträgt 90 Sekunden:
 - Die ADS-Antwort für den Avail stellt 45 Sekunden an Werbung bereit (zu 50 % gefüllt).
 - MediaTailor füllt 45 Sekunden der verfügbaren Anzeigenzeit aus (50% gefüllt).
- Die Dauer des zweiten Ad-Avail beträgt 120 Sekunden:
 - Die ADS-Antwort für den Avail stellt 120 Sekunden an Werbung bereit (zu 100 % gefüllt).
 - MediaTailor füllt 90 Sekunden der verfügbaren Werbezeit aus (75% gefüllt).

Die Metriken sind wie folgt:

- Der Wert von `Avail.Duration` ist 120, nämlich die Summe der Dauer der beiden Ad-Avail-Zeiträume: $90 + 120$.
- `AdDecisionServer.Duration` ist 165, nämlich die Summe der beiden Werte der Antwortdauer: $45 + 120$.
- `Avail.FilledDuration` ist 135, nämlich die Summe der beiden Werte für die gefüllte Dauer: $45 + 90$.
- Der Wert von `AdDecisionServer.FillRate` ist 75 %, nämlich der gefüllte prozentuale Durchschnitt für jeden Avail: $(50 \% + 100 \%) / 2$. Dies ist der einfache Durchschnittswert.
- Der gewichtete Durchschnittswert für die ADS-Füllraten ist 78,57 %, was dem Wert für `AdDecisionServer.Duration` als Prozentsatz des Wertes für `Avail.Duration` entspricht: $(165 * 100) / 210$. Diese Berechnung berücksichtigt die Unterschiede bezüglich der Dauer.
- Der Wert von `Avail.FillRate` ist 62.5 %, nämlich der gefüllte prozentuale Durchschnitt für jeden Avail: $(50 \% + 75 \%) / 2$. Dies ist der einfache Durchschnittswert.
- Der gewichtete Durchschnitt der MediaTailor verfügbaren Füllraten liegt bei 64,29%, was einem Prozentsatz der folgenden `Avail.FilledDuration` Werte entspricht `Avail.Duration`: $(135 * 100) / 210$. Diese Berechnung berücksichtigt die Unterschiede bezüglich der Dauer.

Der höchste Wert `Avail.FillRate`, der für jede MediaTailor verfügbare Anzeige erreicht werden kann, liegt bei 100%. Der ADS gibt möglicherweise mehr Werbezeit zurück, als im Avail verfügbar ist, MediaTailor kann aber nur den verfügbaren Zeitraum füllen.

AWS Elemental MediaTailor CloudWatch Abmessungen

Sie können die AWS Elemental MediaTailor Daten anhand der folgenden Dimension filtern.

Dimension	Description
Configuration Name	Gibt die Konfiguration an, zu der die Metrik gehört.

Verwenden von Metriken zur Diagnose veralteter Manifeste von AWS Elemental MediaTailor

Ein veraltetes Manifest ist ein Manifest, das in letzter Zeit nicht aktualisiert wurde. Bei verschiedenen Workflows zum Einfügen von Anzeigen kann es aufgrund einer Vielzahl von Faktoren (z. B. Anforderungen an nachgeschaltete Systeme) unterschiedlich sein, wie lange es dauern muss, bis ein Manifest als veraltet angesehen wird. Sie können CloudWatch Amazon-Metriken verwenden, um Manifeste zu identifizieren, die die Veraltungstoleranz für Ihren Workflow überschreiten, und dabei helfen, herauszufinden, was die Ursache für die Verzögerungen bei der Aktualisierung von Manifesten sein könnte.

Die folgenden Kennzahlen helfen dabei, veraltete Manifeste und ihre Ursachen zu identifizieren.

Informationen zu allen ausgegebenen Metriken MediaTailor finden Sie unter [AWS Elemental MediaTailor CloudWatch Metriken](#)

Metrik	Definition	Verwenden Sie
GetManifest.Age	Misst das Gesamalter des Manifests, einschließlich GetManifest.MediaTailorAge sowohl als auch Origin.Age für diese Konfiguration.	Sie können diese Metrik verwenden, um Manifeste zu identifizieren, die Ihren Aktualisierungsschwellenwert überschritten haben und veraltet sind. Stellen Sie Alarme für diese Metrik ein, sodass Sie benachrichtigt werden, wenn veraltete Manifeste zugestellt werden. Informationen zu Alarmen finden Sie unter Alarming on metrics im CloudWatch Amazon-Benutzerhandbuch. Wenn Sie einen Alarm erhalten, verwenden Sie Origin.Age und, GetManifest.MediaTailorAge um festzustellen, ob MediaTailor oder der Ursprung für die Veralterung verantwortlich ist.
Origin.Age	Misst, wie lange der Ursprung über das Manifest verfügt, bevor es an	Wenn Sie veraltete Manifeste mit identifizierenGetManifest.Age,

Metrik	Definition	Verwenden Sie
	MediaTailor diese Konfiguration gesendet wird. Diese Metrik wird nicht ausgegeben, wenn die Antwort von einem Content Delivery Network (CDN) kommt. Die Antwort muss vom Ursprung kommen, damit <code>Origin.Age</code> sie gesendet werden kann.	können Sie die <code>Origin.Age</code> Metrik und die <code>GetManifest.MediaTailorAge</code> Metrik analysieren, um festzustellen, was zur manifesten Veralterung beiträgt. Wenn Sie feststellen, dass dies länger <code>Origin.Age</code> ist als Ihre typischen Verarbeitungszeiten am Ursprung, ist es wahrscheinlich, dass das Problem vom vorgelagerten System verursacht wird, und Sie sollten die Diagnose darauf konzentrieren.
<code>GetManifest.MediaTailorAge</code>	Misst, wie MediaTailor lange dieses Manifest für diese Konfiguration gespeichert wurde.	Wenn Sie veraltete Manifeste mit identifizieren <code>GetManifest.Age</code> , können Sie die <code>GetManifest.MediaTailorAge</code> Metrik und die <code>Origin.Age</code> Metrik analysieren, um festzustellen, was zur manifesten Veralterung beiträgt. Wenn das länger <code>GetManifest.MediaTailorAge</code> dauert als die typische Personalisierung von Manifesten, ist es wahrscheinlich MediaTailor, dass das Problem dadurch verursacht MediaTailor wird, und Sie sollten die Diagnose darauf konzentrieren. <code>GetManifest.Latency</code> kann außerdem ermitteln, wie lange es dauert, ein personalisiertes Manifest MediaTailor zu erstellen.

Metrik	Definition	Verwenden Sie
GetManifest.Latency	Misst die Zeit, die benötigt wird MediaTailor , um die Anfrage zu bearbeiten und ein personalisiertes Manifest für diese Konfiguration zu erstellen.	Wenn Sie vergleichen <code>Origin.Age</code> <code>GetManifest.MediaTailorAge</code> und feststellen, dass dies die Ursache für die verzögerte Manifestzustellung MediaTailor ist, können Sie die <code>GetManifest.Latency</code> Metrik analysieren, um festzustellen, ob der Prozess der Manifest-Personalisierung zur Veralterung des Manifests beiträgt. <code>GetManifest.MediaTailorAge</code> misst die Gesamtzeit, in der das Manifest gespeichert ist. MediaTailor <code>GetManifest.Latency</code> misst, wie viel von dieser Speicherzeit auf die MediaTailor Personalisierung des Manifests als Antwort auf eine Anfrage entfällt.

AWS Elemental MediaTailor API-Aufrufe aufzeichnen

AWS Elemental MediaTailor ist in einen Dienst integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die ein Benutzer, eine Rolle oder ein AWS Dienst in ausgeführt hat MediaTailor. CloudTrail erfasst alle API-Aufrufe MediaTailor als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der MediaTailor Konsole und Codeaufrufen für die MediaTailor API-Operationen. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignissen für MediaTailor. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole im Ereignisverlauf anzeigen. Anhand der von gesammelten Informationen können Sie die Anfrage ermitteln CloudTrail, an die die Anfrage gestellt wurde MediaTailor, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details.

Weitere Informationen CloudTrail dazu finden Sie im [AWS CloudTrail Benutzerhandbuch](#).

AWS Elemental MediaTailor Informationen in CloudTrail

CloudTrail ist in Ihrem AWS Konto aktiviert, wenn Sie das Konto erstellen. Wenn eine Aktivität in stattfindet AWS Elemental MediaTailor, wird diese Aktivität zusammen mit anderen CloudTrail AWS Serviceereignissen im Ereignisverlauf in einem Ereignis aufgezeichnet. Sie können aktuelle Ereignisse in Ihrem AWS Konto ansehen, suchen und herunterladen. Weitere Informationen finden Sie unter [Ereignisse mit CloudTrail Ereignisverlauf anzeigen](#).

Für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem AWS Konto, einschließlich der Ereignisse für AWS Elemental MediaTailor, erstellen Sie einen Trail. Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Wenn Sie einen Trail in der Konsole erstellen, gilt der Trail standardmäßig für alle AWS Regionen. Der Trail protokolliert Ereignisse aus allen Regionen der AWS Partition und übermittelt die Protokolldateien an den von Ihnen angegebenen Amazon S3 S3-Bucket. Darüber hinaus können Sie andere AWS Dienste konfigurieren, um die in den CloudTrail Protokollen gesammelten Ereignisdaten weiter zu analysieren und darauf zu reagieren. Weitere Informationen finden Sie hier:

- [Einen Trail für dein AWS Konto erstellen](#)
- [AWS Serviceintegrationen mit Protokollen CloudTrail](#)
- [Konfiguration von Amazon SNS SNS-Benachrichtigungen für CloudTrail](#)
- [Empfangen von CloudTrail Protokolldateien aus mehreren Regionen](#) und [Empfangen von CloudTrail Protokolldateien von mehreren Konten](#)

Alle AWS Elemental MediaTailor Aktionen werden von der [AWS Elemental MediaTailor API-Referenz](#) protokolliert CloudTrail und sind in dieser dokumentiert. Zum Beispiel werden durch Aufrufe der PutPlaybackConfiguration- und ListPlaybackConfigurations-Operationen Einträge in den CloudTrail-Protokolldateien generiert.

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

- Ob die Anfrage mit den Root-Benutzer- oder AWS Identity and Access Management (IAM-) Anmeldeinformationen gestellt wurde
- Ob die Anfrage mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen föderierten Benutzer ausgeführt wurde
- Ob die Anfrage von einem anderen AWS Dienst gestellt wurde

Weitere Informationen finden Sie unter [CloudTrail -Element userIdentity](#).

Grundlegendes zu AWS Elemental MediaTailor Einträgen in Protokolldateien

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket ermöglicht. CloudTrail Protokolldateien enthalten einen oder mehrere Protokolleinträge. Ein Ereignis ist eine einzelne Anforderung aus einer beliebigen Quelle und enthält Informationen zur angeforderten Aktion, zu Datum und Uhrzeit der Aktion, zu den Anforderungsparametern usw. CloudTrail -Protokolldateien stellen kein geordnetes Stack-Trace der öffentlichen API-Aufrufe dar. Daher werden sie nicht in einer bestimmten Reihenfolge angezeigt.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die PutPlaybackConfiguration Aktion demonstriert:

```
{
  "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/exemplename/"
    },
    "VideoContentSourceUrl": "http://examplevideo.com",
    "Name": "exemplename"
},
"requestID": "1a2b3c4d-1234-5678-1234-1a2b3c4d5e6f",
"eventID": "987abc65-1a2b-3c4d-5d6e-987abc654def",
"readOnly": false,
"eventType": "AwsApiCall",
"recipientAccountId": "111122223333"
}

```

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die `GetPlaybackConfiguration` Aktion demonstriert:

```

{
    "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": "exemplename"
    },
    "responseElements": null,

```

```

"requestID": "0z1y2x3w-0123-4567-9876-6q7r8s9t0u1v",
"eventID": "888ddd77-3322-eeww-uuii-abc123jkl343",
"readOnly": true,
"eventType": "AwsApiCall",
"recipientAccountId": "111122223333"
}

```

Empfangen von Benachrichtigungen zur AWS Elemental MediaTailor Kanalmontage

MediaTailor erstellt Warnmeldungen für Probleme oder potenzielle Probleme, die mit Ihren Ressourcen für die Kanalmontage auftreten. Die Warnung beschreibt das Problem, wann das Problem aufgetreten ist und welche Ressourcen betroffen sind.

Sie können die Warnungen in der AWS-Managementkonsole, AWS Command Line Interface (AWS CLI) oder programmgesteuert mithilfe der MediaTailor [ListAlerts](#) API anzeigen. AWS SDKs

Important

Benachrichtigungen sind nur für Channel-Assembly-Ressourcen verfügbar, die am oder nach dem 14. Juli 2021 erstellt wurden.

Benachrichtigungen zur Kanalmontage

Art der Warnung	Warnungsc ode	Warnmeldung	Hinweise
VOD-Quelle	NOT_PROCESSED	MediaTailor hat die Paketkonfiguration <i>configurationPath</i> nicht verarbeitet.	
	UNREACHABLE	Wir können die URL nicht erreichen <i>url</i> .	
	UNAUTHORIZED	<i>url</i> hat die Anfrage nicht autorisiert.	

Art der Warnung	Warnungscode	Warnmeldung	Hinweise
	TIMEOUT	Bei der Verbindung mit <i>url</i> trat eine Zeitüberschreitung auf.	
	UNPARSABLE_MANIFEST	MediaTailor ist beim Analysieren des Manifests von auf ein Problem gestoßen. <i>url</i>	
	VARIANT_DURATION_MISMATCH	MediaTailor ist beim Parsen des Manifests von auf Varianten gestoßen, deren Gesamtdauer nicht übereinstimmt. <i>url</i> Dies kann zu Verzögerungen bei der Wiedergabe führen.	Ihr Manifest hat je nach Varianten/Repräsentationen unterschiedliche Dauern. Dies kann dazu führen, dass Bildunterschriften fehlen oder falsch sind und Anzeigen nicht eingefügt MediaTailor werden können.
	SEGMENT_DURATION_TOO_LONG	MediaTailor ist beim Analysieren des Manifests von auf ein Segment mit einer Dauer von mehr als dreißig Sekunden gestoßen. <i>url</i> Dies kann zu Verzögerungen bei der Wiedergabe, fehlenden oder falschen Untertiteln und der Unfähigkeit, Werbung einzufügen, führen.	Ihr Manifest enthält ein Segment, das länger als 30 Sekunden ist.

Art der Warnung	Warnungscode	Warnmeldung	Hinweise
	TARGET_DURATION_MISMATCH	MediaTailor ist beim Parsen des Manifests von auf eine Nichtübereinstimmung der EXT-X-TARGETDURATION Werte in den HLS-Manifesten gestoßen. <i>url</i> Dies kann zu Verzögerungen bei der Wiedergabe führen.	Die Zieldauer stimmt nicht für alle Manifeste in der Quelle überein.
Standort der Quelle	NOT_PROCESSED	MediaTailor hat die Ressource nicht verarbeitet <i>resourceName</i> .	
Programm	VOD_SOURCE_ALERT	Die VOD-Quelle <i>vodSourceName</i> in diesem Programm hat die folgende Warnung: <i>vodSourceAlertCode</i> <i>vodSourceAlertMessage</i>	
	SOURCE_LOCATION_ALERT	Für den <i>sourceLocationName</i> Quellpfad in diesem Programm wurde die folgende Warnung angezeigt: <i>sourceLocationAlertCode</i> <i>sourceLocationAlertMessage</i>	

Art der Warnung	Warnungscode	Warnmeldung	Hinweise
	CODEC_MISMATCH	MediaTailor Im <i>channelName</i> Zeitplan ist ein nicht übereinstimmender Codec aufgetreten. Die Nichtübereinstimmung liegt <i>sourceGroupName</i> zwischen dem Manifest <i>manifestUrl</i> und <i>programName1</i> <i>programName2</i> dem Manifest. <i>manifestUrl</i>	
	RESOLUTION_MISMATCH	MediaTailor ist auf eine nicht übereinstimmende Lösung im <i>channelName</i> Zeitplan gestoßen. Die Nichtübereinstimmung liegt <i>sourceGroupName</i> zwischen <i>programName1</i> dem Manifest <i>manifestUrl</i> und <i>programName2</i> dem Manifest. <i>manifestUrl</i>	
	BANDWIDTH_MISMATCH	MediaTailor ist im Zeitplan auf eine nicht übereinstimmende Bandbreite gestoßen. <i>channelName</i> Die Nichtübereinstimmung liegt <i>sourceGroupName</i> zwischen <i>programName1</i> dem Manifest <i>manifestUrl</i> und <i>programName2</i> dem Manifest. <i>manifestUrl</i>	

Art der Warnung	Warnungscode	Warnmeldung	Hinweise
	FRAMERATE_MISMATCH	MediaTailor ist im Zeitplan auf eine nicht übereinstimmende Framerate gestoßen. <i>channelName</i> Die Nichtübereinstimmung liegt <i>sourceGroup</i> zwischen dem <i>programName1</i> Manifest und dem Manifest <i>manifestUrl</i> . <i>programName2</i> <i>manifestUrl</i>	
	TARGET_DURATION_MISMATCH	MediaTailor Es wurden nicht übereinstimmende EXT-X-TARGETDURATION Werte für alle HLS-Manifestdateien im Zeitplan festgestellt. <i>channelName</i> Die Nichtübereinstimmung liegt <i>sourceGroup</i> zwischen dem Manifest <i>manifestUrl</i> und <i>programName1</i> <i>programName2</i> dem Manifest. <i>manifestUrl</i>	

Art der Warnung	Warnungscode	Warnmeldung	Hinweise
	SEGMENT_DURATION_MISMATCH	MediaTailor Es wurde festgestellt, dass die Werte für die Segmentdauer der einzelnen Manifeste im <i>channelName</i> Zeitplan nicht übereinstimmten. Die Nichtübereinstimmung liegt <i>sourceGroupName</i> zwischen <i>programName1</i> dem Manifest <i>manifestUrl</i> und <i>programName2</i> dem Manifest. <i>manifestUrl</i>	
	NO_COMMON_SEGMENT_BOUNDARY_FOR_AD_SLATE	MediaTailor Ich war nicht in der Lage, ein Ad Slate am Offset <i>offsetMillis</i> für das Programm <i>programName</i> einzufügen. Bei der Startzeit des Ad Slate gibt es keine gemeinsame Segmentgrenze.	
	NOT_PROCESSED	MediaTailor hat die Ressource nicht verarbeitet <i>resourceName</i> .	
	TOO_MANY_ALERTS	MediaTailor hat zu viele Alerts gefunden und gibt keine weiteren Alerts für <i>programName</i> . Löschen Sie bestehende Benachrichtigungen , für die Sie weiterhin Benachrichtigungen erhalten möchten <i>programName</i> .	

Art der Warnung	Warnungsc ode	Warnmeldung	Hinweise
Kanal	PROGRAM_A LERT	Das in diesem Kanal <i>programName</i> enthaltene Programm hat die folgende Warnung <i>programAl ertCode : programAl ertMessage</i>	

Behebung von Warnmeldungen im Zusammenhang mit Manifesten

Wenn Sie auf manifestbezogene Warnmeldungen wie UNPARSABLE_MANIFEST, oder stoßen VARIANT_DURATION_MISMATCHTARGET_DURATION_MISMATCH, überprüfen Sie Folgendes:

1. Überprüfen Sie die Quellenforderungen: Stellen Sie sicher, dass Ihr Inhalt die unter beschriebenen Anforderungen erfüllt. [the section called “Integration einer Inhaltsquelle”](#)
2. Überprüfen Sie die Manifestprotokolle: Eine detaillierte Analyse von Manifestfehlern und Ereignistypen finden Sie unter [the section called “Manifest-Protokolle”](#).
3. Überprüfen Sie die Fehlercodes: Eine detaillierte Fehleranalyse finden Sie unter [Fehlerbehebung](#).
4. Überprüfen Sie das Manifestformat: Stellen Sie sicher, dass Ihr Manifest den korrekten Formatspezifikationen für Ihr Streaming-Protokoll (HLS oder DASH) entspricht.

Benachrichtigungen anzeigen

Sie können Benachrichtigungen für jede Ressource zur MediaTailor Kanalmontage anzeigen. Wenn Sie sich die Benachrichtigungen für Kanäle und Programme ansehen, werden MediaTailor auch alle zugehörigen Ressourcen angezeigt, die im Kanal oder Programm enthalten sind. Wenn Sie sich beispielsweise die Benachrichtigungen für ein bestimmtes Programm ansehen, werden Ihnen auch Benachrichtigungen für den Quellstandort und die VOD-Quellen angezeigt, die das Programm enthält.

Gehen Sie wie folgt vor, um Benachrichtigungen anzuzeigen.

Console

Um Benachrichtigungen in der Konsole anzuzeigen

1. Öffnen Sie die MediaTailor Konsole unter <https://console.aws.amazon.com/mediatailor/>.
2. Wählen Sie die Ressource aus, für die Sie Benachrichtigungen anzeigen möchten.
3. Wählen Sie die Registerkarte „Benachrichtigungen“, um die Benachrichtigungen anzuzeigen.

