aws

用户指南

AWS Elemental MediaTailor

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

AWS Elemental MediaTailor: 用户指南

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

Amazon 的商标和商业外观不得用于任何非 Amazon 的商品或服务，也不得以任何可能引起客户混淆、贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其他商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

Table of Contents

什么是 AWS Elemental MediaTailor？ 1
源服务器要求 1
MediaTailor 概念 1
HLS 播放列表类型 2
DASH 清单类型 4
广告插入概念 9
渠道组装概念 10
广告插入的工作原理 10
广告插入事件流程 12
相关服务 14
正在访问 MediaTailor 14
定价 15
Regions 15
设置 16
注册参加 AWS 账户 16
入门 MediaTailor 17
MediaTailor 广告插入入门 17
先决条件 18
第 1 步：访问 MediaTailor 18
第 2 步：准备直播 18
第 3 步：配置 ADS 请求 URL 和查询参数 20
第 4 步：创建配置 21
第 5 步：测试配置 22
第 6 步：将播放请求发送至 MediaTailor 24
（可选）步骤 7：监控活动 24
步骤 8：清除 25
MediaTailor 频道汇编入门 26
先决条件 27
第 1 步：创建来源位置 27
第 2 步：将 VOD 源添加到您的源位置 28
第 3 步：创建频道 29
第 4 步：将节目添加到频道的时间表中 30
第 5 步 (optional)： MediaTailor 用于在您的直播中插入个性化广告 31
第 6 步：启动您的频道 32

第 7 步：测试您的频道 32
步骤 8：清除 32
插入广告 34
支持的音频和视频编解码器 35
了解广告插入行为 35
VOD 的广告拼接行为 35
直播广告拼接行为 38
服务器引导式广告插入概述 40
在播放配置中启用 40
创建服务器引导式会话 41
使用 SGAI 进行广告跟踪 42
引导式预取，带有明显的心跳声 42
SGAI 功能兼容性 43
SGAI 直播配置 45
SGAI VOD 配置 48
广告服务器集成要求 51
大量的要求 52
VPAID 要求 54
广告自助餐 55
示例 VAST 55
重要概念 56
优势 56
正在启用 57
工作原理 59
VAST 包装器行为 60
指标和监控 67
ADS 活动变更 68
另请参阅 69
配置管理 70
创建配置 70
查看配置详情 78
编辑配置设置 79
删除配置 79
整合内容源 79
输入源要求 80
集成 HLS 源代码 80

整合 MPEG-DASH 来源 91
使用 SigV4 保护源站交互 111
与谷歌的广告平台集成 115
Server-side 整合 116
Client-side 整合 117
优化广告填充率 118
使用广告中断抑制功能自定义广告中断行为 120
配置广告间隔抑制 120
保险杠广告插页 131
配置保险杠 131
片前广告插播 132
Slate 广告插入 133
配置名单 134
个性化阈值的工作原理 134
Slate 配置和 VPAID 136
广告转码 137
动态转码 137
自定义转码配置文件 139
预取广告 142
预取的工作原理 143
创建预回迁计划 146
TPS-based 流量整形 151
删除预回迁计划 152
预处理广告 152
预处理广告要求 153
预处理广告工作流程 155
盈利功能 156
函数如何融入工作 MediaTailor 流程 157
快速入门指南 158
生命周期钩子 161
函数类型和构成 168
创建和管理 181
jsonata 表达式参考 184
函数示例 195
故障排除和监控 198
限制 206

动态广告变量 210
会话变量 211
玩家变量 223
域变量 225
服务变量 225
配置别名 233
传递 ADS 参数 240
参数路由 244
MediaPackage 集成 245
会话行为 247
参数参考 248
参数疑难解答 250
别名故障排除 254
MediaTailor 清单查询参数 256
来源查询参数 257
MediaTailor字符限制 258
MediaTailorHLS 隐式会话 259
MediaTailor DASH 隐式会话 260
MediaTailor 显式会话初始化 262
MediaTailor特定于协议的行为 263
MediaTailor 内容分发网络集成 266
报告广告跟踪数据 267
Server-side 追踪 268
Client-side 追踪 281
叠加广告 354
使用叠加广告的先决条件 354
开始使用 355
日志和指标 368
中叠加广告的计费 MediaTailor 369
广告 ID 装饰 370
会话状态 371
清单和广告元数据插入 372
广告决策服务器 (ADS) 互动 396
客户端跟踪 API 397
创建线性组装流 399
渠道汇编工作流程类型 399

VOD-to-live 400
Live-to-live 402
VOD + Live-to-live （混合） 405
选择正确的工作流程 408
后续步骤 409
使用来源位置 409
创建源位置 409
为您的来源位置配置身份验证 411
使用 VOD 来源 419
使用实时信号源 422
使用软件包配置 426
清单缓存 426
使用通道 426
创建通道 427
将来源组与频道的输出配合使用 429
删除频道 429
添加程序 430
创建程序 430
定义受众群体和替代内容 436
生成针对特定受众的清单 438
插入广告和广告插播时间 439
设置广告插入 439
SCTE-35 广告中断消息 440
对频道的播放进行时移 446
使用 CDN 448
内容分发网络选择 449
规划 CDN 集成 450
估算流量需求 451
配置优化策略 452
自定义规划 453
设置监控和扩展 455
优化成本 456
测试您的实现 457
常见问题故障排除 459
设置 CDN 集成 460
CDN 集成组件和要求 461

CDN 集成的先决条件 18
CDN 集成设置步骤 462
MediaTailor CDN 集成所需的标头 462
设置 CDN 路由行为 464
设置 CDN 映射 467
安全最佳实践 469
使用 CDN 进行广告插入 471
你需要什么 471
开始前的准备工作 472
CDN 集成的好处 472
了解 CDN 架构 473
设置基本的广告插入 479
带频道组装的 SSAI 482
优化 CDN 性能 488
监控 CDN 的运行情况 490
对广告插入进行故障排除 CDNs 495
使用 CDN 进行频道组装 498
你需要什么 498
开始前的准备工作 499
CDN 集成的好处 500
了解 CDN 架构 501
基本设置 503
配置基地 URLs 504
实现广告插入 504
配置时移视图 507
监控 CDN 的运行情况 508
完整的优化指南 509
MediaPackage 内容分发网络集成 509
了解 MediaPackage 和 CDN 工作流程 509
先决条件 510
基本设置配置 510
集成验证 512
后续步骤 513
清单筛选 514
CDN 缓存 519
CDN 性能 522

CDN 集成疑难解答 527
CloudFront 集成 530
基本 CloudFront 设置 531
性能优化 536
Multi-Region 弹性 540
监控和排查 543
第三方 CDN 设置 547
先决条件 548
配置 CDN 缓存规则 549
验证您的 CDN 配置 551
优化 CDN 性能 552
解决第三方 CDN 问题 553
CDN 性能优化 555
缓存优化 555
请求路由优化 559
性能基准 559
高级优化 560
内容分发网络监控 563
基本 CDN 性能指标 563
设置监控工具 565
实施清单 566
相关主题 567
CDN 集成测试 567
测试先决条件 568
系统化测试方法 568
部署前清单 573
测试工具参考 574
对 CDN 集成进行故障排除 575
快速诊断清单 576
集成测试 577
Manifest 404 错误 581
舱单配送问题 582
区段交付问题 583
会话管理问题 585
广告中断时间问题 587
性能和延迟问题 588

设备行为不一致 594
疑难解答准备 594
特定于工作流程的疑难解答指南 597
CDN 集成日志分析参考 598
CDN 日志解释 598
错误代码参考 600
日志分析工具 602
MediaTailor 记录配置资源 602
CloudFormation 自动化 603
为什么使用 AWS CloudFormation 604
准备部署 605
部署 模板 607
使用已部署的资源 608
测试和验证您的部署 611
排查 部署问题 612
自定义模板 613
模板参考 616
生产 CloudFront配置 622
Three-origin 建筑 622
缓存行为的优先级和配置 623
获取 CDN 集成支持 629
何时升级到 Support 629
收集信息以获得支持 630
Support 资源 630
Support 案例最佳实践 631
安全性 633
数据保护 633
数据加密 634
身份和访问管理 634
受众 635
使用身份进行身份验证 635
使用策略管理访问 636
操作方法 AWS Elemental MediaTailor 与 IAM 一起使用 638
Identity-based 政策示例 642
Resource-based 政策示例 644
AWS 托管策略 645

使用服务关联角色 646
排除 身份和访问问题 649
合规性验证 652
恢复能力 652
基础结构安全性 652
Cross-service 困惑的副手预防 653
日志记录和监控 653
CloudWatch 警报 654
CloudTrail 日志 654
AWS Trusted Advisor 654
监控和标记 655
查看日志 655
ADS 日志 656
清单日志 709
转码日志 713
使用已售日志 716
将日志写入 CloudWatch 日志 719
控制广告插入会话日志的数量 727
筛选日志和事件 730
生成调试日志 732
使用 CloudWatch 指标进行监控 736
AWS Elemental MediaTailor CloudWatch 指标 737
AWS Elemental MediaTailor CloudWatch 维度 750
使用指标诊断陈旧清单 750
记录 API 调用 752
AWS Elemental MediaTailor 中的信息 CloudTrail 752
了解 AWS Elemental MediaTailor 日志文件条目 753
接收频道集合警报 755
对与清单相关的警报进行故障排除 759
查看警报 760
处理警报 761
标注资源 761
支持的资源 761
标签限制 761
管理标签 762
工作流监视器 762

工作流监视器的组件 763
受支持的服务 764
配置工作流监视器 764
使用工作流监视器 781
Dual-stack 端点 783
什么是双栈终端节点？ 783
支持的功能 783
Dual-stack 端点格式 783
使用双堆栈端点 784
AWS 管理控制台 784
AWS Command Line Interface 784
AWS SDK 784
注意事项 785
配额 786
广告插播配额 786
渠道组装配额 789
问题排查 793
事件流疑难解答 793
流程不完整 793
时机分析 795
常见问题 797
监控最佳实践 799
其他资源 800
播放疑难解答 801
客户端错误 801
服务器错误数 803
示例 804
信标重复数据删除 805
参数疑难解答 806
跳过广告疑难解答 806
症状 806
跳过理由 807
NEW_CREATIVE 跳过 807
跳过 ADS 超时 809
VAST 解析跳过 810
时长不匹配跳过 811

跳过会话变量 812
监控广告跳过 813
预防最佳实践 814
参考资料 815
MediaTailor 资源 818
文档历史记录 819
..... dcccxxxiv

什么是 AWS Elemental MediaTailor?

AWS Elemental MediaTailor 是一项在 AWS 云端运行的可扩展广告插入和渠道整合服务。借助 MediaTailor，您可以向观众投放有针对性的广告内容并创建线性流，同时在OTT (OTT) 视频应用程序中保持广播质量。 MediaTailor 广告插入支持 Apple HTTP 直播 (HLS) 和基于 HTTP 的 MPEG 动态自适应流媒体 (DASH)，用于视频点播 (VOD) 和直播工作流程。

AWS Elemental MediaTailor 与传统的广告跟踪系统相比，广告插入提供了重要的进步：广告的盈利效果更好，视频质量和分辨率更加一致，并且更易于在多平台环境中进行管理。 MediaTailor 允许所有 IP-connected 设备以与呈现其他内容相同的方式呈现广告，从而简化您的广告工作流程。该服务还提供广告视图的高级跟踪，可进一步促进内容盈利。

AWS Elemental MediaTailor 频道集合是一项仅限清单的服务，允许您使用现有的视频点播 (VOD) 内容创建线性流媒体频道。 MediaTailor 永远不要触摸你的内容片段，这些片段是直接从你的源服务器提供的。取而代之的是，从你的来源 MediaTailor获取清单，然后使用它们来组装一个引用底层内容段的实时滑动清单窗口。

MediaTailor 频道组合使您可以通过在直播中插入广告片段来轻松地通过频道获利，而无需使用标记对其进行 SCTE-35 限制。您可以将渠道组合与 MediaTailor 广告插播一起使用，也可以使用其他服务器端广告插入服务。

源服务器要求

AWS Elemental MediaTailor 对源服务器通信有特定要求：

- 支持的端口- MediaTailor 仅接受使用标准 HTTP 和 HTTPS 端口的起点：
  - 用于 HTTP 连接的端口 80
  - 用于 HTTPS 连接的端口 443

  MediaTailor 不支持源服务器通信的自定义端口。
- 协议要求-为了安全通信，某些源类型和身份验证方案 MediaTailor 需要使用 HTTPS。有关更多信息，请参阅 the section called “整合内容源”。

MediaTailor 概念

以下是整个《AWS Elemental MediaTailor 用户指南》中使用的概念的概述。

HLS 播放列表类型

HTTP Live Streaming (HLS) 使用两种主要类型的播放列表：多变播放列表和媒体播放列表。了解这些播放列表类型之间的区别对于配置和故障排除 MediaTailor 工作流程至关重要。

多变体播放列表

多变体播放列表是 HLS 直播工作流程中的顶级索引文件，它列出了内容的所有可用格式副本。它包含对媒体播放列表的引用，但本身不包含任何媒体片段。多变播放列表允许玩家根据网络条件、设备功能或用户偏好选择最合适的格式副本。