AWS Command Line Interface (AWS CLI)

Um Benachrichtigungen für eine Channel-Assembly-Ressource aufzulisten, benötigen Sie den [Amazon-Ressourcennamen \(ARN\)](#) der Ressource. Sie können den `describe-resource-type` Befehl in der AWS Command Line Interface (AWS CLI) verwenden, um den ARN der Ressource abzurufen. Führen Sie beispielsweise den Befehl [describe-channel](#) aus, um den ARN eines bestimmten Kanals abzurufen:

```
aws mediatailor describe-channel --channel-name MyChannelName
```

Verwenden Sie dann den Befehl [aws mediatailor list-alerts](#), um die mit der Ressource verknüpften [Warnungen](#) aufzulisten:

```
aws mediatailor list-alerts --resource-arn arn:aws:mediatailor:region:aws-account-id:resource-type/resource-name
```

API

Um Benachrichtigungen für eine Channel-Assembly-Ressource aufzulisten, benötigen Sie den [Amazon-Ressourcennamen \(ARN\)](#) der Ressource. Sie können den `DescribeResource` Vorgang in der MediaTailor API verwenden, um den ARN der Ressource abzurufen. Verwenden Sie den [DescribeChannel](#) Vorgang beispielsweise, um den ARN eines bestimmten Kanals abzurufen.

Verwenden Sie dann die [ListAlerts](#) API, um die Benachrichtigungen für die Ressource aufzulisten.

Umgang mit Warnmeldungen

Wenn eine Warnung auftritt, sehen Sie sich die Warnungen in der AWS-Managementkonsole Alerts API an oder verwenden Sie die AWS Command Line Interface (AWS CLI) AWS SDKs, oder die MediaTailor Alerts API, um die möglichen Ursachen des Problems zu ermitteln.

MediaTailor Löscht die Warnung, nachdem Sie das Problem behoben haben.

Ressourcen taggen AWS Elemental MediaTailor

Ein Tag ist eine Metadaten-Bezeichnung, die Sie einer Ressource zuweisen oder die sie einer AWS AWS Ressource zuweist. Jedes Tag besteht aus einem Schlüssel und einem Wert. Für Tags, die Sie zuweisen, definieren Sie einen Schlüssel und einen Wert. So können Sie beispielsweise den Schlüssel als `stage` und den Wert für eine Ressource als `test` definieren.

Tags sind für folgende Aktivitäten nützlich:

- Identifizieren und organisieren Sie Ihre AWS Ressourcen. Viele AWS Dienste unterstützen Tagging, sodass Sie Ressourcen aus verschiedenen Diensten dasselbe Tag zuweisen können, um anzuzeigen, dass die Ressourcen miteinander verknüpft sind. Sie könnten beispielsweise einem AWS Elemental MediaPackage Kanal und Endpunkt dasselbe Tag zuweisen, das Sie einer AWS Elemental MediaTailor Konfiguration zuweisen.
- Verfolgen Sie Ihre AWS Kosten. Sie aktivieren diese Tags auf dem AWS Fakturierung und Kostenmanagement Dashboard. AWS verwendet die Tags, um Ihre Kosten zu kategorisieren und Ihnen einen monatlichen Kostenverteilungsbericht zu senden. Weitere Informationen finden Sie unter [Use cost allocation tags](#) (Verwendung von Kostenzuordnungs-Tags) im [AWS Billing - Benutzerhandbuch](#).
- Kontrollieren Sie den Zugriff auf Ihre AWS Ressourcen. Weitere Informationen finden Sie unter [Controlling access using tags](#) (Zugriffssteuerung mit Tags) im [IAM-Benutzerhandbuch](#).

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

Unterstützte Ressourcen in AWS Elemental MediaTailor

Die folgenden Ressourcen in AWS Elemental MediaTailor unterstützen Tagging:

- Kanäle

- Konfigurationen
- SourceLocations
- VodSources

Tag-Einschränkungen

Die folgenden grundlegenden Einschränkungen gelten für Tags auf AWS Elemental MediaTailor Ressourcen:

- Die maximale Anzahl der Tags, die Sie einer Ressource zuweisen können – 50.
- Maximale Schlüssellänge – 128 Unicode-Zeichen.
- Maximale Wertlänge – 256 Unicode-Zeichen.
- Valid characters for key and value – a-z, A-Z, 0-9, space, and the following characters: _ . : / = + - and @
- Bei Schlüsseln und Werten wird die Groß-/Kleinschreibung berücksichtigt
- Verwenden Sie nicht `aws :` als Präfix für den Schlüssel. Dieses Präfix ist für AWS reserviert.

Verwaltung von Tags in AWS Elemental MediaTailor

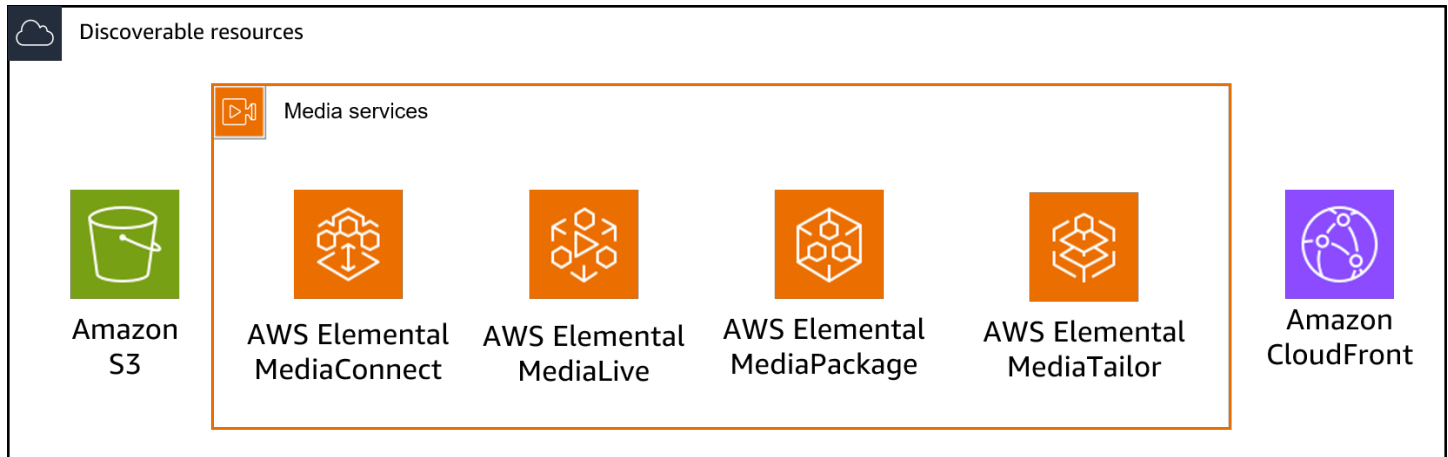
Sie legen Tags als Eigenschaften für eine Ressource fest. Sie können Tags über die AWS Elemental MediaTailor API oder die AWS Command Line Interface (AWS CLI) hinzufügen, bearbeiten und löschen. Weitere Informationen finden Sie in der [AWS Elemental MediaTailor -API-Referenz](#).

Überwachung von AWS Mediendiensten mit Workflow-Monitor

Der Workflow-Monitor ist ein Tool zur Erkennung, Visualisierung und Überwachung von AWS Medien-Workflows. Der Workflow-Monitor ist in der AWS Konsole und in der API verfügbar. Sie können den Workflow-Monitor verwenden, um visuelle Zuordnungen der Ressourcen Ihres Workflows, sogenannte Signalzuordnungen, zu ermitteln und zu erstellen. Sie können CloudWatch Amazon-Alarm- und EventBridge Amazon-Regelvorlagen erstellen und verwalten, um die zugewiesenen Ressourcen zu überwachen. Die von Ihnen erstellten Überwachungsvorlagen werden in bereitstellbare AWS CloudFormation Vorlagen umgewandelt, um Wiederholbarkeit zu gewährleisten. AWS empfohlene Alarmvorlagen bieten vordefinierte Best-Practice-Überwachung.

Entdecken

Nutzen Sie Signalkarten, um automatisch miteinander verbundene AWS Ressourcen zu erkennen, die mit Ihrem Medien-Workflow verknüpft sind. Die Erkennung kann bei jeder unterstützten Serviceresource beginnen und erstellt eine end-to-end Zuordnung des Workflows. Signalkarten können als eigenständige Visualisierungstools verwendet oder mit Überwachungsvorlagen erweitert werden.



Monitor

Sie können benutzerdefinierte CloudWatch Alarm- und EventBridge Regelvorlagen erstellen, um den Zustand und den Status Ihrer Medien-Workflows zu überwachen. Es stehen bewährte Alarmvorlagen für den Import in Ihre Workflow-Monitor-Umgebung zur Verfügung. Sie können die Best-Practice-Alarmvorlagen unverändert verwenden oder sie so bearbeiten, dass sie besser zu Ihrem Arbeitsablauf passen. Alle Vorlagen, die Sie erstellen, werden in CloudFormation Vorlagen für eine wiederholbare Bereitstellung umgewandelt.



Monitoring and deployment services



Amazon CloudWatch



Alarms



Logs



Amazon EventBridge



Rules



AWS CloudFormation



Stacks

Note

Für die Verwendung des Workflow-Monitors fallen keine direkten Kosten an. Mit den Ressourcen, die zur Überwachung Ihres Workflows erstellt und verwendet werden, sind jedoch Kosten verbunden.

Wenn die Überwachung bereitgestellt wird, werden Amazon CloudWatch - und EventBridge Amazon-Ressourcen erstellt. Wenn Sie die AWS Managementkonsole verwenden, werden Sie vor der Bereitstellung der Überwachung auf einer Signalkarte darüber informiert, wie viele Ressourcen erstellt werden. Weitere Informationen zur Preisgestaltung finden Sie unter: [CloudWatchPreise](#) und [EventBridge Preisgestaltung](#).

Der Workflow-Monitor verwendet AWS CloudFormation Vorlagen für die Bereitstellung der CloudWatch EventBridge Ressourcen. Diese Vorlagen werden in einem Amazon Simple Storage Service-Bucket der Standardklasse gespeichert, der während des Bereitstellungsprozesses von Workflow-Monitor in Ihrem Namen erstellt wird und für den Objektspeicher und Rückruf Gebühren anfallen. Weitere Informationen zur Preisgestaltung finden Sie unter: [Amazon S3 S3-Preise](#).

In der Workflow-Monitor-Signalkarte für AWS Elemental MediaPackage Kanäle generierte Vorschauen werden vom MediaPackage Origin-Endpunkt aus bereitgestellt und es fallen Gebühren für ausgehende Datenübertragungen an. [Die Preise findest du unter: MediaPackage Preise](#).

Komponenten des Workflow-Monitors

Der Workflow-Monitor besteht aus vier Hauptkomponenten:

- CloudWatch Alarmvorlagen — Definieren Sie die Bedingungen, anhand derer Sie die Überwachung durchführen möchten CloudWatch. Sie können Ihre eigenen Alarmvorlagen erstellen oder vordefinierte Vorlagen importieren, die von erstellt wurden AWS. Weitere Informationen finden Sie unter: [CloudWatch Alarmgruppen und Vorlagen zur Überwachung Ihres AWS Medienworkflows](#)
- EventBridge Regelvorlagen — Definieren Sie, wie Benachrichtigungen EventBridge gesendet werden, wenn ein Alarm ausgelöst wird. Weitere Informationen finden Sie unter: [EventBridge Regelgruppen und Vorlagen zur Überwachung Ihres AWS Medien-Workflows](#)
- Signalzuordnungen — Verwenden Sie einen automatisierten Prozess, um anhand vorhandener AWS Ressourcen AWS elementare Workflow-Übersichten zu erstellen. Die Signalzuordnungen können verwendet werden, um Ressourcen in Ihrem Workflow zu erkennen und diese Ressourcen zu überwachen. Weitere Informationen finden Sie unter: [Signalkarten des Workflow-Monitors](#)
- Überblick — Auf der Übersichtsseite können Sie den Status mehrerer Signalkarten direkt von einem Standort aus überwachen. Überprüfen Sie Metriken, Protokolle und Alarmer für Ihre Workflows. Weitere Informationen finden Sie unter: [Überblick über den Workflow-Monitor](#)

Unterstützte Services

Der Workflow-Monitor unterstützt die automatische Erkennung und Signalzuweisung von Ressourcen, die mit den folgenden Diensten verknüpft sind:

- AWS Elemental MediaConnect
- AWS Elemental MediaLive
- AWS Elemental MediaPackage
- AWS Elemental MediaTailor
- Amazon S3
- Amazon CloudFront

Topics

- [Workflow-Monitor zur Überwachung von AWS Mediendiensten konfigurieren](#)
- [Verwenden Sie den Workflow-Monitor](#)

Workflow-Monitor zur Überwachung von AWS Mediendiensten konfigurieren

Um den Workflow-Monitor zum ersten Mal einzurichten, erstellen Sie die Alarm- und Ereignisvorlagen und entdecken Signalkarten, die zur Überwachung Ihrer Medien-Workflows verwendet werden. Die folgende Anleitung enthält die erforderlichen Schritte, um IAM-Rollen auf Administrator- und Operator-Ebene einzurichten, Workflow-Monitor-Ressourcen zu erstellen und die Überwachung für Ihre Workflows bereitzustellen.

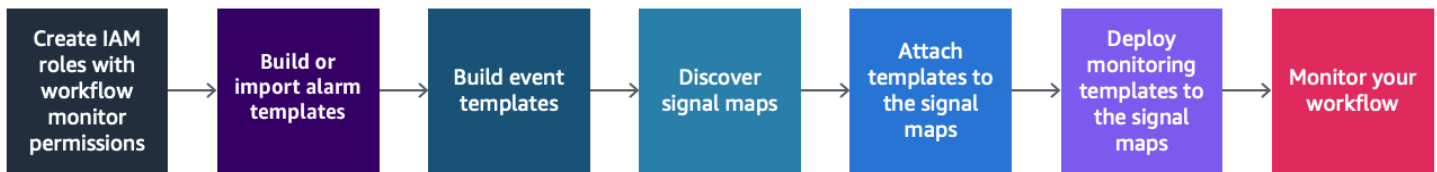
Topics

- [Erste Schritte mit dem Workflow-Monitor](#)
- [Workflow-Monitor-Gruppen und -Vorlagen](#)
- [Signalkarten des Workflow-Monitors](#)
- [Workflow-Monitor-Kontingente](#)

Erste Schritte mit dem Workflow-Monitor

Die folgenden Schritte bieten einen grundlegenden Überblick über die erstmalige Verwendung von Workflow Monitor.

1. Richten Sie Workflow-Monitor-IAM-Berechtigungen für Rollen auf Administrator- und Operatorebene ein: [IAM-Richtlinien zur Workflow-Überwachung](#)
2. Erstellen Sie Alarmvorlagen oder importieren Sie vordefinierte Vorlagen, die erstellt wurden von AWS: [CloudWatch-Alarme](#)
3. Erstellen Sie Benachrichtigungsereignisse, die zugestellt werden von EventBridge: [EventBridge Regeln](#)
4. Entdecken Sie Signalkarten mithilfe Ihrer vorhandenen AWS Elementarressourcen: [Signalkarten](#)
5. Hängen Sie die Alarmvorlagen und Benachrichtigungsregeln an Ihre Signalkarte an: [Vorlagen anhängen](#)
6. Stellen Sie die Vorlagen bereit, um mit der Überwachung der Signalkarte zu beginnen: [Bereitstellung von Vorlagen für die Überwachung](#)
7. Überwachen und überprüfen Sie Ihre Workflow-Monitor-Ressourcen mithilfe des Übersichtsbereichs der AWS Konsole: [-Übersicht](#)



Workflow-Monitor (IAM-Richtlinien)

Der Workflow-Monitor interagiert mit mehreren AWS Diensten, um Signalkarten, Builds CloudWatch und EventBridge Ressourcen sowie Vorlagen zu erstellen. CloudFormation Da der Workflow-Monitor mit einer Vielzahl von Diensten interagiert, müssen für diese Dienste spezifische AWS Identity and Access Management (IAM-) Richtlinien zugewiesen werden. In den folgenden Beispielen werden die erforderlichen IAM-Richtlinien sowohl für Administrator- als auch für Operator-IAM-Rollen beschrieben.

IAM-Richtlinie für Administratoren

Die folgende Beispielrichtlinie bezieht sich auf eine IAM-Richtlinie für den Workflow-Monitor auf Administratorebene. Diese Rolle ermöglicht die Erstellung und Verwaltung von Workflow-Monitor-Ressourcen und der unterstützten Servicere Ressourcen, die mit Workflow-Monitor interagieren.

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": "*"
  }
]
}

```

IAM-Richtlinie für Betreiber

Die folgende Beispielrichtlinie bezieht sich auf eine IAM-Richtlinie für den Workflow-Monitor auf Operatorebene. Diese Rolle ermöglicht eingeschränkten und schreibgeschützten Zugriff auf die Workflow-Monitor-Ressourcen und die unterstützten Serviceressourcen, die mit dem Workflow-Monitor interagieren.

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": "*"
}
]
}

```

Workflow-Monitor-Gruppen und -Vorlagen

Bevor Sie die Workflow-Überwachung in einer Signalmap einsetzen können, müssen Sie die Gruppen und Vorlagen für CloudWatch Alarme und EventBridge Benachrichtigungen erstellen. Die CloudWatch Vorlagen definieren, welche Szenarien und Schwellenwerte verwendet werden, um die Alarme auszulösen. Die EventBridge Vorlagen bestimmen, wie Ihnen diese Alarme gemeldet werden.

Wenn Sie nur Zuordnungen Ihrer verbundenen Ressourcen wünschen und nicht die Funktionen der Monitoring-Vorlagen von Workflow-Monitor verwenden möchten, können Sie Signalzuordnungen auch ohne CloudWatch Vorlagen verwenden. EventBridge Weitere Informationen zur Verwendung von Signalkarten finden Sie unter: [Signalkarten](#)

Topics

- [CloudWatch Alarmgruppen und Vorlagen zur Überwachung Ihres AWS Medienworkflows](#)
- [EventBridge Regelgruppen und Vorlagen zur Überwachung Ihres AWS Medien-Workflows](#)

CloudWatch Alarmgruppen und Vorlagen zur Überwachung Ihres AWS Medienworkflows

Mit Workflow-Monitor-Alarmen können Sie vorhandene CloudWatch Messwerte als Grundlage für Alarme für Ihre Signalkarten verwenden. Sie können eine Alarmvorlagengruppe erstellen, um die Alarmtypen zu sortieren und zu klassifizieren, die für Ihren Arbeitsablauf wichtig sind. Innerhalb jeder Alarmvorlagengruppe erstellen Sie Alarmvorlagen mit bestimmten CloudWatch Metriken und Parametern, die Sie überwachen möchten. Sie können Ihre eigenen Alarmvorlagen erstellen oder empfohlene Alarmvorlagen importieren, die von erstellt wurden AWS. Nachdem Sie eine Alarmvorlagengruppe und Alarmvorlagen innerhalb dieser Gruppe erstellt haben, können Sie eine oder mehrere dieser Alarmvorlagengruppen an eine Signalmap anhängen.

Sie müssen zuerst eine Alarmvorlagengruppe erstellen. Nachdem Sie eine Alarmvorlagengruppe erstellt haben, können Sie Ihre eigenen Vorlagen erstellen oder empfohlene Vorlagen verwenden, die von erstellt wurden AWS. Wenn Sie Ihre eigenen Alarmvorlagen erstellen möchten, fahren Sie auf dieser Seite fort. Weitere Informationen zum Importieren von empfohlenen Vorlagen finden Sie unter:

[Empfohlene Vorlagen](#)

In diesem Abschnitt wird die Erstellung von CloudWatch Alarmen mithilfe des Workflow-Monitors behandelt. Weitere Informationen darüber, wie der CloudWatch Service mit Alarmen umgeht, und Einzelheiten zu den Alarmkomponenten finden Sie unter: [Verwenden von CloudWatch Alarmen](#) im CloudWatch Amazon-Benutzerhandbuch

Alarm-Vorlagengruppen erstellen

Mithilfe von Alarmvorlagengruppen können Sie die Alarmtypen sortieren und klassifizieren, die für Ihren Arbeitsablauf wichtig sind.

Um eine Alarmvorlagengruppe zu erstellen

1. Wählen Sie im Navigationsbereich der Workflow-Monitor-Konsole die Option CloudWatchAlarmvorlagen aus.
2. Wählen Sie Alarmvorlagengruppe erstellen aus.
3. Geben Sie der Alarmvorlagengruppe einen eindeutigen Gruppennamen und optional eine Beschreibung.
4. Wählen Sie Erstellen aus. Sie werden zur Detailseite der neu erstellten Alarmvorlagengruppe weitergeleitet.

Alarmvorlagen erstellen

Sie können Alarmvorlagen mit den CloudWatch Metriken und Parametern erstellen, die Sie überwachen möchten.