这种播放列表类型在各种情况下也被其他几个名称所熟知：

- 主播放列表（已弃用的术语）
- 主舱单（常用行业术语）
- 主要播放列表
- 主播放列表
- 索引文件
- Master M3U8

在 MediaTailor 工作流程中，多变体播放列表是播放请求的入口点，也是广告个性化的起点。

Example多变体播放列表示例

```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-INDEPENDENT-SEGMENTS
#EXT-X-STREAM-INF:BANDWIDTH=2665726,AVERAGE-
BANDWIDTH=2526299,RESOLUTION=960x540,CODECS="avc1.640029,mp4a.40.2",SUBTITLES="subtitles"
index_1.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=4786126,AVERAGE-
BANDWIDTH=4547599,RESOLUTION=1280x720,CODECS="avc1.640029,mp4a.40.2",SUBTITLES="subtitles"
index_2.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=8171787,AVERAGE-
BANDWIDTH=7768157,RESOLUTION=1920x1080,CODECS="avc1.640029,mp4a.40.2",SUBTITLES="subtitles"
index_3.m3u8
```

媒体播放列表

媒体播放列表包含特定内容再现（质量级别）的实际媒体片段信息。它包括播放单个格式副本所需的时间信息 URLs、片段和其他元数据。每个媒体播放列表代表分辨率、比特率和其他特征的不同组合。

这种播放列表类型也有其他几个名称：

- 变体播放列表
- 儿童清单（行业常用术语）
- 区块清单
- Media M3U8
- 格式副本播放列表

在 MediaTailor 工作流程中，媒体播放列表会进行个性化设置，以按适当顺序包括内容细分和广告细分。

Example媒体播放列表示例

```
#EXTM3U
#EXT-X-VERSION:6
#EXT-X-TARGETDURATION:6
#EXT-X-MEDIA-SEQUENCE:123456
#EXT-X-DISCONTINUITY-SEQUENCE:0
#EXTINF:6.0,
segment123456.ts
#EXTINF:6.0,
segment123457.ts
#EXT-X-DISCONTINUITY
#EXTINF:5.0,
ad-segment1.ts
#EXTINF:5.0,
ad-segment2.ts
#EXT-X-DISCONTINUITY
#EXTINF:6.0,
segment123458.ts
```

> **Note**
>
> 多变体播放列表和媒体播放列表都是使用扩展名的基于文本的.m3u8文件。这些文件的术语因不同的文档和上下文而异，但在 HLS 流媒体架构中，基本角色保持不变。

配置时 MediaTailor，您通常会提供内容来源中多变体播放列表的网址。 MediaTailor 然后处理多变体播放列表和媒体播放列表的个性化设置，以便根据您的配置插入广告。

有关 HLS 播放列表规格的更多信息，请参阅 HTTP Live Streaming RFC。

DASH 清单类型

基于 HTTP 的动态自适应流式传输 (DASH) 使用媒体演示描述 (MPD) 清单来传送流媒体内容。了解 DASH 清单的结构和组件对于配置和排除 MediaTailor 工作流程故障至关重要。

MPD（媒体演示说明）

MPD 是 DASH 流媒体中的主要清单文件，用于描述媒体内容的结构和可用性。它包含有关构成流媒体内容的时段、改编集、表示形式和片段的信息。

在各种上下文中，这种清单类型也被其他几个名称所熟知：

- DASH 清单
- DASH MPD
- 主清单（与 HLS 比较时）
- 演示文稿清单

在 MediaTailor 工作流程中，MPD 是播放请求的入口点，也是广告个性化的起点。

Example MPD 清单示例

```
<?xml version="1.0" encoding="UTF-8"?>
<MPD xmlns="urn:mpeg:dash:schema:mpd:2011" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011
 DASH-MPD.xsd" profiles="urn:mpeg:dash:profile:isoff-live:2011"
 type="dynamic" minBufferTime="PT5.000S" maxSegmentDuration="PT2.005S"
 availabilityStartTime="2020-01-01T00:00:00Z" publishTime="2020-01-01T12:30:00Z"
 minimumUpdatePeriod="PT2.000S" timeShiftBufferDepth="PT5M">
  <Period id="1" start="PT0.000S">
```

```
    <AdaptationSet id="1" contentType="video" segmentAlignment="true"
 bitstreamSwitching="true" frameRate="30000/1001" maxWidth="1920" maxHeight="1080"
 par="16:9">
      <Representation id="1" mimeType="video/mp4" codecs="avc1.640028" width="1920"
 height="1080" bandwidth="5000000">
        <SegmentTemplate timescale="90000" initialization="init-stream
$RepresentationID$.m4s" media="chunk-stream$RepresentationID$-$Number%05d$.m4s"
 startNumber="1" duration="180000"/>
      </Representation>
      <Representation id="2" mimeType="video/mp4" codecs="avc1.4d401f" width="1280"
 height="720" bandwidth="2800000">
        <SegmentTemplate timescale="90000" initialization="init-stream
$RepresentationID$.m4s" media="chunk-stream$RepresentationID$-$Number%05d$.m4s"
 startNumber="1" duration="180000"/>
      </Representation>
      <Representation id="3" mimeType="video/mp4" codecs="avc1.4d401e" width="640"
 height="360" bandwidth="1100000">
        <SegmentTemplate timescale="90000" initialization="init-stream
$RepresentationID$.m4s" media="chunk-stream$RepresentationID$-$Number%05d$.m4s"
 startNumber="1" duration="180000"/>
      </Representation>
    </AdaptationSet>
    <AdaptationSet id="2" contentType="audio" segmentAlignment="true" lang="eng">
      <Representation id="4" mimeType="audio/mp4" codecs="mp4a.40.2"
 audioSamplingRate="48000" bandwidth="128000">
        <SegmentTemplate timescale="48000" initialization="init-stream
$RepresentationID$.m4s" media="chunk-stream$RepresentationID$-$Number%05d$.m4s"
 startNumber="1" duration="96000"/>
      </Representation>
    </AdaptationSet>
  </Period>
</MPD>
```

周期

周期是 DASH 演示文稿中的一个时间部分。每个 Period 都包含一个或多个改编集，代表媒体时间跨度。在广告插入工作流程中，通常使用单独的时间段来划分内容和广告。

这个组件也有其他几个名字：

- 内容细分
- 时间轴部分
- 节目片段

在 MediaTailor 工作流程中，使用时间段将主要内容与广告内容分开，每个广告通常由自己的时间段表示。

Example时期示例

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

AdaptationSet

对一个或多个媒体内容组件的一 AdaptationSet 组可互换编码版本进行分组。例如，一个 AdaptationSet 可能包含多个视频质量级别，而另一个可能包含多个音频语言选项。

这个组件也被称为：

- 媒体组件组
- 直播套装
- 曲目组

在 MediaTailor 工作流程中， AdaptationSets 在广告插入期间保留，以保持内容和广告之间媒体类型的一致性。

表示

表示形式是其中媒体内容的特定编码版本。 AdaptationSet每种表示形式通常在比特率、分辨率或其他编码参数上有所不同，允许客户端根据网络条件和设备功能选择最合适的版本。

这个组件也被称为：

- 再现（类似于 HLS）
- 质量级别
- 比特率变体
- 直播变体

在 MediaTailor 工作流程中，广告周期中的表现尽可能与内容时段中的表现形式紧密匹配，以确保流畅的观看体验。

分段

区段是可以通过 URL 单独引用的媒体数据单元。片段包含实际的媒体内容（视频、音频等），并在 MPD 中引用。DASH 中有两种主要的区段类型：

- 初始化片段：包含表示的初始化信息，例如编解码器参数和计时信息。
- 媒体片段：包含表示形式中特定时间范围内的实际媒体数据。

在 MediaTailor 工作流程中，区段 URLs 通常会被修改为指向相应的内容或广告媒体文件。

> **Note**
>
> DASH 清单是基于 XML 的文件，通常使用扩展名。`.mpd`这些文件及其组件的术语可能因不同的文档和上下文而异，但是 DASH 流媒体架构中的基本结构保持不变。

配置时 MediaTailor，您需要在内容来源中提供 MPD 清单的 URL。 MediaTailor 然后处理清单的个性化设置，通常是根据您的配置为广告插入额外的时段。

有关 DASH 清单规范的更多信息，请参阅 DASH 行业论坛实施指南。

高级 DASH 清单概念

在中使用 DASH 清单时 MediaTailor，了解以下高级概念可以帮助您更有效地配置流媒体工作流程并对其进行故障排除：

时段开始时间计算

在 DASH 清单中，周期开始时间是根据 MPD 的`availabilityStartTime`属性和周期的属性计算的`start`。对于直播，开始时间与可用性开始时间有关，而对于 VOD 内容，它通常与演示开始时间有关。

在 MediaTailor 插入广告周期时，它会仔细计算开始时间，以确保内容和广告之间的无缝过渡。此计算考虑了：

- 原始时期的开始时间
- 之前的广告时段的持续时间
- 广告决策服务器响应中指定的任何时间偏移量

片前计时计算

DASH 清单中的预贴广告需要特殊处理，因为它们会在主要内容开始之前出现。 MediaTailor 在清单的开头将预贴片广告作为单独的时段插入，并附上适当的开始时间和持续时间。

对于预贴广告， MediaTailor请执行以下操作：

- 为每则预贴片广告创建一个新的广告周期
- 将第一个前置片段的开始时间设置为 0
- 调整主内容时段的开始时间，以考虑所有前置广告的总时长

Live-to-VOD 过渡

直播结束后，DASH 清单可以从直播转换为 VOD（视频点播）格式。这种过渡包括将 MPD 的`type`属性从更改`dynamic`为，`static`以及调整其他属性，例如`timeShiftBufferDepth`和。`minimumUpdatePeriod`

当 MediaTailor 流程在 live-to-VOD过渡期间出现时，它可以确保：

- 广告标记保留在 VOD 清单中
- 调整周期开始时间以保持适当的时间
- 清单仍然与 VOD 播放客户端兼容

DRM 处理

DASH 清单中的数字版权管理 (DRM) 信息通常包含在 AdaptationSets 或陈述中的`ContentProtection`元素中。 MediaTailor 在广告插入期间保留这些元素，以确保内容保护保持不变。

处理受 DRM 保护的内容时，请： MediaTailor

- 保留原始清单中的所有 ContentProtection 元素
- 确保广告内容使用兼容的 DRM 方案（如果适用）
- 在整个清单中保留所有与 DRM 相关的属性和元素

SCC 标志

DASH 清单中的补充内容控制 (SCC) 标志提供了有关内容特征和播放要求的更多信息。这些标志通常作为属性或元素包含在 MPD 结构中。

MediaTailor 处理的常见 SCC 标志包括：

- 内容评级信息
- 无障碍功能（隐藏式字幕、音频描述）

- 内容咨询通知
- 播放限制

MediaTailor 在清单处理期间保留这些标志，以确保所有内容元数据都保留在个性化清单中。

了解这些高级概念有助于您 MediaTailor 进行配置以获得最佳性能，并解决 DASH 流媒体工作流程中可能出现的任何问题。

广告插入概念

以下是与广告插入相关的概念的概述。

广告可用

广告时段内可以出售给广告商的特定广告时间单位。一个广告插播时间可能包含多个可用的广告。当 MediaTailor 收到来自广告决策服务器的 VAST 响应时，它会用个性化广告填充这些空白。

广告间隔

在节目中放映广告的时间段。广告中断可能发生在内容之前（前置片段）、内容期间（片中）或内容之后（后置片段）。 MediaTailor 通过标记（例如 SCTE-35.）识别清单中的广告中断。

广告决策服务器 (ADS)

一种服务器，可基于包括当前广告活动和观看者偏好在内的标准提供广告插播规范。

配置

你与 MediaTailor 之交互的对象。该配置包含有关源服务器和广告决策服务器 (ADS) 的位置信息。该配置还包含提供进出接入点的端点 MediaTailor。

动态转码

一个流程，用于在请求内容时将广告质量和格式与主要视频内容进行匹配。动态转码可降低存储要求，并确保播放在广告和视频内容之间无缝过渡。

明显的操纵

一个过程，用于从源服务器重写清单以使清单引用合适的广告和内容片段。广告由广告决策服务器 (ADS) 的 VAST 响应决定。随着播放的进行， MediaTailor 在内容流中插入广告或替换广告。

VAST 和 VMAP

视频广告投放模板 (VAST) 和视频多广告播放列表 (VMAP) 是广告决策服务器向其发送的广告请求的 XML 响应。 MediaTailor这些回复决定了广告在清单中 MediaTailor 插入了什么。VMAP 还包括

广告时段以及广告中提供的广告时间。有关 MediaTailor 广告插入背后的逻辑的更多信息，请参阅了解 AWS Elemental MediaTailor 广告插入行为。有关如何将 MediaTailor 与 VAST 结合使用的更多信息，请参阅 the section called “广告服务器集成要求”。

渠道组装概念

以下是与渠道组装相关的概念的概述。

频道

频道将您的源清单汇编成线性流。每个频道都有一个或多个输出，其中包含玩家访问的播放网址。频道输出与您为 VOD 源创建的套餐配置设置相对应。频道包含一个时间表，该时间表决定何时在频道的直播中播放 VOD 源。

Package 配置

打包程序配置代表您的 VOD 源，其中包含特定的打包格式特征。您可以将您的套餐配置与频道输出相关联，以便为 VOD 源的打包格式（例如 HTTP 直播 (HLS)）创建播放流。