Um eine Alarmvorlage zu erstellen

1. Wählen Sie auf der Detailseite der Alarmvorlagengruppe die Option Alarmvorlage erstellen aus.
2. Geben Sie der Alarmvorlage einen eindeutigen Vorlagennamen und optional eine Beschreibung.
3. Gehen Sie im Abschnitt Metrik auswählen wie folgt vor:
 1. Wählen Sie einen Zielressourcentyp aus. Der Zielressourcentyp ist eine Ressource für den jeweiligen Service, z. B. ein Kanal für MediaLive und MediaPackage oder ein Flow für MediaConnect.
 2. Wählen Sie einen Metriknamen aus. Dies ist die CloudWatch Metrik, die als Grundlage für den Alarm dient. Die Liste der Metriken ändert sich je nach ausgewähltem Zielressourcentyp.
4. Gehen Sie im Bereich Alarmeinstellungen wie folgt vor:

Note

Weitere Informationen darüber, wie der CloudWatch Service mit Alarmen umgeht, und Einzelheiten zu den Alarmkomponenten finden Sie unter: [Verwenden von CloudWatch Alarmen](#) im CloudWatch Amazon-Benutzerhandbuch

1. Wählen Sie die Statistik aus. Dies ist ein Wert, z. B. eine Summe oder ein Durchschnitt, der zur Überwachung der Metrik verwendet wird.
2. Wählen Sie den Vergleichsoperator aus. Dieses Feld verweist auf den Schwellenwert, den Sie im nächsten Schritt festgelegt haben.
3. Legen Sie einen Schwellenwert fest. Dies ist ein numerischer Wert, anhand dessen der Vergleichsoperator bestimmt, ob der Status größer als, kleiner oder gleich dem Status ist.
4. Legen Sie einen Zeitraum fest. Dies ist ein Zeitwert in Sekunden. Der Zeitraum ist der Zeitraum, in dem Statistik, Vergleichsoperator und Schwellenwert interagieren, um zu bestimmen, ob der Alarm ausgelöst wird.
5. Legen Sie die Datenpunkte fest. Dieser Wert bestimmt, wie viele Datenpunkte benötigt werden, um den Alarm auszulösen.

6. Wählen Sie aus, wie fehlende Daten behandelt werden sollen. Diese Auswahl bestimmt, wie dieser Alarm auf fehlende Daten reagiert.
5. Wählen Sie Erstellen aus, um den Vorgang abzuschließen.

Ein Beispiel für eine fertige Alarmvorlage könnte die folgenden Parameter haben: Ein MediaConnect Flow-Zielressourcentyp wird auf den Metriknamen für Verbindungsabbrüche überwacht. Der Statistikwert ist auf Summe mit einem Vergleichsoperator von „größer als oder gleich“ und einem Schwellenwert von 10 festgelegt. Der Zeitraum ist auf 60 Sekunden festgelegt und erfordert nur 1 von 1 Datenpunkten. Fehlende Daten behandeln ist auf „Ignorieren“ gesetzt.

Das Ergebnis dieser Einstellungen ist: Der Workflow-Monitor überwacht, ob es im Flow zu Verbindungsabbrüchen kommt. Wenn innerhalb von 60 Sekunden 10 oder mehr Verbindungsabbrüche auftreten, wird der Alarm ausgelöst. 10 oder mehr Verbindungsabbrüche innerhalb von 60 Sekunden müssen nur einmal geschehen, damit der Alarm ausgelöst wird.

Empfohlene Alarmvorlagen für die Überwachung Ihres Medienworkflows AWS

Bei den empfohlenen Vorlagen von Workflow Monitor handelt es sich um eine kuratierte Auswahl von AWS Elemental-Servicemetriken mit vordefinierten Alarmeinstellungen, die für die Metrik geeignet sind. Wenn Sie keine benutzerdefinierten Alarmvorlagen erstellen möchten, bieten Ihnen die empfohlenen Vorlagen Best-Practice-Überwachungsvorlagen, die von erstellt wurden. AWS

Der Workflow-Monitor enthält empfohlene Vorlagengruppen für jeden unterstützten Dienst. Diese Gruppen sind so konzipiert, dass sie die Überwachung nach bewährten Methoden auf bestimmte Arten von Workflows anwenden. Jede Vorlagengruppe enthält eine kuratierte Auswahl von Alarmen, die anhand dienstspezifischer Metriken konfiguriert wurden. Beispielsweise hat eine empfohlene Vorlagengruppe für einen MediaLive Multiplex-Workflow einen anderen Satz vorkonfigurierter Metriken als ein CDI-Workflow. MediaConnect

Um empfohlene Alarmvorlagen zu verwenden

1. Folgen Sie den Schritten, um [eine Alarmvorlagengruppe zu erstellen](#), oder wählen Sie eine bestehende aus.
2. Wählen Sie im Abschnitt Alarmvorlagen die Option Importieren aus. Sie müssen die AWS empfohlenen Vorlagen in Ihre Vorlagengruppe importieren.
3. Verwenden Sie das Drop-down-Menü für CloudWatch Alarmvorlagengruppen, um eine AWS empfohlene Gruppe auszuwählen. Diese Gruppen enthalten kuratierte Alarme für bestimmte Dienste.

4. Wählen Sie mithilfe der Kontrollkästchen die Vorlagen aus, die importiert werden sollen. Jede Vorlage listet ihre Metriken und vorkonfigurierten Überwachungswerte auf und bietet eine Beschreibung der Metrik. Wenn Sie mit der Auswahl der Vorlagen fertig sind, klicken Sie auf die Schaltfläche Hinzufügen.
5. Die ausgewählten Vorlagen werden in den Bereich „Zu importierende Alarmvorlage (n)“ verschoben. Überprüfen Sie Ihre Auswahl und wählen Sie Importieren aus.
6. Nach Abschluss des Imports werden die ausgewählten Vorlagen der Vorlagengruppe hinzugefügt. Wenn Sie weitere Vorlagen hinzufügen möchten, wiederholen Sie den Importvorgang.
7. Importierte Vorlagen können nach dem Import angepasst werden. Die Alarmeinstellungen können an Ihre Alarmanforderungen angepasst werden.

EventBridge Regelgruppen und Vorlagen zur Überwachung Ihres AWS Medien-Workflows

CloudWatch verwendet EventBridge Amazon-Regeln zum Senden von Benachrichtigungen. Sie beginnen mit der Erstellung einer Event-Vorlagengruppe. In dieser Vorlagengruppe für Ereignisse erstellen Sie Ereignisvorlagen, die festlegen, unter welchen Bedingungen eine Benachrichtigung ausgelöst wird und wer benachrichtigt wird.

In diesem Abschnitt wird die Erstellung von EventBridge Regeln mithilfe des Workflow-Monitors behandelt. Weitere Informationen darüber, wie der EventBridge Service Regeln verwendet, finden Sie unter: [EventBridge Regeln](#) im EventBridge Amazon-Benutzerhandbuch

Vorlagengruppen für Veranstaltungen erstellen

Mithilfe von Vorlagengruppen für Ereignisse können Sie Ereignisse anhand Ihres Anwendungsfalls sortieren und klassifizieren.

Um eine Vorlagengruppe für Ereignisse zu erstellen

1. Wählen Sie im Navigationsbereich der Workflow-Monitor-Konsole EventBridge Regelvorlagen aus.
2. Wählen Sie Ereignisvorlagengruppe erstellen aus.
3. Geben Sie der Alarmvorlagengruppe einen eindeutigen Gruppennamen und optional eine Beschreibung.
4. Wählen Sie Erstellen aus. Sie werden zur Detailseite der neu erstellten Alarmvorlagengruppe weitergeleitet.

Event-Vorlagen erstellen

Sie können Benachrichtigungen auf der Grundlage von Ereignisvorlagen senden, die Sie erstellt haben.

Um eine Event-Vorlage zu erstellen

1. Wählen Sie auf der Detailseite der Veranstaltungsvorlagengruppe die Option Eventvorlage erstellen aus.
2. Geben Sie der Veranstaltungsvorlage einen eindeutigen Vorlagennamen und optional eine Beschreibung.
3. Gehen Sie im Abschnitt Regeleinstellungen wie folgt vor:
 1. Wählen Sie einen Ereignistyp aus. Bei der Auswahl eines Ereignistyps können Sie zwischen mehreren Ereignissen wählen, die von einem Alarm erstellt wurden, AWS oder die Option Signal map active auswählen, um einen Alarm zu verwenden, der mit einer Alarmvorlage erstellt wurde.
 2. Wählen Sie einen Target-Service aus. Dies bestimmt, wie Sie über dieses Ereignis informiert werden möchten. Sie können Amazon Simple Notification Service oder CloudWatch Protokolle auswählen.
 3. Nachdem Sie einen Zieldienst ausgewählt haben, wählen Sie ein Ziel aus. Dies ist ein Amazon SNS SNS-Thema oder eine CloudWatch Protokollgruppe, abhängig von Ihrer Zielservice-Auswahl.
4. Wählen Sie Erstellen aus, um den Vorgang abzuschließen.

Signalkarten des Workflow-Monitors

Signalkarten sind visuelle Zuordnungen von AWS Ressourcen in Ihrem Medienworkflow. Sie können den Workflow-Monitor verwenden, um die Signalzuordnungssuche für jeden der unterstützten Ressourcentypen zu starten. Während des Erkennungsprozesses ordnet der Workflow-Monitor automatisch und rekursiv alle verbundenen AWS Ressourcen zu. Nachdem die Signalmap erstellt wurde, können Sie die Workflow-Monitor-Konsole verwenden, um beispielsweise Überwachungsvorlagen bereitzustellen, Metriken anzuzeigen und Details der zugewiesenen Ressourcen anzuzeigen.

Topics

- [Signalkarten für AWS Medien-Workflows erstellen](#)

- [Signalkarten von AWS Medien-Workflows anzeigen](#)
- [Hinzufügen von Alarm- und Ereignisvorlagen zur Signalübersicht Ihres AWS Medienworkflows](#)
- [Bereitstellung von Vorlagen für die Signallampe Ihres AWS Medienworkflows](#)
- [Aktualisierung der Signalkarte Ihres AWS Medienworkflows](#)
- [Löschen der Signallampe Ihres AWS Medien-Workflows](#)

Signalkarten für AWS Medien-Workflows erstellen

Sie können Workflow-Monitor-Signalzuordnungen verwenden, um eine visuelle Zuordnung aller verbundenen AWS Ressourcen in Ihrem Medien-Workflow zu erstellen.

Um eine Signalmap zu erstellen

1. Wählen Sie im Navigationsbereich der Workflow-Monitor-Konsole die Option Signalzuordnungen aus.
2. Wählen Sie Signalkarte erstellen aus.
3. Geben Sie der Signalmap einen Namen und eine Beschreibung.
4. Im Abschnitt Neue Signalkarte entdecken werden die Ressourcen im aktuellen Konto und in der ausgewählten Region angezeigt. Wählen Sie eine Ressource aus, um mit der Erkennung der Signalkarte zu beginnen. Die ausgewählte Ressource wird der Ausgangspunkt für die Suche sein.
5. Wählen Sie Erstellen aus. Warten Sie einen Moment, bis der Entdeckungsprozess abgeschlossen ist. Nach Abschluss des Vorgangs wird Ihnen die neue Signalkarte angezeigt.

Note

In der Workflow-Monitor-Signalmap für AWS Elemental MediaPackage Kanäle generierte Vorschauen werden vom MediaPackage Origin-Endpunkt aus bereitgestellt und es fallen Gebühren für ausgehende Datenübertragungen an. [Die Preise findest du unter: MediaPackage Preise.](#)

Signalkarten von AWS Medien-Workflows anzeigen

Mithilfe der Signalkarten des Workflow-Monitors können Sie sich eine visuelle Zuordnung aller verbundenen AWS Ressourcen in Ihrem Medien-Workflow anzeigen lassen.

Signalkartenansichten

Nachdem Sie eine Signalkarte ausgewählt haben, stehen Ihnen zwei Ansichten zur Verfügung, mit denen Sie die Signalkarte überwachen oder konfigurieren können. Signal-Map überwachen und Signal-Map konfigurieren ist eine kontextsensitive Schaltfläche, die sich oben rechts im Bereich der Signal-Map-Konsole befindet.

Wenn Sie die Signalkarte im Abschnitt Signalkarten des Navigationsfensters auswählen, wird Ihre Signalkarte in der Konfigurationsansicht angezeigt. In der Konfigurationsansicht können Sie Änderungen an den Vorlagengruppen vornehmen, die an diese Signalkarte angehängt sind, die angehängten Vorlagen bereitstellen und die grundlegenden Details und Tags der Signalkarte anzeigen.

Wenn Sie die Signalkarte im Abschnitt „Übersicht“ des Navigationsfensters auswählen, wird Ihre Signalkarte in der Überwachungsansicht angezeigt. In der Überwachungsansicht werden die CloudWatch Alarmer, EventBridge Regeln, Warnungen, Protokolle und Messwerte für diese Signalkarte angezeigt.

Die Ansicht kann jederzeit geändert werden, indem Sie oben rechts auf die Schaltfläche Signalmap überwachen/konfigurieren klicken. Für die Konfigurationsansicht sind IAM-Berechtigungen auf Administratorebene erforderlich. Die erforderlichen IAM-Berechtigungen können hier eingesehen werden: [IAM-Richtlinien zur Workflow-Überwachung](#)

In der Signalkarte navigieren

Eine Signalmap enthält Knoten für jede unterstützte AWS-Ressource, die vom Workflow-Monitor erkannt wurde. Bestimmte Ressourcen, wie MediaLive Kanäle und MediaPackage Endpunkte, können Vorschaubilder des Inhalts anzeigen, sofern Vorschaubilder verfügbar sind.

Wenn Sie einen Ressourcenknoten auswählen und im Dropdownmenü Aktionen die Option Ausgewählte Ressourcendetails anzeigen auswählen, gelangen Sie zur Detailseite des zugehörigen Dienstes. Wenn Sie beispielsweise einen MediaLive Kanal auswählen und Ausgewählte Ressourcendetails anzeigen auswählen, wird die Detailseite der MediaLive Konsole für diesen Kanal geöffnet.

Wenn Sie einen Ressourcenknoten auswählen, wird die Liste der aktiven Alarmer nur auf diesen Knoten gefiltert. Wenn Sie im aktiven Alarm den Ziel-ARN der Ressource auswählen, werden Sie zur Detailseite des zugehörigen Dienstes weitergeleitet, auf der die ausgewählte Ressource geöffnet ist.

Hinzufügen von Alarm- und Ereignisvorlagen zur Signalübersicht Ihres AWS Medienworkflows

Nachdem Sie Alarm- und Ereignisvorlagen erstellt haben, müssen Sie diese an eine Signalkarte anhängen. Jede der von Ihnen erstellten Alarm- und Ereignisvorlagen kann an alle erkannten Signalkarten angehängt werden.

Um Alarm- und Ereignisvorlagen an Ihre Signalkarte anzuhängen

1. Wählen Sie im Navigationsbereich der Workflow-Monitor-Konsole die Option Signalzuordnungen und dann die Signalkarte aus, mit der Sie arbeiten möchten.
2. Wählen Sie oben rechts auf der Signalzuordnungsseite auf der Registerkarte CloudWatchAlarmvorlagengruppen die Option Alarmvorlagengruppen anhängen CloudWatch aus.
 1. Wählen Sie in dem neuen Bereich, der geöffnet wird, alle Alarmvorlagengruppen aus, die Sie auf diese Signalzuordnung anwenden möchten, und wählen Sie dann Hinzufügen aus. Dadurch werden die ausgewählten Alarmvorlagengruppen in den Abschnitt Angehängte CloudWatch Alarmvorlagengruppen verschoben.
 2. Wenn Sie Speichern auswählen, werden Ihre Änderungen gespeichert und Sie kehren zur Signalzuordnungsseite zurück.
3. Wählen Sie rechts auf der Signalzuordnungsseite die Registerkarte EventBridge Regelvorlagengruppen und dann EventBridgeRegelvorlagengruppen anhängen aus.
 1. Wählen Sie im neuen Bereich, der geöffnet wird, alle Event-Vorlagengruppen aus, die Sie auf diese Signalmap anwenden möchten, und wählen Sie dann Hinzufügen aus. Dadurch werden die ausgewählten Regelvorlagengruppen in den Abschnitt Angehängte EventBridge Regelvorlagengruppen verschoben.
 2. Wenn Sie Speichern auswählen, werden Ihre Änderungen gespeichert und Sie kehren zur Signalzuordnungsseite zurück.
4. Sie haben der Signalkarte CloudWatch Alarm- und EventBridge Regelvorlagen zugewiesen, aber die Überwachung ist noch nicht implementiert. Im nächsten Abschnitt wird der Einsatz der Überwachungsressourcen behandelt.

Bereitstellung von Vorlagen für die Signallampe Ihres AWS Medienworkflows

Nachdem Sie die Alarm- und Ereignisvorlagen an Ihre Signalkarte angehängt haben, müssen Sie die Überwachung bereitstellen. Bis die Bereitstellung abgeschlossen ist, ist die Überwachung Ihrer Signalkarte nicht aktiv.

Der Workflow-Monitor löst nur Alarme aus, die für die ausgewählte Signallampe relevant sind. Beispielsweise kann die angehängte Alarmvorlagengruppe Alarme für mehrere Dienste enthalten, z. B. MediaLive MediaPackage, und MediaConnect. Wenn die ausgewählte Signalzuordnung nur MediaLive Ressourcen enthält, werden keine MediaPackage MediaConnect Alarme ausgelöst.

Um die Überwachungsvorlagen bereitzustellen

1. Nachdem Sie Ihrer Signalübersicht Gruppen mit Alarm- und Ereignisvorlagen hinzugefügt und Ihre Änderungen gespeichert haben, wählen Sie im Dropdownmenü Aktionen die Option Monitor bereitstellen aus.
2. Sie werden aufgefordert, die Bereitstellung zu bestätigen, und es werden die Anzahl CloudWatch und die EventBridge Ressourcen angezeigt, die erstellt werden sollen. Wenn Sie fortfahren möchten, wählen Sie Bereitstellen aus.

Note

Für die Nutzung des Workflow-Monitors fallen keine direkten Kosten an. Mit den Ressourcen, die zur Überwachung Ihres Workflows erstellt und verwendet werden, sind jedoch Kosten verbunden.

Wenn die Überwachung bereitgestellt wird, werden Amazon CloudWatch - und EventBridge Amazon-Ressourcen erstellt. Wenn Sie die AWS Managementkonsole verwenden, werden Sie vor der Bereitstellung der Überwachung auf einer Signalkarte darüber informiert, wie viele Ressourcen erstellt werden. Weitere Informationen zur Preisgestaltung finden Sie unter: [CloudWatchPreise](#) und [EventBridge Preisgestaltung](#). Der Workflow-Monitor verwendet AWS CloudFormation Vorlagen für die Bereitstellung der CloudWatch EventBridge Ressourcen. Diese Vorlagen werden in einem Amazon Simple Storage Service-Bucket der Standardklasse gespeichert, der während des Bereitstellungsprozesses von Workflow-Monitor in Ihrem Namen erstellt wird und für den Objektspeicher und Rückruf Gebühren anfallen. Weitere Informationen zur Preisgestaltung finden Sie unter: [Amazon S3 S3-Preise](#).

3. Der Status der Bereitstellung wird neben dem Namen der Signalkarte angezeigt. Der Bereitstellungsstatus ist auch im Bereich Stacks der CloudFormation Konsole sichtbar. Nach einigen Augenblicken der Erstellung und Bereitstellung der Ressourcen beginnt die Überwachung Ihrer Signalkarte.

Aktualisierung der Signalkarte Ihres AWS Medienworkflows

Wenn an Ihrem Arbeitsablauf eine Änderung vorgenommen wird, müssen Sie möglicherweise die Signalkarte erneut ermitteln und die Überwachungsressourcen erneut einsetzen. Der Workflow-Monitor ist ein Visualisierungs- und Überwachungstool, das keine Änderungen an Ihrem Workflow vornehmen kann. Signalkarten stellen eine point-in-time Visualisierung Ihres Workflows dar. Für den Fall, dass Sie Teile Ihres Medien-Workflows hinzufügen, entfernen oder erheblich ändern, empfehlen wir Ihnen, die Signalmap erneut zu entdecken. Wenn Sie mit der Signalmap Monitoring-Ressourcen verknüpft haben, empfehlen wir Ihnen, die Überwachung nach dem Rediscovery-Prozess erneut einzusetzen.