日程安排

每个频道都由安排在频道时间表中的节目组成。时间表决定了节目将在频道的线性直播中播放的时间。

来源位置

源位置表示存储您的资产的源服务器。它可以是 Amazon S3、HTTP 服务器、内容分发网络 (CDN) 或打包基础设施，例如 MediaPackage。

VOD 来源

VOD 来源代表单个内容，例如电影或电视节目的剧集。您可以将 VOD 源与节目关联，将其添加到频道的线性直播中。

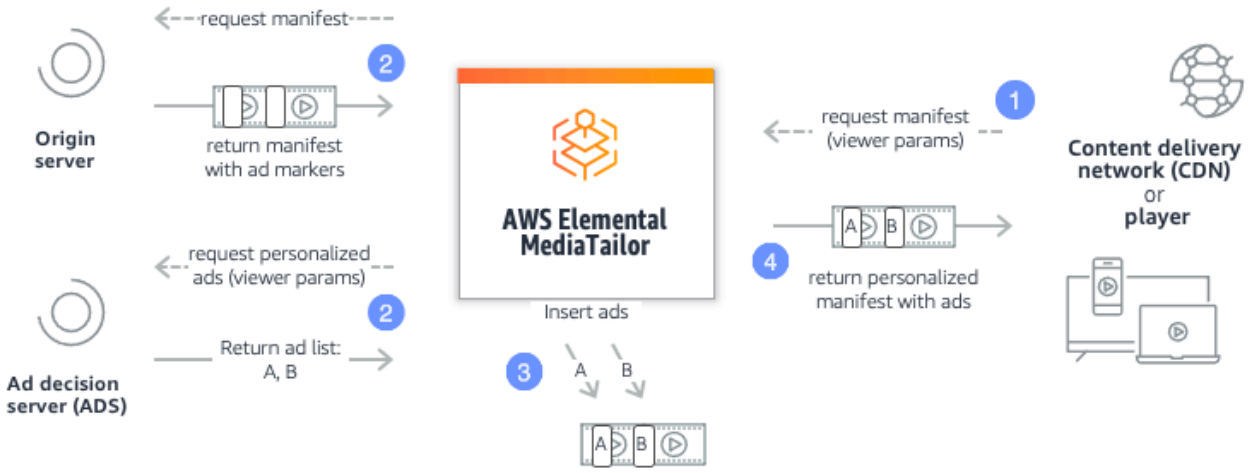
观众

受众定义了一个观众群体，该群组可以选择使用其他内容。您可以在标准线性频道上定义受众。

MediaTailor 广告插入的工作原理

AWS Elemental MediaTailor 在您的内容分发网络 (CDN)、源站服务器和广告决策服务器 (ADS) 之间进行交互，将个性化广告拼接到直播和视频点播内容中的广告插播中。

以下是 MediaTailor 广告插入工作原理的概述：



1. 玩家或 CDN（例如 Amazon） CloudFront 向发送请求，要求获取 HLS 或 DASH 内容。MediaTailor 该请求包含来自玩家的参数，其中包含有关观看者的信息，这些信息用于广告个性化。
2. MediaTailor 向 ADS 发送包含观众信息的请求。ADS 根据观看者信息和当前广告活动选择广告。它会在 VAST 或 VMAP 响应中返回广告素材的网址。 MediaTailor

   如果您对广告进行了预处理，则网址会指向预先转码的广告。有关使用预转码广告拼接广告的信息，请参阅。预处理广告
3. MediaTailor 操纵清单以包含从 ADS 返回的广告网址，这些网址经过转码以匹配原始内容的编码特征。如果您使用的是预处理广告，则您有责任确保广告与模板清单相匹配。

   如果尚未对广告进行转码以匹配内容，则 MediaTailor 将跳过插入广告并用于 MediaConvert 准备广告，以便为下一个请求做好准备。
4. MediaTailor 将完全个性化的清单返回给请求的 CDN 或玩家。

ADS 根据观看里程碑（例如广告开始、广告中间和广告结束）来跟踪查看的广告。随着播放的进行，玩家或 MediaTailor 向ADS广告跟踪网址发送广告跟踪信标，以记录广告的浏览量。在与的会话初始化中 MediaTailor，玩家会指示 MediaTailor 是否要为会话发送这些信标。

有关如何开始广告插入的信息，请参阅入门 MediaTailor。

AWS Elemental MediaTailor 广告插入事件流程

AWS Elemental MediaTailor 流程通过一系列可预测的事件来展示服务器端广告插入的个性化设置。了解此事件流程有助于您理解如何将广告机会 MediaTailor 转化为个性化的观看体验，并为问题发生时进行故障排除奠定基础。

每个广告插入机会都遵循按时间顺序 MediaTailor记录的事件序列，以提高可见性和监控。这些事件代表了清单个性化流程中的关键里程碑，从检测广告机会到提供跟踪信息。

标准事件序列

MediaTailor 成功处理广告插入机会后，通常会发生以下事件序列：

1. 广告机会检测- MediaTailor 检测内容清单中的广告标记（例如 SCTE-35），表明存在广告个性化机会。
2. 广告决策服务器请求- MediaTailor 向已配置的广告决策服务器 (ADS) 发送请求，以检索检测到的机会的广告，包括观众参数和广告中断时长。
3. 广告响应处理- MediaTailor 接收来自广告的响应，其中包含广告素材信息、跟踪网址和要包含的广告的元数据。有关 VAST、VMAP 和 VPAID 响应格式的更多信息，请参阅。广告服务器集成要求
4. M anifest personalization- MediaTailor 成功处理广告响应，必要时对广告进行转码，并生成包含广告细分和来源内容的相对网址的个性化清单。 MediaTailor 然后将个性化清单发送到播放设备。
5. 跟踪信标激活-触发跟踪信标，将广告事件（例如展示次数、四分位数和完成量）报告给广告服务器或其他衡量系统。在服务器端报告模式（默认）下，根据玩家分段请求 MediaTailor 触发信标。在客户端报告模式下，播放设备使用个性化清单中提供的 URL 来触发信标。

典型时机：整个序列通常在 2-5 秒内完成，具体取决于广告决策服务器的响应时间和广告创意处理要求。

常见的事件变体

并非所有的广告插入尝试都遵循标准的成功顺序。 MediaTailor处理可能改变事件流的各种场景：

空广告响应场景

当广告决策服务器未返回任何广告时：

1. 广告机会检测正常进行
2. 广告决策服务器请求已成功发送
3. ADS 返回的响应为空，广告为零

4. 此机会的个性化清单中不包含任何广告

这是由于广告库存可用性、定位标准或广告决策服务器中配置的业务规则而可能发生的正常情况。在这种情况下，除非直播流是直播或直播到 VOD 的直播且已配置 slate，否则将显示底层内容。有关更多信息，请参阅 MediaTailor slate 广告插入。

错误情形

当技术问题导致广告无法成功插入时，事件流程可能会在不同时刻中断：

- 广告决策服务器超时-向 ADS 发出的请求超过了配置的超时阈值，导致无法检索广告。
- 通信错误-网络或连接问题导致无法 MediaTailor 到达广告决策服务器。
- 响应解析错误-ADS 返回的响应由于 VAST 格式或结构无效而 MediaTailor 无法解析。

在错误情况下， MediaTailor 通常会继续播放没有广告的内容，从而确保不间断的观众体验。对于配置了 slate 的直播或 Live-to-VOD 直播，可能会改为显示 slate 内容。有关更多信息，请参阅 MediaTailor slate 广告插入。

大量的重定向场景

当初始 VAST 响应包含指向其他广告服务器的重定向时：

1. 广告机会检测和初始 ADS 请求正常进行
2. 初始响应包含重定向指令
3. MediaTailor 按照重定向来检索最终的广告响应
4. 在最终的广告内容中继续进行清单个性化设置
5. 跟踪信标正常激活

VAST 重定向在程序化广告中很常见，允许广告服务器链接和实时决策。 MediaTailor 在广告检索过程中允许最多 7 个包装器重定向和无限量的 HTTP 重定向。

活动时间概述

了解 MediaTailor 事件流的时间特征有助于为广告插入效果设定适当的预期：

- 广告机会检测-在 MediaTailor 收到玩家的`GetManifest`请求并遇到内容清单中的广告标记时发生。
- 广告决策服务器交互 ——通常需要 100-500 毫秒，具体取决于 ADS 响应时间和网络条件。

- 清单个性化设置-通常在收到 ADS 响应后的 50-200 毫秒内完成。
- 跟踪信标时机-因报告模式而异。对于服务器端信标，计时基于客户端玩家的分段请求；客户端信标会根据玩家的实现触发。

性能注意事项：从广告机会检测到清单投放的总时间通常应保持在 5 秒以内，以保持最佳的观众体验。如上所述，各个组件在毫秒内完成。但是，5 秒的阈值考虑了在整个广告插入工作流程中可能出现的潜在超时值、重试尝试次数和网络变异性。延迟时间较长可能表示 ADS 性能问题或网络连接问题。

使用事件流知识

了解 MediaTailor 事件流为以下方面奠定了基础：

- 实施规划-了解事件顺序有助于设计玩家集成和广告服务器配置。
- 性能优化-了解时间预期可以识别瓶颈和优化机会。
- 故障排除准备-熟悉正常的事件流可以更轻松地识别何时出现问题。

有关使用事件流分析的详细故障排除指南，请参阅问题排查。有关事件记录和监控的技术细节，请参阅查看日志。

相关服务

- Amazon CloudFront 是一项全球内容分发网络 (CDN) 服务，可将数据和视频安全地传送给您的观众。使用 CloudFront 以最佳的性能分发内容。有关的更多信息 CloudFront，请访问 Amazon CloudFront 网站。
- AWS Elemental MediaPackage 是一项即时打包和生成服务，用于自定义直播视频资产以便采用与发出请求的设备兼容的格式进行分发。使用 AWS Elemental MediaPackage 作为源服务器，以在将流发送到 MediaTailor 之前准备内容并添加广告标记。有关 MediaTailor 如何使用源服务器的更多信息，请参阅MediaTailor 广告插入的工作原理。
- AWS Identity and Access Management (IAM) 是一项 Web 服务，可帮助您安全地控制用户对 AWS 资源的访问权限。使用 IAM 控制谁可以使用您的 AWS 资源（身份验证）以及他们可以通过哪些方式使用哪些资源（授权）。有关更多信息，请参阅 设置 AWS Elemental MediaTailor。

正在访问 MediaTailor

MediaTailor 您可以使用该服务的控制台进行访问。

AWS 账户 通过提供证书来验证您是否有权使用服务，从而访问您的。

要登录 MediaTailor 控制台，请使用以下链接：**https://console.aws.amazon.com/mediatailor/home**。

的定价 MediaTailor

与其他 AWS 产品一样，没有合同或最低使用承诺 MediaTailor。您将根据自己对服务的使用量付费。有关更多信息，请参阅MediaTailor 定价。

的区域 MediaTailor

为了减少应用程序中的数据延迟，请 MediaTailor 提供区域终端节点来提出您的请求。要查看可用区域列表 MediaTailor ，请参阅区域终端节点。

设置 AWS Elemental MediaTailor

本节将指导您完成配置用户访问 AWS Elemental MediaTailor所需的步骤。有关身份和访问管理的背景和其他信息 MediaTailor，请参阅的身份和访问管理 AWS Elemental MediaTailor。

要开始使用 AWS Elemental MediaTailor，请完成以下步骤。

主题

- 注册参加 AWS 账户

注册参加 AWS 账户

要开始使用 AWS，你需要一个 AWS 账户. 有关创建的信息 AWS 账户，请参阅《AWS 账户管理 参考指南》 AWS 账户中的 “入门” 。

入门 MediaTailor

首先 MediaTailor，你可以在两个教程之间进行选择——一个是关于设置广告插入，另一个是关于频道集成的教程。您还可以探索现场活动的动手研讨会。广告插入教程将指导您发送播放请求，MediaTailor 将个性化广告纳入您的内容流。频道汇编教程将向您展示如何使用播放网址直接在浏览器中观看频道的直播，包括个性化广告。

您还可以探索现场活动框架研讨会。本次实践研讨会指导您完成 AWS Elemental MediaTailor 直播活动设置的完整生命周期，从初始发现和部署到优化和扩展。您可以关注研讨会，了解对所有活动和频道使用一种 MediaTailor 设置的简单配置，也可以关注高级架构，为每个活动或频道使用专用配置。

主题

- MediaTailor 广告插入入门
- MediaTailor 频道汇编入门

MediaTailor 广告插入入门

要使用 AWS Elemental MediaTailor，您需要 AWS 账户 和权限才能访问、查看和编辑 MediaTailor 配置。有关如何执行此操作的信息，请参阅 设置 AWS Elemental MediaTailor。

此入门教程演示如何执行以下任务：

- 准备好您的 HLS 或 DASH 内容流
- 配置广告决策服务器 (ADS) 模板网址
- 创建包含播放端点的 MediaTailor 配置
- 使用您的播放器或内容分发网络 (CDN) 向发送播放请求 MediaTailor

有关使用动态变量和玩家参数的高级 ADS 网址配置，请参阅MediaTailor ADS 请求的动态广告变量。有关通过 CDN 传递参数的信息，请参阅MediaTailor 清单查询参数。