Um eine Signalkarte wieder zu entdecken

1. Wählen Sie im Navigationsbereich der Workflow-Monitor-Konsole die Option Signalzuordnungen und dann die Signalkarte aus, mit der Sie arbeiten möchten.
2. Stellen Sie sicher, dass Sie sich in der Ansicht Signalmap konfigurieren befinden. Weitere Informationen zum Ändern von Ansichten finden Sie unter: [Signalkarten anzeigen](#)
3. Wählen Sie oben rechts auf der Signalzuordnungsseite das Dropdownmenü Aktionen aus. Wählen Sie Wiederentdecken aus.
4. Ihnen wird der Wiederentdeckungsbildschirm angezeigt. Wählen Sie eine Ressource aus, die Teil des Workflows ist, den Sie gerade wiederentdecken. Wählen Sie die Schaltfläche „Wiederentdecken“.
5. Die Signalkarte wird gemäß dem aktuellen Arbeitsablauf neu erstellt. Wenn Sie Monitoring-Ressourcen erneut einsetzen müssen, bleiben Sie auf der Seite dieser Signalübersicht. Alle zuvor angehängten Überwachungsvorlagen bleiben angehängt, müssen jedoch erneut bereitgestellt werden.

Um Überwachungsvorlagen nach einer erneuten Erkennung der Signalmap erneut bereitzustellen

1. Nach der Wiederentdeckung werden Sie zur aktualisierten Signalkarte weitergeleitet. Um die Überwachungsvorlagen erneut bereitzustellen, wählen Sie im Dropdownmenü Aktionen die Option Monitor bereitstellen aus.
2. Sie werden aufgefordert, die Bereitstellung zu bestätigen, und es wird die Anzahl der EventBridge Ressourcen angezeigt CloudWatch , die erstellt werden sollen. Wenn Sie fortfahren möchten, wählen Sie Bereitstellen aus.
3. Der Status der Bereitstellung wird neben dem Namen der Signalkarte angezeigt. Nach einigen Augenblicken der Erstellung und Bereitstellung der Ressourcen beginnt die Überwachung der Signalkarte.

Löschen der Signallampe Ihres AWS Medien-Workflows

Wenn Sie eine Signalkarte nicht mehr benötigen, können Sie sie löschen. Wenn Sie Monitoring-Vorlagen auf der Signal-Map bereitgestellt haben, werden Sie beim Löschen aufgefordert, alle CloudWatch EventBridge Ressourcen zu löschen, die auf dieser Signal-Map bereitgestellt wurden. Das Löschen der bereitgestellten Ressourcen hat keine Auswirkungen auf die Vorlagen, mit denen sie erstellt wurden. Durch das Löschen von Ressourcen soll sichergestellt werden, dass Sie nicht über EventBridge Ressourcen verfügen CloudWatch , die bereitgestellt, aber nicht verwendet werden.

Um eine Signalkarte zu löschen

1. Wählen Sie im Navigationsbereich der Workflow-Monitor-Konsole die Option Signalzuordnungen aus und klicken Sie auf das Optionsfeld neben der Signalmap, die Sie löschen möchten.
2. Wählen Sie die Schaltfläche Löschen aus. Sie werden aufgefordert, das Löschen der Monitoring-Ressourcen zu bestätigen. Wählen Sie Löschen aus, um mit dem Löschen der Überwachungsressourcen zu beginnen.
3. In der Spalte Bereitstellung überwachen wird der aktuelle Status angezeigt. Wenn sich der Status in DELETE_COMPLETE geändert hat, klicken Sie erneut auf die Schaltfläche Löschen.
4. Sie werden aufgefordert, das Löschen der Signalkarte zu bestätigen. Wählen Sie Löschen, um fortzufahren und die Signalkarte zu löschen.

Workflow-Monitor-Kontingente

Der folgende Abschnitt enthält Kontingente für Workflow-Monitor-Ressourcen. Jedes Kontingent wird „pro Konto“ berechnet. Wenn Sie ein Kontingent für Ihr Konto erhöhen müssen, können Sie die [AWS Servicekontingenten-Konsole](#) verwenden, um eine Erhöhung zu beantragen, sofern in der folgenden Tabelle nichts anderes angegeben ist.

Kontingente

Ressourcentyp	Kontingent
CloudWatch Gruppen von Alarmvorlagen	20
CloudWatch Alarm-Vorlagen	200
EventBridge Gruppen von Regelvorlagen	20
EventBridge Regelvorlagen	200
Signalkarten	30
Signalkarten: CloudWatch Alarmvorlagengruppen, die an eine einzelne Signalkarte angehängt sind	5 Sie können dieses Kontingent nicht erhöhen.
Signalzuordnungen: Gruppen von EventBridge Regelvorlagen, die an eine einzelne Signalmap angehängt sind	5 Sie können dieses Kontingent nicht erhöhen.

Verwenden Sie den Workflow-Monitor

Verwenden Sie die Bereiche Übersicht und Signalzuordnungen der Workflow-Monitor-Konsole, um den aktuellen Status der Workflows und alle zugehörigen Alarme, Metriken und Protokolle zu überprüfen.

Topics

- [Überblick über den Workflow-Monitor](#)
- [Übersicht über Protokolle und Metriken für den Workflow-Monitor](#)
- [Verwendung von Signalkarten im Workflow-Monitor](#)

Überblick über den Workflow-Monitor

Der Bereich „Übersicht“ der Workflow-Monitor-Konsole ist ein Dashboard, das at-a-glance Informationen zu Ihren Signalzuordnungen enthält. Im Übersichtsbereich können Sie den aktuellen Status der Überwachung der einzelnen Signalkarten sowie die CloudWatch Metriken und alle zugehörigen CloudWatch Protokolle einsehen. Sie können eine beliebige Signalmap auswählen, um auf die Konsolenseite der Signal Maps zu gelangen.

Filterung im Überblick

Mithilfe der Suchleiste im Übersichtsbereich können Sie die Liste der Signalzuordnungen anhand kontextsensitiver Einschränkungen filtern. Nachdem Sie die Suchleiste ausgewählt haben, wird Ihnen eine Liste mit Eigenschaften angezeigt, nach denen Sie filtern können. Wenn Sie eine Eigenschaft auswählen, werden Operatoren wie „Ist gleich“, „Enthält“, „Entspricht nicht“ und „Enthält nicht“ angezeigt. Wenn Sie einen Operator auswählen, wird eine Liste mit Ressourcen aus dem ausgewählten Eigenschaftstyp erstellt. Wenn Sie eine dieser Ressourcen auswählen, werden in der Signalzuordnungsliste nur Signalkarten angezeigt, die der von Ihnen definierten Einschränkung entsprechen.

Übersicht über Protokolle und Metriken für den Workflow-Monitor

Um CloudWatch Metriken und Protokolle für eine Signal-Map anzuzeigen, wählen Sie das Optionsfeld neben dem Namen der Signal-Map aus. Unter der Liste der Signalzuordnungen wird eine Oberfläche mit Registerkarten für Metriken und Protokolle angezeigt.

CloudWatch Metriken

CloudWatch Die Metriken für die ausgewählte Signalmap sind kontextsensitiv und zeigen nur Metriken an, die mit den in diesem Signalkarten-Workflow verwendeten Diensten verknüpft sind. Sie können die Metriktools auf dem Bildschirm verwenden, um die angezeigten Metrikperioden und Zeitbereiche anzupassen.

CloudWatch Protokolle

Wenn Sie der Signalkarte eine CloudWatch Protokollgruppe zugeordnet haben, wird diese Gruppe hier angezeigt.

Verwendung von Signalkarten im Workflow-Monitor

Im Übersichtsbereich der Konsole können Sie eine bestimmte Signal-Map auswählen, um weitere Informationen zu dieser Signal-Map und den zugehörigen Monitoring-Ressourcen zu erhalten.

Nachdem Sie eine Signal-Map ausgewählt haben, werden Ihnen die Signal-Map und eine Reihe von Abschnitten mit Tabs angezeigt, die weitere Informationen enthalten:

- CloudWatch Alarme
- EventBridge Regeln
- AWS Elementare Warnmeldungen
- Kennzahlen
- Protokolle
- Grundlegende Einzelheiten

In der Signalkarte navigieren

Eine Signalmap enthält Knoten für jede unterstützte AWS-Ressource, die vom Workflow-Monitor erkannt wurde. Bestimmte Ressourcen, wie MediaLive Kanäle und MediaPackage Endpunkte, können Vorschaubilder des Inhalts anzeigen, sofern Vorschaubilder verfügbar sind.

Wenn Sie einen Ressourcenknoten auswählen und im Dropdownmenü Aktionen die Option Ausgewählte Ressourcendetails anzeigen auswählen, gelangen Sie zur Detailseite des zugehörigen Dienstes. Wenn Sie beispielsweise einen MediaLive Kanal auswählen und Ausgewählte Ressourcendetails anzeigen auswählen, wird die Detailseite der MediaLive Konsole für diesen Kanal geöffnet.

Wenn Sie einen Ressourcenknoten auswählen, wird die Liste der aktiven Alarme nur auf diesen Knoten gefiltert. Wenn Sie im aktiven Alarm den Ziel-ARN der Ressource auswählen, werden Sie zur Detailseite des zugehörigen Dienstes weitergeleitet, auf der die ausgewählte Ressource geöffnet ist.

Kontingente in AWS Elemental MediaTailor

MediaTailor Anfragen für Ressourcen und Betriebsabläufe unterliegen den folgenden Kontingenten (früher als „Beschränkungen“ bezeichnet).

Sie können den AWS Service Quotas verwenden, um Kontingente einzusehen und Kontingenterhöhungen für MediaTailor viele andere AWS Dienste zu beantragen. Weitere Informationen zu diesem Service finden Sie im [Benutzerhandbuch für Service Quotas](#).

Kontingente für das Einfügen von Anzeigen

In der folgenden Tabelle werden die Kontingente für das Einfügen von AWS Elemental MediaTailor Anzeigen beschrieben. Sofern nicht anders angegeben, sind die Kontingente nicht anpassbar.

Name	Default quota value (Standardkontingentwert)	Description	
Länge für Ad Decision Server (Entscheidungsserver für Werbung, ADS)	25,000	Die maximale Anzahl von Zeichen in einer Spezifikation des Ad Decision Server (Entscheidungsserver für Werbung, ADS).	
Umleitungen für Ad Decision Server (Entscheidungsserver für Werbung, ADS)	7	Die maximale Tiefe der Weiterleitungen, die in VAST-Wrap per-Tags MediaTailor folgt. MediaTailor gibt auf, wenn es zusätzliche Weiterleitungen gibt.	
Zeitüberschreitung für Ad Decision Server	3	Die maximale Anzahl von Sekunden, die MediaTailor warten,	

Name	Default quota value (Standardkontingen twert)	Description	
(Entscheidungsserver für Werbung, ADS)		bis das Timeout bei einer offenen Verbindung zu einem Ad Decision Server (ADS) überschri tten wird. Wenn bei einer Verbindung das Zeitlimit überschri tten MediaTailor wird, weil der ADS nicht reagiert, kann die verfügbare Anzeige nicht mit Werbung gefüllt werden.	
Werbeeinblendungs- Anforderungen	10.000	Die maximale Anzahl an Anfragen pro Sekunde für personali sierte Manifeste bei serverseitiger Anzeigeneinblendun g. Die Anzeigene infügung verarbeitet eingehende Anfragen nach Manifesten, Sitzungsinitialisierung, Tracking-Daten und Anzeigensegmenten. Dieses Kontingent ist einstellbar.	

Name	Default quota value (Standardkontingen twert)	Description	
Konfigurationen	1.000	Die maximal zulässige Anzahl von Konfigurationen MediaTailor .	
Länge für Ursprungs- Server der Inhalte	512	Die maximale Anzahl von Zeichen in einer Spezifikation des Ursprungs-Servers der Inhalte.	
Zeitüberschreitung für Ursprungs-Server der Inhalte	2	Die maximale Anzahl von Sekunden, die bei der Anforderung von Vorlagenmanifesten vor einem Timeout bei einer offenen Verbindung zum Inhaltsursprungsse rver MediaTail or gewartet wird. Zeitüberschreitung en generieren HTTP 504 (GatewayT imeoutExc eption) -Antwort- Fehler.	

Name	Default quota value (Standardkontingen- twert)	Description	
Manifestgröße	2	Die maximale Größe aller originalen Wiedergabemanifeste in MB. Um sicherzustellen, dass das Kontingent nicht überschritten wird, können Sie Ihre Eingabemanifeste mit gzip in MediaTailor komprimieren.	
Verpackungskonfigurationen	5	Die maximale Anzahl von Paketkonfigurationen pro Quelle (ob Live- oder Video-on-Demand).	
Zeitpläne vorab abrufen	25	Die maximale Anzahl aktiver Prefetch-Zeitpläne pro Wiedergabekonfiguration. Abgelaufene Prefetch-Zeitpläne werden nicht auf dieses Limit angerechnet.	

Name	Default quota value (Standardkontingen twert)	Description	
Timeout für serversei tige Reporting- Beacon-Anfragen	3 Sekunden	Die maximale Anzahl von Sekunden, die vor einem MediaTailor Timeout bei einer offenen Verbindung zum Server gewartet wird, wenn ein Beacon für serverseitige Berichte ausgelöst wird. Wenn bei einer Verbindung das Timeout überschri tten MediaTailor wird, der Beacon nicht ausgelöst werden kann und der Dienst eine ERROR_FIR ING_BEACO N_FAILED Meldung unter MediaTailor/ AdDecisionServe rInteractionsprotokoll iert, melden Sie sich an. CloudWatch	

Name	Default quota value (Standardkontingen twert)	Description	
Sitzungsablauf	Ein 10-faches der Manifestdauer	Die maximale Zeit, während der eine MediaTailor Sitzung inaktiv bleiben kann, bevor die Sitzung beendet wird. Zu Sitzungsaktivitäten können Player-An forderungen oder Kontaktversuche durch den Ursprungs -Server zählen. Wenn die Sitzung abläuft, wird ein HTTP 400 (Bad Request) Antwortfe hler MediaTailor zurückgegeben.	

Kontingente für die Kanalmontage

In der folgenden Tabelle werden die Kontingente für die AWS Elemental MediaTailor Kanalmontage beschrieben. Sofern nicht anders angegeben, sind die Kontingente [anpassbar](#).

Name	Default quota value (Standardkontingen twert)	Description	
Anfragen nach Channel-Manifests, pro Konto	400	Die maximale Anzahl von ausgehenden Manifestanfragen pro Sekunde für alle	

Name	Default quota value (Standardkontingen twert)	Description	
		Channel Assembly- Kanäle eines Kontos.	
Kanalmanifestanfra gen pro Kanal	50	Die maximale Anzahl von ausgehenden Manifestanfragen pro Sekunde für einen einzelnen Channel Assembly-Kanal.	
Kanal-Ausgaben	5	Die maximale Anzahl von Ausgaben pro Kanal.	
Kanäle pro Konto	100	Die maximale Anzahl von Kanälen pro Konto.	
Live-Quellen	50	Die maximale Anzahl von Live-Quellen für den Quellstandort.	
Programme pro Kanal	400	Die maximale Anzahl der Programme pro Kanal.	
Konfigurationen für die Segmentzu stellung	5	Die maximale Anzahl von Konfigurationen der Segmentbe reitstellung pro Quellstandort.	
Quellstandorte	50	Die maximale Anzahl von Quellstandorten pro Konto.	

Name	Default quota value (Standardkontingen twert)	Description	
VOD-Quellen	1.000	Die maximale Anzahl von Video-on-Demand-Quellen (VOD) für den Quellstandort.	

In der folgenden Tabelle werden die Drosselungsgrenzen für die Kanalmontage beschrieben. AWS Elemental MediaTailor [Sofern nicht anders angegeben, sind die Kontingente anpassbar.](#)

Name	transactions-per-secondStandard-Höchstlimit	Description	
ConfigureLogsForChannel	1	Konfigurieren Sie Protokolle für den Kanal.	
CreateChannel	1	Erstellen Sie einen Kanal.	
CreateLiveSource	1	Erstellen Sie eine Live-Quelle.	
CreateProgram	3	Erstellen Sie ein Programm.	
CreateSourceLocation	1	Erstellen Sie einen Quellspeicherort.	
CreateVodSource	1	Erstellen Sie eine VOD-Quelle.	
DeleteChannel	1	Löscht einen Kanal.	

Name	transactions-per-secondStandard-Höchstlimit	Description	
DeleteChannelPolicy	1	Löscht eine Kanalrichtlinie.	
DeleteLiveSource	1	Löscht eine Live-Quelle.	
DeleteProgram	3	Löscht ein Programm.	
DeleteSourceLocation	1	Löscht einen Quellspeicherort.	
DeleteVodSource	1	Löscht eine VOD-Quelle.	
DescribeChannel	5	Beschreiben Sie einen Kanal.	
DescribeLiveSource	5	Beschreiben Sie eine Live-Quelle.	
DescribeProgram	5	Beschreiben Sie ein Programm.	
DescribeSourceLocation	5	Beschreiben Sie einen Quellstandort.	
DescribeVodSource	5	Beschreiben Sie eine VOD-Quelle.	
GetChannelPolicy	5	Holen Sie sich eine Kanalrichtlinie.	
GetChannelSchedule	5	Holen Sie sich einen Kanalplan.	

Name	transactions-per-secondStandard-Höchstlimit	Description	
ListAlerts	5	Benachrichtigungen auflisten.	
ListChannels	5	Kanäle auflisten.	
ListLiveSources	5	Live-Quellen auflisten.	
ListPrograms	5	Programme auflisten.	
ListSourceLocations	5	Quellorte auflisten.	
ListTagsForResource	5	Listen Sie Tags für eine Ressource auf.	
ListVodSources	5	VOD-Quellen auflisten .	
PutChannelPolicy	3	Legen Sie eine Kanalrichtlinie fest.	
StartChannel	1	Starte einen Kanal.	
StopChannel	1	Stoppen Sie einen Kanal.	
TagResource	1	Versieht eine Ressource mit Tags	
UntagResource	1	Entfernt Tags von einer Ressource	
UpdateChannel	1	Aktualisiere einen Kanal.	
UpdateLiveSource	1	Aktualisiere eine Live-Quelle.	

Name	transactions-per-secondStandard-Höchstlimit	Description	
UpdateProgram	1	Aktualisieren Sie ein Programm.	
UpdateSourceLocation	1	Aktualisieren Sie einen Quellspeicherort.	
UpdateVodSource	1	Aktualisieren Sie eine VOD-Quelle.	

Problembehebung MediaTailor

Wenn bei der Arbeit mit Wiedergabefehler oder ähnliche Schwierigkeiten auftreten AWS Elemental MediaTailor, lesen Sie die Themen in diesem Abschnitt.

Themen

- [Behebung von Problemen mit dem MediaTailor Ereignisablauf](#)
- [Problembehebung bei der Wiedergabe von MediaTailor](#)
- [Beacon-Deduplizierung](#)
- [Fehlerbehebung bei Parametern](#)
- [MediaTailor Leitfaden zur Fehlerbehebung beim Überspringen von Anzeigen](#)

Behebung von Problemen mit dem MediaTailor Ereignisablauf

Das Verständnis des AWS Elemental MediaTailor Ereignisablaufs bietet eine solide Grundlage für die Behebung von Problemen bei der Anzeigeneinfügung. Durch die Analyse der Reihenfolge, des Zeitpunkts und der Muster von Ereignissen können Sie schnell erkennen, wo Probleme auftreten, und gezielte Lösungen implementieren.

Dieser Abschnitt enthält praktische Anleitungen zur Verwendung der Ereignisflussanalyse zur Diagnose von Problemen. Informationen zu den grundlegenden Konzepten des Ereignisablaufs finden Sie unter [Ablauf des Ereignisses beim Einfügen von Anzeigen](#).

Identifizierung unvollständiger Ereignisabläufe

Unvollständige Ereignisabläufe treten auf, wenn die erwartete Abfolge von Ereignissen unterbrochen wird, bevor eine erfolgreiche Manifest-Personalisierung (der Vorgang des MediaTailor Einfügens personalisierter Anzeigeninformationen in das Manifest) erreicht ist. Die Identifizierung der Stellen, an denen der Fluss unterbrochen wird, hilft dabei, die Hauptursache für Fehler bei der Anzeigeneinfügung zu ermitteln.

Häufig auftretende unvollständige Flussmuster

Verschiedene Fehlerpunkte im Ereignisablauf weisen auf bestimmte Arten von Problemen hin, z. B. auf die folgenden.

- Der Ablauf wird unterbrochen, nachdem eine Werbechance erkannt wurde: Weist auf Probleme mit Werbemarkierungen oder dem Manifest selbst hin, die MediaTailor verhindern, dass eine ADS-Anfrage gestellt werden kann. Nach der ADS-Anfrage würden Probleme mit der ADS-Konnektivität, der Konfiguration oder dem Timeout auftreten.
- Der Datenfluss wird nach einer ADS-Anfrage gestoppt: deutet auf Probleme mit der ADS-Antwort, Probleme beim VAST-Parsing, kreative Verarbeitungsfehler, ADS-Timeouts, Konnektivitätsfehler oder Konfigurationsprobleme wie ungültige ADS hin URLs , die erst bei der Anfrage entdeckt werden.
- Fehlender Tracking-Beacon: Kann auf Probleme mit der Tracking-Konfiguration, serverseitige Berichtsprobleme oder Implementierungslücken auf der Clientseite hinweisen.

CloudWatch Abfragen für unvollständige Flussanalysen

Verwenden Sie diese Amazon CloudWatch Logs Insights-Abfragen, um unvollständige Ereignisabläufe zu identifizieren. Führen Sie diese Abfragen je nach Art der benötigten Analyse anhand der entsprechenden Protokollgruppen aus.

Auswahl der Protokollgruppe:

- MediaTailor/AdDecisionServerInteractions— Wird für Abfragen verwendet, die Interaktionen mit Anzeigenentscheidungsservern, Werbemöglichkeiten und Fehler im Zusammenhang mit Werbung analysieren.
- MediaTailor/TranscodeService— Wird zur Analyse von Problemen verwendet, bei denen Anzeigen aufgrund von Transcodierungsproblemen, Fehlern bei der kreativen Verarbeitung oder anderen Problemen, die nichts mit Werbung zu tun haben, nicht eingefügt wurden.

Example Identifizieren Sie Werbemöglichkeiten ohne erfolgreiche Manifest-Personalisierung

Protokollgruppe: / MediaTailorAdDecisionServerInteractions

Die folgende Abfrage identifiziert Werbemöglichkeiten, die nicht zu einer erfolgreichen Manifest-Personalisierung geführt haben:

```
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 Analysieren Sie die Abschlussquoten im Event-Flow

Gruppe protokollieren: MediaTailor/AdDecisionServerInteractions

Die folgende Abfrage analysiert die Abschlussquoten verschiedener Ereignistypen:

```

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 sucht nach Sitzungen mit fehlenden Beacon-Ereignissen

Gruppe protokollieren: / MediaTailorAdDecisionServerInteractions

Die folgende Abfrage identifiziert Sitzungen, für die zwar alle Avails, aber keine entsprechenden Beacon-Ereignisse vorliegen:

```

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 identifiziert Fehler beim Einfügen von Anzeigen im Zusammenhang mit der Transcodierung

Protokollgruppe: / MediaTailorTranscodeService

Die folgende Abfrage identifiziert Probleme mit der Transcodierung, die eine erfolgreiche Anzeigeneinblendung verhindern:

```
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
```

Probleme beim Timing von Ereignissen werden analysiert

Die Analyse des Zeitpunkts von Ereignissen hilft dabei, Leistungsengpässe zu identifizieren und Arbeitsabläufe bei der Anzeigenschaltung zu optimieren. Ungewöhnliche Zeitmuster deuten oft auf grundlegende Probleme hin, die das Zuschauererlebnis beeinträchtigen.

Zeitliche Schwellenwerte für die Leistung

Verwenden Sie diese zeitlichen Schwellenwerte, um potenzielle Leistungsprobleme zu identifizieren.

- Gesamtflussdauer von mehr als 5 Sekunden: Kann das Nutzererlebnis beeinträchtigen und auf ADS-Leistungsprobleme, Probleme mit dem Ursprungsserver (wie Timeouts beim Abrufen von Manifesten) oder interne MediaTailor Probleme wie Infrastrukturprobleme mit NAT Gateway, DynamoDB, EC2 oder anderen Systemkomponenten hinweisen.
- ADS-Reaktionszeit von mehr als 2 Sekunden: Weist auf ADS-Leistungsprobleme oder Netzwerklatenzprobleme hin.
- Manifest-Personalisierung länger als 1 Sekunde: Kann auf Verzögerungen bei der kreativen Verarbeitung, Probleme mit dem Ursprungsserver (z. B. Timeouts beim Abrufen von Manifesten) oder interne MediaTailor Systemprobleme hinweisen, einschließlich Infrastruktureinschränkungen mit NAT Gateway, DynamoDB, EC2 oder anderen Komponenten.

Abfragen zur Timing-Analyse

Verwenden Sie diese Abfragen, um Zeitmuster für Ereignisse zu analysieren.

Example messen Sie die Gesamtdauer des Ereignisablaufs

Die folgende Abfrage misst die Gesamtdauer von Ereignisabläufen und identifiziert Ereignisse, die länger als 5 Sekunden dauern:

```
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 Analysieren Sie den Zeitpunkt der ADS-Antwort

Die folgende Abfrage analysiert die ADS-Antwortzeiten und identifiziert diejenigen, die länger als 2 Sekunden sind:

```
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 Identifizieren Sie die langsame Manifest-Personalisierung

Die folgende Abfrage identifiziert Prozesse zur Manifest-Personalisierung, die länger als 1 Sekunde dauern:

```
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
```

Häufig auftretende Probleme und Lösungen für den Ablauf von Ereignissen

In diesem Abschnitt finden Sie Lösungen für häufig auftretende Probleme mit dem Ereignisablauf, geordnet nach Problemtyp und Symptomen.

Fehler bei Anfragen des AdDecision Servers

Symptome: Der Ereignisfluss stoppt, nachdem eine Werbemöglichkeit erkannt wurde. Es wurden keine ADS-Anforderungsereignisse protokolliert.

Häufige Ursachen und Lösungen

- Fehler bei der ADS-URL-Konfiguration: Stellen Sie sicher, dass die ADS-URL in Ihrer Wiedergabekonfiguration korrekt und zugänglich ist. Im Interaktionsprotokoll für Werbeanzeigen siehst du ein ADS-Anforderungsereignis (MAKING_ADS_REQUEST), aber keine entsprechende VAST-Antwort, oft begleitet von einem `ERROR_UNKNOWN` oder einem ähnlichen Fehlerereignis.
- Probleme mit der Netzwerkverbindung: Überprüfe die Netzwerkkonnektivität zwischen MediaTailor und deinem ADS, einschließlich der Firewallregeln und der DNS-Auflösung.
- Probleme mit SSL/TLS-Zertifikaten: Stellen Sie sicher, dass Ihr ADS gültige SSL-Zertifikate von einer vertrauenswürdigen Zertifizierungsstelle verwendet. Speziell für Google Ad Manager müssen Sie sich möglicherweise an den [AWS-Support](#) wenden, um ein Konfigurations-Flag zu aktivieren, das die SSL-Zertifikate von Google akzeptiert.

Diagnoseabfrage

Die folgende Abfrage hilft bei der Diagnose fehlgeschlagener ADS-Anfragen, indem sie die Ereignissequenz verfolgt:

```
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
```

Fehler bei der Antwort des Ad Decision Servers

Symptome: ADS-Anfragen sind erfolgreich, erhalten aber MediaTailor keine Antwort, oder es treten Analysefehler auf.

Häufige Ursachen und Lösungen

- Ungültiges VAST-Format: Überprüfen Sie Ihre ADS VAST-Antworten anhand der VAST-Spezifikationsstandards.
- Probleme mit dem ADS-Timeout: Erhöhen Sie die ADS-Timeout-Einstellungen oder optimieren Sie die ADS-Reaktionszeit.
- Leeres Anzeigeninventar: Überprüfe die Verfügbarkeit von Anzeigeninventar und die Targeting-Kriterien in deiner ADS-Konfiguration.

Diagnoseabfrage

Die folgende Abfrage hilft bei der Diagnose von ADS-Antwortfehlern, indem sie Anforderungs- und Antwortereignisse untersucht:

```
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
```

Offensichtliche Personalisierungsfehler

Symptome: VAST-Antworten sind eingegangen, aber die manifeste Personalisierung schlägt fehl oder Werbung wird übersprungen.

Häufige Ursachen und Lösungen:

- Probleme mit der kreativen Transcodierung: Prüfen Sie, ob es sich bei der Anzeige um eine Anzeige handelt `NEW_CREATIVE`, für die vor dem Einfügen eine Transcodierung erforderlich ist. Sie können auch nach Transkodierungsfehlern suchen, indem Sie das MediaTailor/TranscodeService Protokoll auf Fehlerereignisse wie `INTERNAL_ERROR`, oder überprüfen. `MISSING_VARIANTS`, `PROFILE_NOT_FOUND`
- Probleme mit nicht übereinstimmender Dauer: Stellen Sie sicher, dass die Anzeigendauer innerhalb der verfügbaren Dauer von Werbeunterbrechungen liegt.
- Probleme mit dem Personalisierungsgrenzwert: Überprüfe die Einstellungen für den Personalisierungsgrenzwert in deiner Wiedergabekonfiguration.

Diagnoseabfrage

Die folgende Abfrage hilft bei der Diagnose von offensichtlichen Personalisierungsfehlern, indem sie VAST-Antworten und ausgefüllte Avails untersucht:

```
fields @timestamp, eventType, sessionId, skippedAds
| filter sessionId = "your-session-id"
| filter eventType in ["VAST_RESPONSE", "FILLED_AVAIL", "WARNING_NO_ADVERTISEMENTS"]
| sort @timestamp asc
```

Frage nach Gründen übersprungener Anzeigen

Die folgende Abfrage enthält detaillierte Informationen darüber, warum Anzeigen übersprungen wurden:

```
fields @timestamp, eventType, sessionId, skippedAds.reason, skippedAds.creativeUniqueId
| filter sessionId = "your-session-id"
| filter eventType = "WARNING_NO_ADVERTISEMENTS" or ispresent(skippedAds)
| sort @timestamp asc
```

Frage nach Gründen für übersprungene Anzeigen und nach individuellem Creative IDs

Die folgende Abfrage enthält detaillierte Informationen zu übersprungenen Anzeigen, einschließlich der Gründe und der eindeutigen Werbeeinblendung IDs für die ersten beiden Anzeigen.

```
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
```

Verfolgen von Beacon-Ausfällen

Symptome: Erfolgreiche Manifest-Personalisierung, aber fehlende oder fehlgeschlagene Tracking-Beacons.

Häufige Ursachen und Lösungen

- Probleme bei der clientseitigen Implementierung: Die meisten Probleme mit Tracking-Beacons sind auf Implementierungsprobleme auf der Client-Seite zurückzuführen, z. B. wenn das Tracking nicht URLs häufig genug für das clientseitige Tracking abgerufen wird, oder auf Probleme mit der spieterspezifischen Beacon-Auslöselogik.
- Probleme mit der URL-Barrierefreiheit verfolgen: Stellen Sie sicher, dass Tracking in VAST-Antworten verfügbar ist, und geben Sie entsprechende Antworten zurück. URLs Probleme können auftreten, wenn wir nicht erreichbar URLs sind oder wenn interne MediaTailor Probleme auftreten, die eine erfolgreiche Zustellung von Antworten verhindern.

- Probleme bei der Anforderung von Spielersegmenten: Offensichtliche Fehler beim Tracking Beacon können auftreten, wenn der Client-Player eigentlich keine Segmente anfordert. Dies führt dazu, dass keine Beacons gesendet werden, was als Tracking-Fehler erscheint, aber in Wirklichkeit ist es eher ein Problem mit der Spielerimplementierung als ein Beacon-Problem.

Diagnoseabfrage

Die folgende Abfrage hilft bei der Diagnose von Tracking-Beacon-Ausfällen, indem sie ausgefüllte Avails und Beacon-Ereignisse untersucht:

```
fields @timestamp, eventType, sessionId
| filter sessionId = "your-session-id"
| filter eventType in ["FILLED_AVAIL", "BEACON_FIRED", "ERROR_FIRING_BEACON_FAILED"]
| sort @timestamp asc
```

Bewährte Methoden zur Überwachung des Ereignisflusses

Implementieren Sie diese Überwachungspraktiken, um Probleme im Ereignisablauf proaktiv zu identifizieren und zu lösen:

Alarmer einrichten CloudWatch

Erstellen Sie CloudWatch Amazon-Alarmer, um wichtige Kennzahlen zum Ablauf von Ereignissen zu überwachen.

- Alarm zur Abschlussrate des Workflows: Warnmeldung, wenn das Verhältnis zwischen erfolgreicher Personalisierung von Manifesten und Werbemöglichkeiten unter akzeptable Schwellenwerte fällt.
- Alarm bei der ADS-Reaktionszeit: Überwachen Sie die durchschnittlichen ADS-Antwortzeiten und benachrichtigen Sie, wenn diese Leistungsgrenzwerte überschreiten.
- Alarm bei der Fehlerquote: Verfolgen Sie die Häufigkeit von Fehlerereignissen und warnen Sie bei ungewöhnlichen Spitzenwerten bei bestimmten Fehlertypen.

Regelmäßige Überwachungsanfragen

Führen Sie diese Abfragen regelmäßig aus, um einen Überblick über den Zustand des Ereignisablaufs zu behalten:

Example Erfolgsquote des täglichen Ablaufs von Ereignissen

Die folgende Abfrage bietet einen täglichen Überblick über die Erfolgsquoten des Ereignisablaufs nach Ereignistyp:

```
fields @timestamp, eventType
| filter @timestamp > datefloor(@timestamp, 1d)
| stats count() as total_events by eventType
| sort total_events desc
```

Example Entwicklung der stündlichen Fehlerrate

Mit der folgenden Abfrage werden die Fehlerraten pro Stunde erfasst, um Trendprobleme zu identifizieren:

```
fields @timestamp, eventType
| filter eventType like /ERROR_/
| stats count() as error_count by datefloor(@timestamp, 1h) as hour
| sort hour desc
```

Anleitungen zur Leistungsoptimierung

Verwenden Sie die Analyse des Ereignisablaufs, um die Leistung der Anzeigenschaltung zu optimieren.

- **ADS-Optimierung:** Arbeiten Sie mit Ihrem ADS-Anbieter zusammen, um die Antwortzeiten zu optimieren und die Latenz zu reduzieren.
- **Kreative Vorbereitung:** Transkodieren Sie Werbemittel vorab, damit sie zu Ihren Inhaltsprofilen passen und Verzögerungen bei der Bearbeitung reduzieren.
- **Konfigurationsoptimierung:** Passen Sie die Timeout-Einstellungen, Personalisierungsschwellen und andere Konfigurationsparameter auf der Grundlage der Ereignisflussanalyse an.

Weitere Ressourcen zur Fehlerbehebung

Weitere Hinweise zur Fehlerbehebung, die über die Ereignisflussanalyse hinausgehen, finden Sie unter:

- Ausführliche Informationen zum Protokollformat und zu den technischen Spezifikationen finden Sie unter [Anzeigen von Protokollen](#).

- Eine umfassende Lösung für häufig auftretende Probleme beim Einfügen von Anzeigen finden Sie unter [Behebung häufiger Probleme](#).
- Anleitungen zur Einrichtung von Überwachungs- und Warnmeldungen finden Sie unter [Überwachung AWS Elemental MediaTailor mit CloudWatch Amazon-Metriken](#).
- Informationen zur Debug-Protokollierung finden Sie unter [Debug-Protokolle generieren](#)

Problembehebung bei der Wiedergabe von MediaTailor

Dieser Abschnitt enthält Informationen über die HTTP-Fehlercodes, die Sie möglicherweise beim Testen Ihrer Player-Software und während der normalen Verarbeitung von Player-Anforderungen erhalten.

Note

Möglicherweise erhalten Sie auch Fehler von der AWS Elemental MediaTailor API bei Konfigurationsvorgängen wie `PutPlaybackConfiguration` und `GetPlaybackConfiguration`. Informationen zu diesen Fehlertypen finden Sie in der [AWS Elemental MediaTailor API-Referenz](#).

Wenn dein Spieler direkt oder über ein CDN eine Anfrage an sendet, MediaTailor antwortet er mit einem Statuscode. AWS Elemental MediaTailor Wenn die Anfrage MediaTailor erfolgreich bearbeitet wurde, wird der HTTP-Statuscode `200 OK`, der den Erfolg anzeigt, zusammen mit dem ausgefüllten Manifest zurückgegeben. Wenn die Anfrage nicht erfolgreich ist, werden ein HTTP-Statuscode, ein Ausnahmenname und eine Fehlermeldung MediaTailor zurückgegeben.

AWS Elemental MediaTailor gibt zwei Klassen von Fehlern zurück:

- **Client-Fehler** — Fehler, die normalerweise durch ein Problem in der Anfrage selbst verursacht werden, z. B. durch eine falsch formatierte Anfrage, einen ungültigen Parameter oder eine fehlerhafte URL. Diese Fehler haben einen HTTP- `4xx`-Antwort-Code.
- **Serverfehler** — Fehler, die normalerweise durch ein Problem mit MediaTailor oder einer seiner Abhängigkeiten verursacht werden, z. B. durch den Ad Decision Server (ADS) oder den Ursprungsserver. Diese Fehler haben einen HTTP- `5xx`-Antwort-Code.

Themen

- [Client-Wiedergabefehler wurden zurückgegeben von AWS Elemental MediaTailor](#)
- [Fehler bei der Serverwiedergabe wurden zurückgegeben von AWS Elemental MediaTailor](#)
- [Beispiele für Fehler bei der Wiedergabe](#)

Client-Wiedergabefehler wurden zurückgegeben von AWS Elemental MediaTailor

Allgemeine Hinweise:

- Detaillierte Informationen zu den meisten Fehlern finden Sie in den Kopfzeilen und im Hauptteil der Antwort.
- Bei einigen Fehlern müssen Sie Ihre Konfigurationseinstellungen überprüfen. Sie können die Einstellungen für Ihre Wiedergabekonfiguration abrufen AWS Elemental MediaTailor. Für die API ist die Ressource `GetPlaybackConfiguration/Name`. Einzelheiten finden Sie in der [AWS Elemental MediaTailor API-Referenz](#).

In der folgenden Tabelle sind die Client-Fehlercodes aufgeführt, die durch die Manifest-Manipulationsaktivitäten zurückgegeben werden AWS Elemental MediaTailor, mögliche Ursachen und Maßnahmen, die Sie zu ihrer Behebung ergreifen können.

Code	Name der Ausnahme	Bedeutung	Vorgehensweise
400	<code>BadRequestException</code>	MediaTailor kann die Anfrage aufgrund eines oder mehrerer Formatierungs- oder Inhaltsfehler nicht bearbeiten. Ein Parameter ist möglicherweise nicht ordnungsgemäß formatiert oder die Anforderung enthält möglicherweise eine ungültige Playback-Konfiguration oder Sitzungs-ID.	Stellen Sie sicher, dass Ihre Anforderung ordnungsgemäß formatiert ist und korrekte Informationen enthält. Stellen Sie sicher, dass die Einstellung des Wiedergabe-Endpunkts am Player mit der Manifest-EndpointPrefix-Einstellung übereinstimmt, die von <code>GetPlaybackConfiguration</code> zurückgegeben wird.

Code	Name der Ausnahme	Bedeutung	Vorgehensweise
			Wiederholen Sie Ihre Anforderung.
40	AccessDeniedException	Der in der Anforderung angegebene Host-Header stimmt nicht mit dem Präfix des Manifest-Endpunkts überein, das in der MediaTailor-Wiedergabe-URL konfiguriert ist. Ihr CDN ist möglicherweise falsch konfiguriert.	Überprüfen Sie Ihre CDN-Einstellungen und stellen Sie sicher, dass Sie das richtige Präfix des Manifest-Endpunkts für MediaTailor verwenden. Wiederholen Sie Ihre Anforderung.
40	NotFoundException	MediaTailor kann die angegebenen Informationen nicht finden. Mögliche Gründe sind eine URL, die keiner Entsprechung im Service zugeordnet ist, eine nicht definierte Konfiguration oder eine nicht verfügbare Sitzung.	Überprüfen Sie Ihre Konfiguration und die Gültigkeit Ihrer Anforderung und initialisieren Sie die Session dann neu.
40	ConflictException	Ein Player hat versucht, für eine einzige Sitzung mehrere Wiedergabelisten gleichzeitig zu laden. Infolgedessen wurde ein Sitzungskonsistenzkonflikt festgestellt. Dieses Problem tritt bei HLS-Playern auf.	Stellen Sie sicher, dass Ihr Player nur jeweils eine Wiedergabeliste anfordert. Dies stimmt mit der HLS-Spezifikation überein.

Cc	Name der Ausnahme	Bedeutung	Vorgehensweise
41	Gone	Ein AWS Support-Mitarbeiter hat eine Spielsitzung oder Kundenkonfiguration blockiert. AWS Der Support tut dies in seltenen Fällen, wenn wir eine sehr hohe Anzahl von 4xx-Anfragen feststellen, die auf fehlerhaften Datenverkehr für eine einzelne Sitzung oder Konfiguration zurückzuführen sind.	Wenn Sie der Meinung sind, dass die Anfrage nicht blockiert werden sollte, wenden Sie sich an den AWS-Support . Sie können das überprüfen und nötigenfalls den Blockfilter entfernen.

Wenn Sie weitere Support benötigen, wenden Sie sich an den [AWS-Support](#).

Fehler bei der Serverwiedergabe wurden zurückgegeben von AWS Elemental MediaTailor

Allgemeine Hinweise:

- Detaillierte Informationen zu den meisten Fehlern finden Sie in den Kopfzeilen und im Hauptteil der Antwort.
- Bei einigen Fehlern müssen Sie Ihre Konfigurationseinstellungen überprüfen. Sie können die Einstellungen für Ihre Wiedergabekonfiguration unter abrufen AWS Elemental MediaTailor. Für die API ist die Ressource `GetPlaybackConfiguration/Name`. Einzelheiten finden Sie in der [AWS Elemental MediaTailor API-Referenz](#).

In der folgenden Tabelle sind die Serverfehlercodes aufgeführt, die von den Aktivitäten zur Manipulation von Manifesten zurückgegeben wurden AWS Elemental MediaTailor, mögliche Ursachen und Maßnahmen, die Sie zu ihrer Behebung ergreifen können.

Code	Name der Ausnahme	Bedeutung	Vorgehensweise
50	InternalServerError	Unbehandelte Ausnahme	Wiederholen Sie die Anforderung. Wenn das Problem weiterhin besteht, überprüfen Sie den gemeldeten Status von MediaTailor für Ihre AWS Region unter https://status.aws.amazon.com/ .
50	BadGatewayException	Entweder die Adresse des Ursprungsservers oder die Adresse des Ad Decision Servers (ADS) ist ungültig. Beispiele für ungültige Adressen sind eine private IP-Adresse und localhost .	Stellen Sie sicher, dass Ihre Konfiguration über die richtigen Einstellungen für Ihren ADS und Ursprungs-Server verfügt, und wiederholen Sie die Anforderung.
50	UnsupportedManifestException	Entweder wurde das Ursprungsmanifest geändert, sodass es nicht personalisiert werden MediaTailor kann, oder es unterstützt das Manifestformat des Ursprungs MediaTailor nicht.	Dies wirkt sich möglicherweise nur auf diese einzelne Sitzung aus. Initialisieren Sie die Sitzung erneut. Sie können dies gewöhnlich erreichen, indem Sie die Seite auf der Anzeige aktualisieren. Wenn das Problem weiterhin besteht, stellen Sie sicher, dass das Manifest-Format des Ursprungs-Servers von MediaTailor unterstützt wird. Weitere Informationen finden Sie unter Integration einer Inhaltsquelle .
50	LoadShed	MediaTailor bei der Bearbeitung Ihrer Anfrage kam es zu einer Ressourcenbeschränkung.	Wiederholen Sie die Anforderung. Falls das Problem weiterhin besteht, überprüfen Sie den

Code	Name der Ausnahme	Bedeutung	Vorgehensweise
			MediaTailor für Ihre AWS Region gemeldeten Status unter https://status.aws.amazon.com/ .
50	ThrottlingException	Da Ihre Transaktionen pro Sekunde das für Sie geltende Kontingent erreicht haben, wird Ihre Nutzung von MediaTailor abgelehnt.	Wiederholen Sie die Anforderung. Sie können den gemeldeten Zustand von MediaTailor für Ihre AWS Region auch unter https://status.aws.amazon.com/ überprüfen. Möglicherweise möchten Sie das Kontingent für Ihre Transaktionen pro Sekunde erhöhen. Weitere Informationen finden Sie unter the section called “Kontingente für das Einfügen von Anzeigen” .
50	GatewayTimeoutException	Bei der Kontaktaufnahme mit dem Ursprungsserver ist ein Timeout aufgetreten MediaTailor .	Wiederholen Sie die Anforderung. Wenn das Problem weiterhin besteht, überprüfen Sie den Zustand des Ursprungs-Servers und stellen Sie sicher, dass der Ursprungs-Server innerhalb des Timeouts des Ursprungs-Servers für Inhalte antwortet, das unter the section called “Kontingente für das Einfügen von Anzeigen” aufgeführt wird.

Wenn Sie weitere Support benötigen, wenden Sie sich an den [AWS-Support](#).

Beispiele für Fehler bei der Wiedergabe

In diesem Abschnitt sind einige Beispiele für Wiedergabefehler aufgeführt, die bei Interaktionen mit der Befehlszeile auftreten können AWS Elemental MediaTailor.

Das folgende Beispiel zeigt das Ergebnis, wenn ein Timeout zwischen dem AWS Elemental MediaTailor Ad Decision Server (ADS) oder dem Ursprungsserver auftritt.