完成后，您可以向发送播放请求，以 MediaTailor 获取直播中的个性化广告内容。

主题

- 先决条件
- 第 1 步：访问 AWS Elemental MediaTailor

- 第 2 步：准备直播
- 第 3 步：配置 ADS 请求 URL 和查询参数
- 第 4 步：创建配置
- 第 5 步：测试配置
- 第 6 步：将播放请求发送至 AWS Elemental MediaTailor
- 步骤 7（可选）：监控 AWS Elemental MediaTailor 活动
- 步骤 8：清除

先决条件

开始之前，请确保您已完成 设置 AWS Elemental MediaTailor 中的步骤。

第 1 步：访问 AWS Elemental MediaTailor

使用您的 IAM 证书登录 MediaTailor 控制台，网址为**`https://console.aws.amazon.com/mediatailor/home`**。

第 2 步：准备直播

配置您的源服务器，以便为 HLS 或 DASH 生成与 AWS Elemental MediaTailor兼容的清单。您的源服务器必须使用标准 HTTP 端口（端口 80）或 HTTPS 端口（端口 443）。 MediaTailor 不支持用于源服务器通信的自定义端口。

准备 HLS 直播

HLS 清单必须满足以下要求：

- 清单必须可在公共 Internet 上访问。
- 清单必须是直播或视频点播 (VOD)。
- 清单必须具有 `EXT-X-VERSION 3` 或更高版本。
- 对于直播内容，清单必须包含描述广告效用的标记。这对于 VOD 内容是可选的，VOD 内容可以改用 VMAP 时间偏移。

  清单文件必须包含标有以下内容之一的广告位：

  - #EXT-X-CUE-OUT /# EXT-X-CUE-IN（更常见），持续时间如以下示例所示。

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

- # EXT-X-DATERANGE（不太常见），其持续时间如以下示例所示。

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

  为 #EXT-X-DATERANGE 显示的所有字段都是必需的。

在清单中配置广告标记的方式将决定广告是插入到流中还是替换流中的其他片段。有关更多信息，请参阅 the section called “了解广告插入行为”。

- HLS 多变量播放列表必须遵循 HTTP 直播中记录的 HLS 规范：主播放列表标签。具体而言，#EXT-X-STREAM-INF 必须包含字段 RESOLUTION、BANDWIDTH 和 CODEC。

配置直播后，记下多变量播放列表的内容来源 URL 前缀。在本教程的后面，您需要使用它在 AWS Elemental MediaTailor中创建配置。

准备 DASH 直播

DASH 清单必须满足以下要求：

- 清单必须可在公共 Internet 上访问。
- 清单必须是直播或视频点播 (VOD)。
- 清单必须使用拼接插入标记或时间信号标记将事件标记为广告效用。您可以使用清除 XML 或 base64 编码的二进制提供广告标记。对于拼接插入，必须启用网络外指示符。对于时间信号标记，位于分段 UPID 内的分段类型 ID 必须是可识别的提示值。 AWS Elemental MediaTailor广告效用随相关事件的开始而开始，持续时间与事件持续时间（如果已指定）相同，或者持续到下一个事件开始。

  以下示例显示使用拼接插入标记指定为广告效用的事件。此广告效用的持续时间与事件的持续时间相同。

```
 <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>
```

- 广告效用必须具有与内容流相同的 AdaptationSet 和 Representation 设置。 AWS Elemental MediaTailor 使用这些设置来转码广告以匹配内容流，从而实现两者之间的流畅拼接。

配置流后，请记下 DASH 清单的内容源 URL 前缀。在本教程的后面部分中 AWS Elemental MediaTailor，您需要它来创建配置。

第 3 步：配置 ADS 请求 URL 和查询参数

要确定 ADS 需要的查询参数，请从 ADS 生成一个广告标签 URL。此 URL 充当对 ADS 请求的模板，并包含以下内容：

- 静态值
- 由 AWS Elemental MediaTailor （由session或avail查询参数表示）生成的值
- 播放器生成的值，从客户端应用程序获取（由 player_params. 查询参数指示）

Example来自 ADS 的广告标签 URL

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

其中：

- output 和 content_id 是静态值

- PlayerSession= [session.id] 是由提供的动态值。 AWS Elemental MediaTailor[session.id] 的值将随每个播放器会话更改，并为针对每个会话的 VAST 请求生成不同的 URL。
- cust_params 是播放器提供的动态值

来自玩家的多变量播放列表请求必须提供与 ADS 请求 URL 中的`player_params.`查询参数相对应的键值对。有关在向的请求中配置键值对的更多信息 AWS Elemental MediaTailor，请参阅。MediaTailor ADS 请求的动态广告变量

在中创建源 server/ADS 映射时，输入配置的 “模板” URL 第 4 步：创建配置。 MediaTailor

测试

您可以使用来自 ADS 的静态 VAST 响应进行测试。理想情况下，VAST 响应会返回 AWS Elemental MediaTailor 可以转码的夹层品质的 MP4 呈现。如果来自 ADS 的响应包含多个播放格式副本，则 MediaTailor 选择最高质量和分辨率的 MP4 格式副本并将其发送到转码器。

第 4 步：创建配置

该 AWS Elemental MediaTailor 配置包含源服务器和 ADS 的映射信息。

创建配置（控制台）

1. 在处打开 MediaTailor 控制台https://console.aws.amazon.com/mediatailor/。
2. 在 Configurations (配置) 页面上，选择 Create configuration (创建配置)。
3. 在页面底部的 Configuration (配置) 部分，在 Configuration name (配置名称) 中，输入用于描述配置的唯一名称。此名称是配置的主要标识符。允许的最大长度为 512 个字符。
4. 对于视频内容来源，输入该直播的 HLS 多变量播放列表或 DASH 清单的网址前缀，减去资产 ID。例如，如果多变播放列表 URL 为`http://origin-server.com/a/master.m3u8`，则应输入`http://origin-server.com/a/`。或者，您也可以输入一个较短的前缀（如 `http://origin-server.com`），但您随后必须在针对内容的播放器请求的资产 ID 中包含 `/a/`。最大长度为 512 个字符。

> **Note**
>
> 如果您的内容源使用 HTTPS，则其证书必须来自知名的证书颁发机构。（它不能是自签名证书。）否则，将 AWS Elemental MediaTailor 无法连接到内容来源，也无法提供清单以响应玩家的请求。

5. 对于广告决策服务器，输入您的广告的网址。这是包含变量的 URL（如第 3 步：配置 ADS 请求 URL 和查询参数中所述），或用于测试用途的静态 VAST URL。最大长度为 25000 个字符。

   > **Note**
   >
   > 如果您的 ADS 使用 HTTPS，则其证书必须来自知名的证书颁发机构。（它不能是自签名证书。）这同样适用于 ADS 返回的夹层广告 URL。否则， MediaTailor 无法检索广告并将其拼接到来自内容源的清单中。