```
~[ ]> 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]"}%
```

Beacon-Deduplizierung

Dieser Abschnitt enthält Anleitungen zur Fehlerbehebung für die MediaTailor Beacon-Deduplizierungsfunktion im serverseitigen Tracking.

Symptom: Niedrigere Anzahl von Beacons als zuvor beobachtet

Ursache: Durch die Beacon-Deduplizierung werden doppelte Ereignisse verhindert, die zuvor mehrfach gezählt wurden.

Lösung: Dieses Verhalten ist zu erwarten und weist auf eine genauere Messung hin. Bei den neuen Zählern handelt es sich eher um tatsächliche Ereignisse beim Aufrufen von Werbeanzeigen als um Anfragen aus technischen Segmenten. Aktualisieren Sie Ihre Berichtsbasislinien, um die deduplizierten Zahlen widerzuspiegeln.

Symptom: Die Anzahl der Beacons variiert je nach Player-Implementierung

Ursache: Verschiedene Player haben möglicherweise unterschiedliche Muster für Segmentanfragen, aber die Deduplizierung gewährleistet unabhängig von diesen Unterschieden eine konsistente Anzahl von Beacons.

Lösung: Dieses Verhalten wird erwartet. Die Deduplizierung ermöglicht konsistente Messungen für verschiedene Spielertypen und Netzwerkbedingungen.

Symptom: Diskrepanz zwischen der erwarteten und der tatsächlichen Anzahl von Beacons

Ursache: Frühere Erwartungen beruhten möglicherweise auf überhöhten Zahlen, die auf die doppelte Auslösung von Signalen zurückzuführen waren.

Lösung: Überprüfen Sie historische Daten, um Muster doppelter Beacons zu identifizieren. Verwenden Sie die neuen deduplizierten Zählungen als genaue Ausgangsbasis für Messungen und Berichte.

Fehlerbehebung bei Parametern

Informationen zur Behebung von Problemen im Zusammenhang mit dynamischen Anzeigenvariablen, Manifest-Abfrageparametern, Zeichenbeschränkungen, Längenbeschränkungen und Konfigurationsaliasnamen finden Sie [MediaTailor Leitfaden zur Fehlerbehebung bei Parametern](#) unter [MediaTailor dynamische Anzeigenvariablen für ADS-Anfragen](#).

MediaTailor Leitfaden zur Fehlerbehebung beim Überspringen von Anzeigen

Das Überspringen von Werbeanzeigen ist eines der häufigsten Probleme, über die MediaTailor Kunden berichten. Dieser Leitfaden hilft Ihnen bei der Identifizierung von Problemen beim Überspringen von Anzeigen und bietet Anleitungen step-by-step zur Problembehebung.

Symptome und Auswirkungen

Wenn Werbeanzeigen übersprungen werden, treten möglicherweise die folgenden Symptome auf:

- Während erwarteter Werbepausen werden keine Anzeigen geschaltet
- Werbeunterbrechungen zeigen Inhalte statt Werbung
- Inkonsistente Anzeigenwiedergabe über verschiedene Zuschauersitzungen
- CloudWatch Protokolle, die AdSkipped Ereignisse mit verschiedenen Gründen zum Überspringen anzeigen

Das Überspringen von Werbeanzeigen wirkt sich direkt auf den Umsatz aus und kann zu schlechten Zuschauererlebnissen führen. Beheben Sie diese Probleme umgehend, um eine optimale Leistung aufrechtzuerhalten.

Identifizieren von Gründen für das Überspringen

Verwenden Sie CloudWatch Logs Insights, um die MediaTailor/AdDecisionServerInteractions Protokollgruppe nach übersprungenen Anzeigen abzufragen:

```
fields @timestamp, avail.availId, skippedAds.0.skippedReason,  
skippedAds.0.creativeUniqueId  
| filter eventType = "FILLED_AVAIL" and ispresent(skippedAds.0.skippedReason)  
| sort @timestamp desc
```

Diese Abfrage gibt die letzten Ereignisse, bei denen Werbung übersprungen wurde, mit ihren spezifischen Gründen zurück, sodass Sie Muster leichter erkennen können.

Themen

- [MediaTailor NEW_CREATIVE — Problembehebung beim Überspringen von Werbeanzeigen](#)
- [MediaTailor Fehlerbehebung bei ADS-Timeout und Überspringen](#)
- [MediaTailor Fehlerbehebung beim Analysieren und Überspringen von VAST](#)
- [MediaTailor Nichtübereinstimmung bei Dauer und Problembehebung beim Überspringen](#)
- [MediaTailor Problembehandlung bei Sitzungsvariablen und Überspringen](#)
- [MediaTailor und das Überspringen von Überwachung und Warnmeldungen](#)
- [MediaTailor Bewährte Methoden zur Vorbeugung gegen das Überspringen von Anzeigen](#)
- [MediaTailor Referenzleitfaden zum Überspringen von Anzeigen](#)

MediaTailor NEW_CREATIVE — Problembehebung beim Überspringen von Werbeanzeigen

Wenn Anzeigen mit Angabe des NEW_CREATIVE Grundes übersprungen werden, ist AWS Elemental MediaTailor auf eine Anzeige gestoßen, die vor dem Einfügen transkodiert werden muss. In dieser Anleitung zur Fehlerbehebung werden die Ursachen erläutert und step-by-step Lösungsverfahren beschrieben.

Was verursacht das Überspringen von NEW_CREATIVE

MediaTailor transkodiert Anzeigen auf der Grundlage von drei Schlüsselfaktoren: Creative-ID, AWS-Konto-ID und Transcode-Varianten-Set (die Wiedergabeversionen für den zugrunde liegenden Inhaltsstream). Wenn sich ein Teil der Creative-ID oder des Transcodierungsvariantensatzes unterscheidet, wird die Anzeige als neue Variante MediaTailor erkannt, für die eine Transcodierung erforderlich ist.

Die Bitrate ändert sich

Änderungen der Bitrate können dazu führen, dass NEW_CREATIVE überspringt, wenn die Bitrate nach dem Runden auf die nächsten 8.000 Bit nicht übereinstimmt. Dies tritt häufig in den folgenden Situationen auf:

- Der zugrunde liegende Inhaltsstream ändert das primäre Manifest
- Neue Sitzungen werden mit anderen Varianten als bestehende Sitzungen erstellt
- Die Bitraten für Inhaltsstreams sind bei allen Wiedergabesitzungen inkonsistent

Bandbreitenvarianz bei Harvest-Aufträgen

MediaTailor verwendet bei Harvest-Jobs einen Schwellenwert für die Bandbreitenvarianz von 15%, wenn Anzeigen Inhaltsstreams zugeordnet werden. Wenn zwischen der Bandbreite des Livestreams und der Bandbreite der gesammelten HLS-Datei ein Unterschied von 15% oder mehr besteht, wird die Anzeige erneut transkodiert. Dies kann dazu führen, dass NEW_CREATIVE während des Transcodierungsvorgangs überspringt.

Dieses Verhalten tritt häufig auf, wenn:

- Dieselben vorcodierten Anzeigen werden sowohl für Live-Streaming- als auch für VOD-Inhalte verwendet
- Der gesammelte Inhalt hat andere Kodierungsparameter als der ursprüngliche Livestream
- Bandbreitenunterschiede überschreiten den Schwellenwert von 15% zwischen Live-Inhalten und gesammelten Inhalten

Konflikte bei der kreativen ID

Wenn Sie MediaTailor auf eine andere Creative-ID für eine Mediendatei stoßen, die bereits transkodiert wurde, passiert die folgende Sequenz:

1. Die Anzeige wurde aus dem Grund NEW_CREATIVE übersprungen
2. Dies führt zu einem unnötigen Transcodierungsversuch
3. Das Creative ist als DUPLICATE_TRANSCODE oder COPY_DEDUP markiert

Note

MediaTailor läuft nicht ab und löscht keine transkodierten Anzeigen. Wir speichern sie auf unbestimmte Zeit in einem MediaTailor eigenen S3-Bucket.

Lösungsschritte

So lösen Sie Probleme beim Überspringen von NEW_CREATIVE-Anzeigen:

1. Stellen Sie sicher, dass Ihr Ad Decision Server konsistente Werbeanzeigen IDs für denselben Anzeigeninhalt zurückgibt.

2. Vergewissern Sie sich, dass Ihr Content-Stream konsistente Bitraten und Variantengruppen beibehält.
3. Erwägen Sie die Implementierung von Anzeigenvorabruf, um sicherzustellen, dass Anzeigen vor der Wiedergabe transkodiert werden. Weitere Informationen finden Sie unter [the section called “Anzeigen vorab abrufen”](#).
4. Bei anhaltenden Problemen wenden Sie sich an den [AWS-Support](#), um weitere Support bei der Fehlerbehebung zu erhalten.

Überwachung von NEW_CREATIVE-Mustern

Verwenden Sie diese CloudWatch Logs Insights-Abfrage, um Muster beim Überspringen von NEW_CREATIVE-Anzeigen zu analysieren:

```
fields @timestamp, sessionId, creativeId, skipReason, MediaFileSourceUrl
| filter skipReason = "NEW_CREATIVE"
| stats count() by creativeId, MediaFileSourceUrl
| sort count desc
| limit 50
```

MediaTailor Fehlerbehebung bei ADS-Timeout und Überspringen

Wenn Anzeigen aus ADS_TIMEOUT oder ähnlichen Gründen übersprungen werden, haben Sie Verbindungs- oder Leistungsprobleme mit Ihrem Anzeigenentscheidungsserver. AWS Elemental MediaTailor erfordert eine zuverlässige Kommunikation mit Ihrem ADS, um Anzeigen erfolgreich einzufügen. In dieser Anleitung zur Fehlerbehebung wird erklärt, wie Sie diese Verbindungsprobleme identifizieren und beheben können.

Häufige Probleme mit der ADS-Konnektivität

Zu den häufigsten Problemen mit der ADS-Konnektivität gehören die folgenden:

- Auf den AD-Entscheidungsserver kann nicht zugegriffen werden MediaTailor
- ADS reagiert nicht innerhalb des konfigurierten Timeout-Zeitraums
- ADS ist in Spitzenzeiten nicht in der Lage, das Anforderungsvolumen zu bewältigen
- Probleme mit der Netzwerkverbindung zwischen MediaTailor und Ihrem ADS

Lösungsschritte

So beheben Sie Probleme mit dem ADS-Timeout:

1. Stellen Sie sicher, dass Ihr Anzeigenentscheidungsserver von aus MediaTailor erreichbar ist.
2. Vergewissern Sie sich, dass Ihr ADS innerhalb des konfigurierten Timeouts reagiert.
3. Stellen Sie sicher, dass Ihr ADS das Anforderungsvolumen in Spitzenzeiten bewältigen kann.
4. Erwägen Sie die Implementierung einer alternativen Anzeigenstrategie für den Fall, dass Ihr primäres ADS nicht verfügbar ist.

Überwachung der ADS-Leistung

Richten Sie CloudWatch Alarmer für die `AdDecisionServer.Timeouts` Metrik ein, um ADS-Konnektivitätsprobleme proaktiv zu erkennen.

MediaTailor Fehlerbehebung beim Analysieren und Überspringen von VAST

Wenn Anzeigen mit `VAST_PARSING_ERROR` oder übersprungen werden `MEDIA_FILE_UNAVAILABLE`, haben Sie Probleme mit Ihren VAST-Antworten oder Anzeigenmediendateien. AWS Elemental MediaTailor Für eine erfolgreiche Anzeigenschaltung sind ordnungsgemäß formatierte VAST-Antworten und barrierefreie Mediendateien erforderlich. In dieser Anleitung zur Fehlerbehebung wird erklärt, wie Sie diese Probleme identifizieren und lösen können.

Häufig auftretende VAST-Probleme

Zu den häufigsten VAST-Problemen gehören die folgenden:

- Das VAST-Antwortformat entspricht nicht der VAST-Spezifikation
- Die Mediendatei URLs in der VAST-Antwort ist nicht öffentlich zugänglich
- Sonderzeichen sind in VAST XML nicht richtig codiert
- Bei der VAST-Antwort fehlen die erforderlichen Mediendateien oder Formate

Lösungsschritte

So lösen Sie Probleme beim VAST-Parsing:

1. Überprüfen Sie Ihr VAST-Antwortformat anhand der VAST-Spezifikation.

2. Stellen Sie sicher, dass alle Mediendateien URLs in der VAST-Antwort öffentlich zugänglich sind.
3. Prüfen Sie die korrekte Kodierung von Sonderzeichen in Ihrem VAST-XML.
4. Stellen Sie sicher, dass Ihre VAST-Antwort Mediendateien in Formaten enthält, die mit kompatibel sind MediaTailor.

Fehlerbehebung beim VAST-Wrapper

Bei Problemen mit `INVALID_VAST_WRAPPER_AD` oder `REJECTED_REPLICA_VAST`:

- Überprüfen Sie die VAST-Wrapper-Antworten anhand der VAST-Spezifikation
- Stellen Sie sicher, dass alle Wrapper-Elemente richtig formatiert sind und einen gültigen TagURI enthalten VASTAd
- Überprüfen Sie die Ad-Server-Konfiguration auf Richtlinien zur Erkennung doppelter Inhalte
- Stellen Sie sicher, dass VAST-Antworten innerhalb derselben Sitzung einzigartige kreative Inhalte enthalten

MediaTailor Nichtübereinstimmung bei Dauer und Problembehebung beim Überspringen

Wenn Anzeigen aus Gründen oder aufgrund des Formats übersprungen werden

`AVAIL_DURATION_EXCEEDEDLEFTOVER_AVAIL_EXCEEDED_THRESHOLD`, treten Probleme mit der Dauer oder der Formatkompatibilität auf. AWS Elemental MediaTailor erfordert die korrekte Formatierung der Dauer und den richtigen Abgleich zwischen dem Anzeigeninhalt und der verfügbaren Zeit für Werbeunterbrechungen. In dieser Anleitung zur Fehlerbehebung wird erklärt, wie Sie diese Probleme im Zusammenhang mit der Dauer identifizieren und lösen können.

Häufig auftretende Probleme mit der Dauer

Zu den häufigsten Problemen mit der Dauer gehören:

- Anzeigen, die länger als die verfügbare Dauer der Werbeunterbrechungen sind
- Falsche Markierungen für Werbeunterbrechungen im Inhalt signalisieren eine falsche Dauer
- Der Personalisierungsschwellenwert für die Anzeigeneinblendung wurde nicht erreicht
- Falsches Format für die Dauer in den Tags `EXT-X-CUE-OUT`

Anforderungen an das Format für die Dauer

Der Parameter für die EXT-X-CUE-OUT Tag-Dauer muss als Ganzzahl formatiert werden, nicht als ISO 8601-Dauerformat.

Anforderungen an das Format für die Dauer

Format	Beispiel	Status
Ganzzahl (richtig)	32	Unterstützt — steht für 32 Sekunden
Dezimal (richtig)	30.000	Unterstützt — steht für 30 Sekunden
ISO 8601 (falsch)	PT32S	Nicht unterstützt — verursacht Fehler beim Einfügen

Lösungsschritte

So beheben Sie Probleme mit nicht übereinstimmender Dauer:

1. Stellen Sie sicher, dass in Ihrem ADS Anzeigen angezeigt werden, die innerhalb der verfügbaren Dauer der Werbeunterbrechung liegen.
2. Vergewissern Sie sich, dass Ihre Markierungen für Werbeunterbrechungen im Inhalt die beabsichtigte Dauer korrekt angeben.
3. Vergewissern Sie sich, dass die Parameter für die EXT-X-CUE-OUT Dauer das Ganzzahlformat verwenden.
4. Erwägen Sie, den Schwellenwert für die Personalisierung gegebenenfalls für Ihren Anwendungsfall anzupassen.

MediaTailor Problembehandlung bei Sitzungsvariablen und Überspringen

Sitzungsvariablen spielen eine entscheidende Rolle bei der Ausrichtung und Auswahl von Anzeigen in AWS Elemental MediaTailor. Eine falsche Konfiguration der Sitzungsvariablen ist eine häufige Ursache für Probleme beim Überspringen von Anzeigen. In diesem umfassenden Leitfaden zur Fehlerbehebung wird erklärt, wie Probleme mit Sitzungsvariablen identifiziert und gelöst werden können, die eine erfolgreiche Anzeigeneinblendung verhindern können.

Häufig auftretende Probleme mit Sitzungsvariablen

Zu den häufigsten Problemen mit Sitzungsvariablen gehören:

- **Fehlende erforderliche Variablen:** Ihr Anzeigenentscheidungsserver benötigt möglicherweise bestimmte Variablen, die nicht bereitgestellt werden.
- **Falsche Variablensyntax:** Variablen müssen die richtige Syntax verwenden (z. B. `[session.id]` anstelle von `${session.id}`).
- **Probleme mit der URL-Codierung:** Für Sonderzeichen in Variablenwerten ist möglicherweise eine korrekte URL-Codierung erforderlich.
- **Inkonsistente Player-Parameter:** Player-Parameter müssen über Sessions hinweg einheitlich weitergegeben werden.
- **Fehler bei der dynamischen Variablenuflösung:** Variablen, die nicht aufgelöst werden können, werden durch leere Zeichenketten ersetzt.
- **Probleme beim SCTE-35 UPID-Parsing:** Probleme mit der UPID-Verarbeitung der Segmentierung können zu Fehlern bei der Auflösung von Sitzungsvariablen führen.

Die Auflösung der Sitzungsvariablen wird überprüft

Gehen Sie wie folgt vor, um zu überprüfen, ob Ihre Sitzungsvariablen korrekt aufgelöst wurden:

1. Aktivieren Sie die Debug-Protokollierung für Ihre Konfiguration MediaTailor
2. Suchen Sie in der MediaTailor/AdDecisionServerInteractions Protokollgruppe nach der eigentlichen ADS-Anfrage URLs
3. Stellen Sie sicher, dass alle Variablen in der Vorlagen-URL durch die entsprechenden Werte ersetzt wurden
4. Suchen Sie nach Variablen, die durch leere Zeichenfolgen ersetzt wurden, was auf Auflösungsfehler hinweisen kann

Fehlerbehebung beim SCTE-35 UPID-Parsen

Probleme mit der UPID-Verarbeitung der SCTE-35-Segmentierung können zu Problemen mit Sitzungsvariablen führen:

- **Formatanforderungen:** UPID muss den Wert 12 haben und folgende Werte enthalten, damit UPID `segmentation_upid_type` ordnungsgemäß verarbeitet werden kann `format_identifizier`

- **Parseregeln:** Die dekodierte UPID kann Doppelpunkttrennzeichen für mehrere Werte enthalten. Die Anzahl der Vorlagenvariablen und der dekodierten UPID-Token muss gleich sein
- **Ungültige Formate:** Vermeiden Sie doppelte Doppelpunkte ohne Werte (z. B. `::` oder `:46175218:::4053`), da diese zu Fehlern beim Parsen führen

MediaTailor und das Überspringen von Überwachung und Warnmeldungen

Proaktive Überwachung hilft Ihnen dabei, Probleme beim Überspringen von Werbeanzeigen zu erkennen und zu lösen, bevor sie sich erheblich auf Ihren Umsatz auswirken. AWS Elemental MediaTailor bietet umfassende Metriken und Protokollierungsfunktionen, die eine effektive Überwachung der Leistung der Anzeigeneinblendung ermöglichen. In diesem Leitfaden wird erklärt, wie Sie eine effektive Überwachung für Probleme beim Überspringen von Anzeigen einrichten.

Wichtige CloudWatch Kennzahlen, die es zu überwachen gilt

Richten Sie CloudWatch Alarme für diese wichtigen MediaTailor Kennzahlen ein:

- `AdDecisionServer.Ads.Skipped`- Anzahl der übersprungenen Anzeigen
- `AdDecisionServer.Timeouts`- Anzahl der ADS-Timeouts
- `Avail.FilledDuration`- Dauer der aufgefüllten Werbeunterbrechungen
- `Avail.SlateOnly`- Anzahl der Werbeunterbrechungen, die nur mit Slate gefüllt wurden

Erweiterte CloudWatch Logs Insights-Abfragen

Verwenden Sie diese speziellen Abfragen für eine detaillierte Problembehandlung:

Umfassende Sitzungsanalyse

Für eine detaillierte Analyse des Verhaltens bei der Anzeigenschaltung für eine bestimmte Sitzung:

```
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
```

Kreativ finden IDs

Um Creative anhand IDs von FILLED_AVAIL-Ereignissen zu identifizieren:

```
fields @timestamp, sessionId, eventType
| filter sessionId like /sessionId/ and eventType!='BEACON_FIRED'
| sort @timestamp desc
```

Note

Ersetzen Sie es sessionId durch die tatsächliche Sitzungs-ID, die Sie untersuchen.

MediaTailor Bewährte Methoden zur Vorbeugung gegen das Überspringen von Anzeigen

Durch die Implementierung dieser bewährten Methoden können Probleme beim Überspringen von Anzeigen vermieden werden, bevor sie auftreten, wodurch eine bessere Leistung bei der Anzeigeneinblendung und ein besserer Umsatzschutz gewährleistet werden. AWS Elemental MediaTailor Mit diesen proaktiven Maßnahmen werden die häufigsten Ursachen für das Überspringen von Werbeanzeigen behoben und eine konsistente Anzeigenbereitstellung gewährleistet.

Proaktive Maßnahmen

- Implementieren Sie den Vorabruf MediaTailor von Anzeigen: Verwenden Sie die Prefetch-Funktion, um sicherzustellen, dass Anzeigen vor der Wiedergabe transkodiert werden. [the section called “Anzeigen vorab abrufen”](#) Einzelheiten zur Implementierung finden Sie unter
- Sorgen Sie für konsistente IDs Werbeanzeigen: Stellen Sie sicher, dass Ihr Anzeigenentscheidungsserver in allen Sitzungen konsistente Werbemittel IDs für dieselben Anzeigeninhalte verwendet
- Achten Sie auf die richtige Formatierung der Dauer: Verwenden Sie Ganzzahlwerte für die EXT-X-CUE-OUT Dauerparameter anstelle des ISO 8601-Formats
- VOD-Optimierung konfigurieren: Stellen Sie maxConcurrentAds Anfragen für VOD-Streams mit mehreren Werbeunterbrechungen ein, um die Serverlast zu reduzieren

- Optimieren Sie die ADS-Leistung: Konfigurieren Sie Ihren Ad Decision Server so, dass er schnell reagiert und hohe Besucherzahlen bewältigt

Richtlinien für die Implementierung

- Implementieren Sie die richtige Fehlerbehandlung: Konfigurieren Sie Slate-Inhalte so, dass Werbeunterbrechungen aufgefüllt werden, wenn Anzeigen nicht eingefügt werden können
- Testen Sie gründlich: Überprüfen Sie Ihren Arbeitsablauf beim Einfügen von Anzeigen auf verschiedenen Geräten und unter verschiedenen Netzwerkbedingungen
- Implementieren Sie Fallback-Strategien: Konfigurieren Sie Ersatz-Anzeigenquellen oder Standardanzeigen für den Fall, dass primäre Anzeigenquellen ausfallen
- Transcodierungsmuster überwachen: Überwachen Sie die CloudWatch Protokolle auf Effizienz bei der Transcodierung und wenden Sie sich an den [AWS-Support](#), wenn Sie Muster bemerken, die auf Transcodierungsprobleme hinweisen

MediaTailor Referenzleitfaden zum Überspringen von Anzeigen

Dieser Abschnitt enthält umfassende Referenzinformationen zu Gründen für das Überspringen von Anzeigen sowie Links zu entsprechenden Dokumentationen für AWS Elemental MediaTailor. In diesem Referenzhandbuch erfahren Sie mehr über die spezifischen Bedeutungen der verschiedenen Gründe für das Auslassen von Inhalten und finden weitere Ressourcen zur Problembehebung.

Vollständige Referenz zu den Gründen für das Überspringen von Anzeigen

MediaTailor protokolliert bestimmte Gründe, warum Anzeigen übersprungen werden, in der FILLED_AVAIL Ereignisprotokollnachricht der MediaTailor/AdDecisionServerInteractions Protokollgruppe.

Vollständige Gründe für das Überspringen von Anzeigen

Grund überspringen	Description
NEW_CREATIVE	Die Anzeige wurde noch nicht transkodiert. Dies ist der Fall, wenn Sie MediaTailor auf ein neues Werbematerial stoßen, das vor dem Einfügen transkodiert werden muss.

Grund überspringen	Description
PROFILE_NOT_FOUND	Das mit der Sitzungskonfiguration verknüpfte MediaConvert Transcodierungsprofil ist nicht vorhanden, wodurch die Anzeigenvorbereitung verhindert wird.
TRANSCODE_ERROR	Beim Transcodieren von Anzeigen ist ein Fehler aufgetreten und er konnte nicht abgeschlossen werden.
TRANSCODE_IN_PROGRESS	Die Transcodierung der Anzeige ist noch im Gange und kann noch nicht eingefügt werden.
INTERNAL_ERROR	Bei der Bearbeitung der Anzeige ist ein interner MediaTailor Fehler aufgetreten, der das Einfügen verhindert hat.
AVAIL_DURATION_EXCEEDED	Die Anzeige passt nicht in die verbleibende Dauer der Werbepause.
LEFTOVER_AVAIL_EXCEEDED_THRESHOLD	Die Gesamtdauer aller Anzeigen, die hätten eingefügt werden können, entspricht nicht dem für die Sitzung konfigurierten Schwellenwert für die Personalisierung.
VAST_PARSING_ERROR	Die VAST-Antwort vom Ad Decision Server enthält Fehler oder ist falsch formatiert.
ADS_TIMEOUT	Der Anzeigenentscheidungsserver hat innerhalb des konfigurierten Timeout-Zeitraums nicht geantwortet.
MEDIA_FILE_UNAVAILABLE	Auf die in der VAST-Antwort angegebenen Werbemediendateien kann nicht zugegriffen werden.
SESSION_INITIALIZATION_FAILED	Die MediaTailor Sitzung konnte nicht ordnungsgemäß initialisiert werden, was häufig auf falsche Sitzungsvariablen zurückzuführen ist.
EARLY_CUE_IN	Die Werbeunterbrechung wurde aufgrund eines frühen Hinweisignals früher als erwartet beendet, sodass die Anzeige nicht vollständig eingefügt werden konnte.

Grund überspringen	Description
NO_VARIANT_MATCH	Für das Werbemittel gibt es keine Variante, die den Kodierungsparametern des Content-Streams (Bitrate, Auflösung, Codec) entspricht.
NO_MODEL_CREATIVE_MATCH	Das Werbematerial entspricht nicht den erwarteten Modell- oder Formatanforderungen für die aktuelle Wiedergabekonfiguration.
REJECTED_REPLICA_VAST	Die VAST-Antwort wurde aufgrund von Richtlinien zur Erkennung von Replikaten oder doppelten Inhalten abgelehnt.
INVALID_VAST_WRAPPER_AD	Die VAST-Wrapper-Anzeige enthält ungültige oder falsch formatierte Wrapper-Elemente, die eine erfolgreiche Anzeigeneinblendung verhindern.
IMPORT_ERROR	Während des Anzeigenimportvorgangs ist ein Fehler aufgetreten, sodass die Anzeige nicht für das Einfügen verarbeitet werden konnte.
IMPORT_IN_PROGRESS	Der Anzeigenimportvorgang ist derzeit im Gange und noch nicht abgeschlossen.

Zugehörige Ressourcen

Weitere Informationen zur Behebung von Problemen beim Überspringen von Anzeigen finden Sie in den folgenden verwandten Themen:

- [the section called “Dynamische Anzeigenvariablen”](#)— Umfassender Leitfaden zu dynamischen Anzeigenvariablen in MediaTailor
- [the section called “Anzeigen vorab abrufen”](#)- So implementieren Sie das Prefetching von Anzeigen, um zu verhindern, dass bei der Transcodierung übersprungen wird
- [the section called “Überwachung mit CloudWatch Metriken”](#)- MediaTailor CloudWatch Überwachung mit Metriken
- [the section called “Anzeigen von Protokollen”](#)- Wie kann man MediaTailor Logs ansehen und analysieren
- [the section called “Fehlerbehebung beim Ereignisablauf”](#)- Grundlegendes zum Ablauf der Anzeigeneinblendung

AWS Elemental MediaTailor Ressourcen

Die folgende Tabelle enthält verwandte Ressourcen, die für die Arbeit mit AWS Elemental MediaTailor nützlich sind.

Ressource	Description
SCTE-Standard: SCTE 35	Das SCTE-Standarddokument für SCTE35
Kurse und Workshops	Links zu rollen- und Spezialkursen sowie zu Übungen zum Selbststudium, mit denen Sie Ihre AWS Fähigkeiten verbessern und praktische Erfahrungen sammeln können.
AWS Tools für Entwickler	Links zu Entwicklertools SDKs, IDE-Toolkits und Befehlszeilentools für die Entwicklung und Verwaltung von AWS Anwendungen.
AWS Whitepapers	Links zu einer umfassenden Liste von technischen AWS Whitepapers zu Themen wie Architektur, Sicherheit und Wirtschaft, die von Solutions Architects oder anderen technischen Experten verfasst wurden. AWS
AWS Unterstützungszentrum	Der Hub für die Erstellung und Verwaltung Ihrer AWS Support-Fälle. Enthält auch Links zu anderen hilfreichen Ressourcen wie Foren, technischen Informationen FAQs, Servicestatus und AWS Trusted Advisor.
AWS Support	Die wichtigste Webseite für Informationen über AWS Support, einen Support-Kanal mit schnellen Reaktionszeiten one-on-one, der Sie beim Erstellen und Ausführen von Anwendungen in der Cloud unterstützt.

Ressource	Description
Kontakt	Eine zentrale Anlaufstelle für Anfragen zu AWS Rechnungen, Konten, Ereignissen, Missbrauch und anderen Problemen.
AWS Nutzungsbedingungen der Website	Detaillierte Informationen zu unseren Copyright - und Markenbestimmungen, Ihrem Konto, den Lizenzen und anderen Themen.

Dokumentenverlauf für AWS Elemental MediaTailor

In der folgenden Tabelle werden wichtige Änderungen an dieser Dokumentation beschrieben.

Änderung	Beschreibung	Datum
TPS-Traffic-Shaping hinzugefügt	Die Abschnitte wurden aktualisiert und enthalten nun Informationen zur Beschränkung von Anfragen an das ADS auf Basis von TPS.	2. Oktober 2025
Hinzugefügt prefetchScheduleName	Es wurden Informationen zu prefetchScheduleName den ADS-Protokollen hinzugefügt.	24. September 2025
Umstrukturierter Leitfaden zur Fehlerbehebung beim Überspringen von Anzeigen	Der Leitfaden zur Fehlerbehebung wurde neu strukturiert, sodass sich jeder Abschnitt auf eine bestimmte Aufgabe konzentriert, die erledigt werden muss.	26. August 2025
Dokumentation zur Handhabung von Abfrageparametern hinzugefügt	Es wurde ein Abschnitt hinzugefügt, in dem erklärt wird, wie Abfrageparameter für ursprungsspezifische Funktionen wie die zeitversetzte Anzeige mit MediaTailor behandelt werden. MediaPackage	26. August 2025
SGAI VOD-Konfigurationsleitfaden	Konfigurationsanleitung für die servergesteuerte Anzeigeneinfügung mit	7. August 2025

Inhalten hinzugefügt. video-on-demand

[Integrierte Anleitung zur Fehlerbehebung bei Parametern](#)

Es wurde ein umfassender Abschnitt zur Fehlerbehebung hinzugefügt, der Fehler bei der Zeichenbeschränkung, Probleme mit der Längenbeschränkung, Probleme mit Konfigurationsalias, Debugging des Parameterverarbeitungsflusses und bewährte Sicherheitmethoden behandelt.

4. August 2025

[Verbesserte Dokumentation zur Parameterformatierung](#)

Es wurden umfassende Zeichenbeschränkungen, Unterstützung für URL-Kodierung und Längenbeschränkungen für Manifest-Abfrageparameter und ADS-Parameter hinzugefügt.

1. August 2025

[Verbesserte Parameterdokumentation mit umfassender Referenz](#)

Umfassende Zeichenbeschränkungen für Parameter, Längenbeschränkungen, Unterstützung für URL-Kodierung, Dokumentation zu Konfigurationsaliasen und Methoden zur Sitzungsinitialisierung hinzugefügt, die auf der AWS Guide-Analyse basieren.

1. August 2025

[Verbesserte Dokumentation der Manifest-Abfrageparameter](#)

Umfassendes Verhalten bei der Parameterplatzierung, Zeicheneinschränkungen, Unterstützung für URL-Kodierung und protokollspezifische Unterschiede auf der Grundlage der AWS Guide-Analyse hinzugefügt.

1. August 2025

[Verbesserte Dokumentation zu Domänenvariablen](#)

Umfassende Dokumentation zu Konfigurationsaliasen, unterstützte Felder und Parameterregeln auf Domänenebene hinzugefügt, die auf der AWS Guide-Analyse basieren.

1. August 2025

[Es wurden Anleitungen zur Fehlerbehebung bei Event-Abläufen hinzugefügt](#)

Neuer Abschnitt mit praktischen Anleitungen zur Verwendung der MediaTailor Ereignisanalyse zur Diagnose und Lösung von Problemen bei der Anzeigeneinfügung.

31. Juli 2025

[Dokumentation zum Ablauf der Anzeigeneinfügung hinzugefügt](#)

Neuer Abschnitt, in dem der chronologische Ablauf der MediaTailor Ereignisse während der Personalisierung von Manifesten erläutert wird, um Benutzern das Verständnis des Workflows zu erleichtern.

23. Juli 2025

Verbesserte Dokumentation zum clientseitigen Tracking-Workflow	Es wurde eine umfassende Dokumentation zum clientseitigen Berichtsworkflow mit detailliertem Sequenzdiagramm und Prozessklärung hinzugefügt. step-by-step	16. Juli 2025
Dokumentation zur CDN-Integration	Es wurden umfassendere Informationen zum Einrichten eines CDN mit hinzugefügt. MediaTailor	30. Juni 2025
CDN-Integration für Channel Assembly hinzugefügt	Es wurde eine Dokumentation zur Integration eines CDN in Channel Assembly hinzugefügt, um die Bereitstellung von Inhalten zu optimieren.	16. Mai 2025
Typen von HLS-Wiedergabelisten	Definitionen für Multivariant- und Medien-Playlisten von HLS wurden hinzugefügt.	9. Mai 2025
DASH-Manifesttypen	Definitionen für Komponenten des DASH-Manifests wurden hinzugefügt.	9. Mai 2025
Informationen zum HLS-Playlist-Typ hinzugefügt	Es wurden Erläuterungen zu Mehrvarianten- und Varianten-Playlisten hinzugefügt, um den Kontext für die Beispiele zu verbessern.	9. Mai 2025
Informationen zum Typ des DASH-Manifests hinzugefügt	Es wurden Erläuterungen zur MPD-Struktur und zu den Komponenten hinzugefügt, um den Kontext für die Beispiele bereitzustellen.	9. Mai 2025

Wiederkehrender Prefetch hinzugefügt	Die Abschnitte wurden aktualisiert und enthalten nun Informationen zu wiederkehrenden Prefetch-Zeitplänen.	28. April 2025
Es wurden konfigurationsspezifische Informationen zur Protokollfilterung hinzugefügt	MediaTailor Es wurden Informationen zum Filtern von Protokollen hinzugefügt, die für eine Konfiguration ausgegeben werden.	31. März 2025
Informationen zu verkauften Protokollen hinzugefügt	Es wurde ein neuer Abschnitt über die Verwendung von versendeten Protokollen für die Veröffentlichung von Protokollen hinzugefügt, die ausgegeben.. MediaTailor	4. Februar 2025
Übersicht über vorkonditionierte Anzeigen hinzugefügt	Es wurde ein neuer Abschnitt hinzugefügt, in dem beschrieben wird, wie die Anzeigeneinblendung MediaTailor verwaltet wird, wenn Sie vorkonditionierte Anzeigen verwenden.	30. Januar 2025
Anzeigenkonditionierung hinzugefügt	Die Einstellung zur Konditionierung von Streaming-Mediendateien wurde hinzugefügt.	30. Januar 2025
Leitfaden zur Live-Konfiguration von SGAI	Konfigurationsanleitung für die servergesteuerte Anzeigeneinfügung mit Live-Inhalten hinzugefügt.	15. Januar 2025

Protokolltypen wurden hinzugefügt	Es wurde ein neuer Abschnitt hinzugefügt, der die Protokolltypen auflistet, die MediaTailor ausgegeben werden.	15. Januar 2025
Informationen zur Protokollfilterung wurden hinzugefügt	MediaTailor Es wurde ein neuer Abschnitt über das Filtern von ausgegebenen Protokollen hinzugefügt.	15. Januar 2025
Unterstützte Formatierung von Abfrageparametern hinzugefügt	Es wurden Abschnitte für die Formatierung von Manifest-Abfrageparametern und ADS-Abfrageparametern hinzugefügt.	2. Januar 2025
Informationen zur Integration für Google Ad Manager	Es wurde ein Abschnitt zur Integration MediaTailor mit Google Ad Manager von der Server- und Clientseite hinzugefügt.	25. November 2024
Variable aktualisiert	Geändert breakabilityStartTime zuavailabilityStartTime .	6. Mai 2024
Details zur Personalisierung hinzugefügt	Das Insertion Mode Personalisierungsdetail wurde hinzugefügt.	6. Mai 2024
Regeln des Programms	Neue Inhalte zu den Programmregeln hinzugefügt.	25. April 2024
Der Inhalt zum Erstellen von Kanälen wurde aktualisiert	Es wurden Informationen zu Programmregeln beim Erstellen von Kanälen hinzugefügt.	20. April 2024

Der Inhalt zum Hinzufügen von Programmen wurde aktualisiert	Es wurden Informationen zu Programmregeln beim Hinzufügen von Programmen hinzugefügt.	20. April 2024
Workflow-Monitor	Analysieren Sie AWS Mediendienste und erstellen Sie Signalkarten, Visualisierungen des Medienworkflows, zwischen diesen Diensten. Verwenden Sie die Signalzuordnungen, um mithilfe CloudWatch von, und Überwachungsalarme und -benachrichtigungen zu generieren. EventBridge CloudFormation	11. April 2024
AlternateMedia und das As Run Log	Es wurde ein Hinweis hinzugefügt, wie Alternate Media sich das As Run Log auswirkt.	28. Februar 2024
Die Manifest-Einstellungen wurden aktualisiert	Je nach ausgewähltem Anzeigen-Markup-Typ wurden Informationen zu Passthrough-Tags hinzugefügt.	28. November 2023
SCTE-35-Nachrichten für Werbeunterbrechungen	Es wurden Informationen zu eingefügten SCTE-35-Tags für und zu Anzeigenmarkup-Typen hinzugefügt. <code>Daterange</code> . <code>Scte35 Enhanced</code>	28. November 2023

Schlüssel-Wert-Paare für den Markup-Typ „Werbung“ Enhanced Scte35	Es wurden Informationen zum MediaTailor Umgang mit übermittelten Schlüssel-Wert-Paaren für den Markup-Typ „Werbung“ hinzugefügt. Enhanced Scte35	28. November 2023
VOD-Quellen und Möglichkeiten	MediaTailor kann jetzt automatisch Werbemöglichkeiten in VOD-Quellen erkennen.	06. Oktober 2023
Neuer SigV4-Authentifizierungstyp Autodetect	MediaTailor unterstützt jetzt den AUTODETECT_SIGV4 Zugriffstyp.	18. August 2023
Der clientseitige Tracking-Inhalt wurde aktualisiert	Der Inhalt der clientseitigen Nachverfolgung wurde aktualisiert und enthält nun zusätzliche Informationen.	12. August 2023
Einrichtung MediaTailor und MediaPackage Workflows zur Verwendung von Live-Quellen	Es wurden Informationen zur Einrichtung, zu allgemeinen Anforderungen und zum Verhalten bei MediaTailor der Verwendung von Live-Quellen in AWS Elemental MediaPackage Workflows hinzugefügt.	24. Mai 2023
Dokumentation zu den Manifest-Abfrageparametern	Es wurde ein Abschnitt hinzugefügt, der die Manifest-Abfrageparameter beschreibt.	26. April 2023
Dokumentation zu Overlay-Anzeigen	Es wurde ein Abschnitt hinzugefügt, der Overlay-Anzeigen beschreibt.	24. April 2023

Dekoration der Anzeigen-ID in der Dokumentation zu den Manifesten	Es wurde ein Abschnitt hinzugefügt, der die Dekoration von Anzeigen-IDs in Manifesten beschreibt.	24. April 2023
Der AFTER_LIVE_EDGE-Unterdrückungsmodus wurde hinzugefügt	AFTER_LIVE_EDGE Der Werbeunterdrückungsmodus ist jetzt zusätzlich zum Modus verfügbar. BEFORE_LIVE_EDGE	21. Februar 2023
Neues As-Run-Protokoll	Neues Thema im As Run-Protokoll.	19. Januar 2023
Updates zu den bewährten Methoden für IAM	Aktualisierung des Leitfadens zur Ausrichtung an bewährten IAM-Methoden. Weitere Informationen finden Sie unter Bewährte Sicherheitsmethoden in IAM .	27. Dezember 2022
Updates zu bewährten IAM-Praktiken	Aktualisierung des Leitfadens zur Ausrichtung an bewährten IAM-Methoden. Weitere Informationen finden Sie unter Bewährte Sicherheitsmethoden in IAM .	27. Dezember 2022
Neue private UPID-Datenvariablen zur Segmentierung	MediaTailor unterstützt jetzt zusätzliche Sitzungsvariablen <code>segmentation_upid.private_data.{index}</code> , die durch Doppelpunkte getrennte private Daten aus der Segmentierung UPID Type 12 (MPU) trennen.	22. Dezember 2022

Inhalt der Kontingente aktualisiert	Die Informationen zu den Kontingenten wurden aktualisiert und neu organisiert.	13. September 2022
Kontingente für Kanalzusammenstellungen wurden hinzugefügt und korrigiert	Es wurden Kontingente für Live-Quellen, Konfigurationen für die Segmentzuordnung, Manifestanfragen und Kanaltransaktionen im MediaTailor Channel Assembly Service hinzugefügt.	13. September 2022
Kontingente für Channel-Assembly hinzugefügt und korrigiert	Es wurden Kontingente für Protokollierung, Kanal, Live-Quellen, Programme, Quelladressen und Kanalrichtlinien im MediaTailor Channel Assembly Service hinzugefügt.	11. September 2022
Neue Tabelle mit Warnmeldungen zur Kanalbaugruppe	Sie können jetzt eine Tabelle sehen, in der die Warnmeldungen zur Kanalbaugruppe erklärt werden.	01. September 2022
Neue CloudWatch Amazon-Metriken	Neue CloudWatch Metriken hinzugefügt.	26. Juni 2022
Anzeige ruft Themen auf	Link zur Prefetch-Unterstützung für VAST-Antworten hinzugefügt.	25. Mai 2022

[Neue ADS-Anforderungsvariablen](#)

MediaTailor unterstützt jetzt diese zusätzlichen SCTE-35-Variablen in ADS-Anfragen: `scte.segmentation_type_id`, `scte.available_expected`, `scte.delivery_not_restricted_flag`, `scte.segment_num`, `scte.sub_segment_num`, `scte.segments_expected`, `scte.sub_segments_expected`, `scte.device_restrictions`, `scte.no_regional_blackout_flag`, `scte.archive_allowed_flag` und `scte.segmentation_event_id`.

27. April 2022

[Neues Thema zu verwalteten IAM-Richtlinien](#)

Es wurden zwei neue verwaltete Richtlinien für MediaTailor hinzugefügt.

24. November 2021

[Neue von AWSElementalMediaTailorReadOnly verwaltete Richtlinie](#)

Es wurde eine neue AWS verwaltete Richtlinie hinzugefügt, die Berechtigungen gewährt, die den schreibgeschützten Zugriff auf Ressourcen ermöglichen MediaTailor .

10. November 2021

<u>Neue von AWSElementalMediaTailorFullAccess verwaltete Richtlinie</u>	Es wurde eine neue AWS verwaltete Richtlinie hinzugefügt, die vollen Zugriff auf Ressourcen ermöglicht MediaTailor.	10. November 2021
<u>Neues verwirrtes stellvertretendes Thema</u>	Es wurde ein Thema hinzugefügt, in dem erklärt wird, wie das Problem mit dem verwirrten Stellvertreter verhindert werden kann.	4. November 2021
<u>Das Anzeigenthema wird vorab abgerufen</u>	MediaTailor kann jetzt Anzeigen für Werbeunterbrechungen vorab abrufen, bevor sie auftreten.	12. Oktober 2021
<u>Es wurden Konfigurationseinstellungen für die Protokollierung von Wiedergabekonfigurationen hinzugefügt</u>	Verwenden Sie die Konfigurationseinstellungen für die Protokollierung, um die Einstellungen für die Wiedergabekonfigurationsprotokolle zu steuern.	28. September 2021
<u>SCTE-35-Meldungen für Werbeunterbrechungen</u>	Es wurden Informationen zur Verwendung von <code>segmentation_descriptor</code> Nachrichten zusammen mit <code>time_signal</code> Nachrichten hinzugefügt.	1. September 2021
<u>Neuer linearer Wiedergabemodus</u>	Ein neuer linearer Wiedergabemodus wurde hinzugefügt.	1. September 2021

<u>Neuer absoluter Übergangstyp</u>	Es wurde Unterstützung für absolute Übergangstypen hinzugefügt, mit denen Sie eine Wanduhr-Startzeit für Ihr Programm auf linearen Kanälen festlegen können.	1. September 2021
<u>Neues Thema „Benachrichtigungen zur Kanalmontage“</u>	Sie können jetzt Ihre Ressourcen für die Kanalzusammenstellung mithilfe von MediaTailor Benachrichtigungen überwachen. Wenn ein Problem oder ein potenzielles Problem mit Ihren Ressourcen für die Kanalmontage auftritt, werden Warnmeldungen MediaTailor generiert.	14. Juli 2021
<u>Die Kontingente für Kanalzusammenstellungen für ausgehende Kanalanfragen wurden korrigiert</u>	Die Kontingente für ausgehende Kanalanfragen im MediaTailor Channel-Assembly-Service wurden korrigiert.	29. Juni 2021
<u>Neuer Authentifizierungstyp für den Quellstandort</u>	MediaTailor unterstützt jetzt die Secrets Manager Manager-Zugriffstoken-Authentifizierung.	16. Juni 2021
<u>Neue Tier-Informationen</u>	Es wurden Informationen zu den Modi und Quelltypen hinzugefügt, die jede Stufe unterstützt.	13. Juni 2021

Neue Informationen zum Quellentyp	Für Standardkanäle wurden Informationen über die Art der Quelle hinzugefügt, die das Programm wiedergibt.	13. Juni 2021
Neue Dokumentation MediaTailor zu Live-Quellen	Eine Live-Quelle stellt einen einzelnen Live-Stream dar, den Sie zu Ihrem Quellstandort hinzufügen. Nachdem du deinen Kanal erstellt hast, kannst du Live-Quellen zu deinem Quellstandort hinzufügen und jede Live-Quelle einem Programm zuordnen.	13. Juni 2021
Support für zusätzliche UPID-Typen	MediaTailor unterstützt jetzt die UPID-Typen ADS-Information (0xE) und benutzerdefinierte Segmentierung (0x1).	15. April 2021
Neue dynamische UPID-Variablen für die Segmentierung	Es gibt drei neue dynamische Variablen: <code>scte.segmentation_upid.assetId</code> , <code>scte.segmentation_upid.cueData.key</code> , und <code>scte.segmentation_upid.cueData.value</code> . Diese Variablen werden in Verbindung mit dem UPID-Typ der MPU-Segmentierung (0xC) für Podbuster-Workflows verwendet.	15. April 2021
Neue Servicebeschreibung für die Kanalmontage	Es wurden Informationen zum neuen Kanalmontage-Service hinzugefügt.	11. März 2021

Neue Servicedokumentation für die MediaTailor Kanalmontage	Channel Assembly ist ein neuer Dienst, der nur auf Manifest beschränkt ist und es Ihnen ermöglicht, lineare Streaming-Kanäle unter Verwendung Ihrer vorhandenen Video-on-Demand-Inhalte (VOD) zu erstellen.	11. März 2021
Kontingente für die Kanalarmontage hinzugefügt	Kontingente für den neuen MediaTailor Channel-Assembly-Service hinzugefügt.	11. März 2021
Neue Bedingungen für die Kanalarmontage	Es wurden Bedingungen hinzugefügt, die dem neuen Service zur Kanalarmontage entsprechen.	10. März 2021
Unterstützung für die Kennzeichnung von Kanalbaugruppen	Unterstützung für das Taggen von Ressourcen zur Kanalarmontage wurde hinzugefügt. AWS Elemental MediaTailor Kanäle SourceLocations, und VodSources unterstützen das Taggen.	9. März 2021
Neues Thema zu dynamischen Variablen	MediaTailor unterstützt jetzt dynamische Domänenvariablen.	25. Februar 2021
Optionale Alias-Einstellungen für die Konfiguration wurden hinzugefügt	Verwenden Sie Konfigurationsalias zusammen mit Domänenvariablen, um Domänen während der Sitzungsinitialisierung dynamisch zu konfigurieren.	25. Februar 2021

<u>Neue scte.segmentation_upid dynamische Anzeigenvariable</u>	Unterstützung für die dynamische Anzeigenvariable <code>scte.segmentation_upid</code> Sitzungsdaten wurde hinzugefügt.	5. Dezember 2020
<u>Neues Passthrough-Thema zur Anzeigenmarkierung</u>	Passthrough für Werbemarkierungen ist jetzt für HLS-Manifeste verfügbar.	29. Oktober 2020
<u>Die erweiterten Konfigurationseinstellungen wurden aktualisiert</u>	Ad Marker Passthrough ist eine neue erweiterte Einstellung für die Wiedergabekonfiguration.	14. Oktober 2020
<u>Neuer Debug-Log-Modus</u>	Neues Thema zum DEBUG-Logmodus.	14. August 2020
<u>Klarstellung zum EXT-X-CUE-OUT Dauerattribut für Stoßstangen</u>	Die Anforderungen an Bumpers wurden aktualisiert, sodass für HLS das <code>duration</code> Attribut für jedes Tag erforderlich ist. EXT-X-CUE-OUT	5. August 2020
<u>Parameter für den Berichtsmodus der Sitzungsinitialisierung</u>	Unterstützung für den <code>reportingMode</code> Parameter in Anfragen zur Sitzungsinitialisierung hinzugefügt. Sie können jetzt angeben, ob bei der MediaTailor Initialisierung einer Sitzung clientseitiges oder serverseitiges Ad-Tracking durchgeführt wird.	30. Juli 2020
<u>Neues Bumper-Thema</u>	Ein neues Bumper-Thema wurde hinzugefügt	27. Juli 2020

<u>Ad Suppression ist für DASH verfügbar</u>	Für DASH ist jetzt Werbeunterdrückung verfügbar. Die „Nur HLS“-Einschränkung wurde aus dem Thema der Werbeunterdrückung entfernt.	3. Juni 2020
<u>Aktualisieren Sie konsolenspezifische Namen</u>	Die konsolenspezifischen Namen wurden aktualisiert, um eine neuere Version der Konsolen-Benutzeroberfläche wiederzugeben.	1. Mai 2020
<u>Neue avail.index dynamische Anzeigenvariable</u>	Hinzufügung der Unterstützung für die neue dynamische Anzeigenvariable für avail.index -Sitzungsdaten.	13. März 2020
<u>Neu AdVerifications und Extensions Elemente</u>	Für die clientseitige Berichterstellung werden die AdVerifications - und Extensions -Elemente unterstützt.	10. März 2020
<u>Konfiguration des Schwellenwerts für die Personalisierung</u>	Ab sofort wird die optionale Konfiguration einer Personalisierungsschwelle unterstützt.	14. Februar 2020
<u>DASH VOD manifestiert sich</u>	Es wurde Unterstützung für DASH-VOD (Video on Demand)-Manifeste vom Ursprungs-Server mit mehreren Ausgabezeiträumen hinzugefügt.	23. Dezember 2019
<u>Konsolenunterstützung für den Transcode-Profilnamen</u>	Beschreibung für Transcodierungsprofilnamen in die Konfiguration eingefügt.	23. Dezember 2019

Die Grenzwerttabellen wurden aktualisiert	Aktualisierte Limits für ADS-Weiterleitungen und -Timeouts .	18. Dezember 2019
Dokumentieren Sie Live-Pre-Roll-Verhaltensweisen	Es wurde ein Abschnitt zum Einfügen von Pre-Roll-Anzeigen hinzugefügt, in dem beschrieben wird, wie Live-Pre-Roll-Anzeigen mit funktionieren. AWS Elemental MediaTailor	26. November 2019
Support für Live-Pre-Roll-Anzeigen	Unterstützung für das Einfügen von Pre-Roll-Werbung am Anfang eines Live-Streams wurde hinzugefügt.	11. September 2019
Analysieren von ADS-Protokollen in Amazon CloudWatch Logs Insights	Es wurden Informationen zur Verwendung der AWS Elemental MediaTailor ADS-Logs und CloudWatch Logs Insights zur Analyse Ihrer MediaTailor Sitzungen hinzugefügt.	13. August 2019
Neues Kapitel bezüglich der Sicherheit	Es wurde ein Kapitel bezüglich der Sicherheit hinzugefügt, um die Abdeckung des Themas zu verbessern und zu standardisieren.	23. Mai 2019

<u>DASH-Manifeste für einzelne Perioden</u>	Es wurde Unterstützung für DASH-Manifeste mit einem einzigen Zeitraum vom Ursprungs-Server mit Manifest-Ausgabe mit mehreren Zeiträumen hinzugefügt.	4. April 2019
<u>Support für SCTE-35 UPIDs in der ADS-URL</u>	Es wurde Unterstützung für den Einschluss einer eindeutigen Programm-ID (UPID) in die URL des Ad Decision Server (Entscheidungsservers für Werbung, ADS) hinzugefügt. Dies ermöglicht dem ADS, zielgerichtete Werbung auf Programmebene in einem linearen Live-Stream bereitzustellen.	28. März 2019
<u>Die clientseitige Berichterstattung unterstützt Begleitanzeigen</u>	Für die clientseitige Berichterstattung enthält die Antwort auf die AWS Elemental MediaTailor Tracking-URL jetzt Metadaten für Begleitanzeigen.	28. März 2019
<u>Dokumentation zu HLS-Werbemarkern</u>	Es wurde ein Abschnitt zur Beschreibung unterstützter HLS-Werbemarkern hinzugefügt.	1. März 2019

Unterstützung für Tagging	Es wurde Unterstützung für das Tagging von Konfigurationsressourcen in AWS Elemental MediaTailor hinzugefügt. Tagging ermöglicht es Ihnen, Ihre AWS Ressourcen zu identifizieren und zu organisieren, den Zugriff darauf zu kontrollieren und Ihre AWS Kosten zu verfolgen.	14. Februar 2019
AWS CloudTrail Protokollierungsinformationen wurden hinzugefügt	Es wurde ein Thema CloudTrail zur Verwendung zum Protokollieren von Aktionen in der AWS Elemental MediaTailor API hinzugefügt.	11. Februar 2019
Abschnitt zu Wiedergabefehlern hinzugefügt	Es wurden Informationen über die Fehler hinzugefügt, die von MediaTailor während der Wiedergabe als Antwort auf Anforderungen von einem Player oder einem Netzwerk zur Bereitstellung von Inhalten (Content Delivery Network, CDN) zurückgegeben werden.	4. Februar 2019
Base64-codierte DASH-Binärdatei	Es wurde Unterstützung für die Bereitstellung von Splicing-Informationen in Manifesten im base64-kodierten binären Format innerhalb von <scte35:Signal> - und <scte35:Binary> -Markern hinzugefügt.	4. Januar 2019

<u>DASH-Zeitsignal</u>	Es wurde Unterstützung für die Bereitstellung von Splicing-Informationen in Manifesten innerhalb von <code><scte35:TimeSignal></code> -Markern hinzugefügt.	5. Dezember 2018
<u>Unterstützung für DASH-Standorte</u>	Unterstützung für das MPEG-DASH-Tag <code><Location></code> hinzugefügt.	4. Dezember 2018
<u>DASH-Unterstützung</u>	Unterstützung für MPEG-DASH-Manifeste hinzugefügt.	14. November 2018
<u>Die Limittabellen wurden aktualisiert</u>	Aktualisierte Limits für Konfigurationen und Manifest-Größe.	13. Oktober 2018
<u>Neue und aktualisierte Metriken</u>	Es wurden Metriken für den Ad Decision Server (Entscheidungsserver für Werbung, ADS) und Timeouts für den Ursprungs-Server hinzugefügt und die Fehlerdefinitionen des ADS und des Ursprungs-Servers so aktualisiert, dass Antworten mit Zeitüberschreitung eingeschlossen werden.	13. Oktober 2018
<u>Bessere Dokumentation für Anwendungsfälle zur serverseitigen und clientseitigen Anzeigeneinfügung</u>	Eingehendere Beschreibung und Beispiele, die auf die Verwendung von dynamischen Werbevariablen für serverseitige und clientseitige Werbeeinblendung eingehen.	1. Oktober 2018

Neue Regionen	Unterstützung für die Regionen PDX und FRA hinzugefügt.	18. Juli 2018
VAST/VPAID	Informationen über VAST und VPAID hinzugefügt.	16. März 2018
CloudWatch	Es wurden Informationen zu verfügbaren CloudWatch Metriken, Namespaces und Dimensionen hinzugefügt.	16. März 2018
Neue Regionen	Unterstützung für die Regionen Asien-Pazifik (Singapur), Asien-Pazifik (Sydney) und Asien-Pazifik (Tokio) hinzugefügt.	8. Februar 2018
Informationen zur IAM-Richtlinie	Spezifische IAM-Richtlinieninformationen für hinzugefügt. AWS Elemental MediaTailor Es wurden Anweisungen für die Erstellung von Nicht-Admin-Rollen mit eingeschränkten Berechtigungen hinzugefügt.	3. Januar 2018
Erste Veröffentlichung	Erste Version dieser Dokumentation.	27. November 2017

Note

- Die AWS Mediendienste sind nicht für die Verwendung mit Anwendungen oder in Situationen konzipiert oder vorgesehen, in denen eine ausfallsichere Leistung erforderlich ist, wie z. B. bei der Betriebssicherheit, bei Navigations- oder Kommunikationssystemen, bei der Flugsicherung oder bei lebenserhaltenden Maschinen, in denen die

Nichtverfügbarkeit, Unterbrechung oder der Ausfall der Dienste zu Tod, Personen-, Sach- oder Umweltschäden führen kann.

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