6. （根据 DASH 需要可选）如果您设置了 CDN 路由规则来访问 MediaTailor 清单，并且您正在使用客户端举报或者您的玩家支持粘性 HTTP 重定向，请在 “位置” 中选择 “禁用”。

   有关 Location (位置) 功能的更多信息，请参阅 the section called “位置功能”。

7. （可选）如果您的源服务器生成单周期 DASH 清单，请选择 DASH mpd manifest origin type (DASH mpd 清单原始类型)，然后选择 SINGLE_PERIOD。默认情况下，将 DASH 清单作为多周期清单 MediaTailor 处理。有关更多信息，请参阅 the section called “整合 MPEG-DASH 来源”。

8. 选择创建配置。

   AWS Elemental MediaTailor 在 “配置” 页面上显示新配置。

第 5 步：测试配置

保存配置后，使用适合您的流式处理协议的格式的 URL 测试流：

- 示例：HLS

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

- 示例：DASH

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

其中：

- `playback-endpoint` 是 AWS Elemental MediaTailor 在创建配置后生成的唯一播放终端节点。

  示例

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

- `hashed-account-id`是你的 AWS 账户 身份证。

  示例

  ```
  777788889999
  ```

- `origin-id` 是您在创建配置时提供的名称。

  示例

  ```
  myOrigin
  ```

- `master.m3u8` 或 `manifest.mpd` 是来自测试流的清单的名称加上其文件扩展名。定义此项，从而在将它附加到您在the section called “第 4 步：创建配置”中配置的视频内容源时获得完全标识的清单。

通过使用前面示例中的值，完整的 URL 如下所示。

- 示例：HLS

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

- 示例：DASH

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

您可以使用以下方法之一测试流。

- 按以上示例中所示，在独立的播放器输入 URL。

- 在您自己的播放器环境中测试流。

第 6 步：将播放请求发送至 AWS Elemental MediaTailor

配置下游播放器或 CDN 以将播放请求发送到从 AWS Elemental MediaTailor提供的配置的播放终端节点。您在第 3 步：配置 ADS 请求 URL 和查询参数中的 ADS 请求 URL 中使用的任何播放器定义的动态变量必须在来自播放器的清单请求中定义。

Example

假如您的模板 ADS URL 如下。

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

然后[player_params.cust_params]在玩家请求中通过在键值对开头进行定义。ads. AWS Elemental MediaTailor 将前面没有的参数传递ads.给源服务器而不是 ADS。

播放器请求 URL 是以下 HLS 和 DASH 示例的某个变体。

```
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
```

当 AWS Elemental MediaTailor 收到玩家请求时，它会根据请求中的信息定义玩家变量。生成的 ADS 请求 URL 是以下内容的某个变体。

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

有关配置要传递到 ADS 的键-值对的更多信息，请参阅 MediaTailor ADS 请求的动态广告变量。

步骤 7（可选）：监控 AWS Elemental MediaTailor 活动

使用亚马逊 CloudWatch 和亚马逊 CloudWatch 日志来跟踪 AWS Elemental MediaTailor 活动，例如请求数、错误数和已填充的广告数量。

如果这是您首次 CloudWatch 使用 with AWS Elemental MediaTailor，请创建 AWS Identity and Access Management (IAM) 角色以允许服务之间的通信。

允许 AWS Elemental MediaTailor 访问 CloudWatch（控制台）

1. 使用 https://console.aws.amazon.com/iam/ 打开 IAM 控制台。
2. 在 IAM 控制台的导航窗格中，选择角色，然后选择创建角色。
3. 选择其他 AWS 账户角色类型。
4. 在“账户 ID”中，输入您的 AWS 账户 ID。
5. 选择 Require external ID (需要外部 ID) 并输入 **`midas`**。此选项会在信任策略中自动添加一个条件，即仅当请求包含正确的 `sts:ExternalID` 时，该服务才代入该角色。
6. 选择下一步: 权限。
7. 添加指定此角色可完成的操作的权限策略。从以下选项中选择一项，然后选择 Next: Review (下一步: 审核)：

   - CloudWatchLogsFullAccess提供对亚马逊 CloudWatch 日志的完全访问权限
   - CloudWatchFullAccess提供对亚马逊的完全访问权限 CloudWatch
8. 对于角色名称，输入 **`MediaTailorLogger`**，然后选择创建角色。
9. 在 Roles (角色) 页上，选择您刚刚创建的角色。
10. 编辑信任关系以更新委托人：

    1. 在角色的 Summary (摘要) 页上，选择 Trust relationship (信任关系) 选项卡。
    2. 选择编辑信任关系。
    3. 在策略文档中，将委托人更改为 AWS Elemental MediaTailor 服务。它应如下所示。

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

       整个策略的内容现在应如下所示。
    4. 选择更新信任策略。

步骤 8：清除

为避免产生多余的费用，请删除所有不必要的配置。

删除配置（控制台）

1. 在 AWS Elemental MediaTailor 配置页面上，执行以下操作之一：

- 为要删除的配置选择 Configuration name (配置名称)。
- 在 Configuration name (配置名称) 列中，选择单选按钮，然后选择 Delete (删除)。

2. 在 Delete configuration (删除配置) 确认框中，输入 **`Delete`**，然后再次选择 Delete (删除)。

   AWS Elemental MediaTailor 移除配置。

MediaTailor 频道汇编入门

此入门教程演示如何执行以下任务：

- 创建源位置并向其添加源内容
- 创建通道
- 创建节目列表以按计划播放您频道的内容
- 使用广告插入向频道流添加个性化 AWS Elemental MediaTailor 广告

完成后，您将能够打开浏览器，输入频道的播放网址，并查看包含个性化广告的频道直播。

本教程将引导您完成开始 MediaTailor 信道汇编的基本步骤。有关更多高级信息，请参阅使用 AWS Elemental MediaTailor 创建线性组装流。

估算费用

- 活跃频道的费用为每小时0.10美元。您无需为处于非活动状态的频道付费。

主题

- 先决条件
- 第 1 步：创建来源位置
- 第 2 步：将 VOD 源添加到您的源位置
- 第 3 步：创建频道
- 第 4 步：将节目添加到频道的时间表中
- 第 5 步 (optional)： MediaTailor 用于在您的直播中插入个性化广告
- 第 6 步：启动您的频道
- 第 7 步：测试您的频道

- 步骤 8：清除

先决条件

在开始本教程之前，必须完成以下要求：

- 确保您已完成中的步骤设置 AWS Elemental MediaTailor。
- 你必须有可用于 VOD 源内容和广告列表的资源。你必须知道资产清单的路径。

> **Note**
>
> 如果您使用自动自适应比特率 (ABR) 或按标题编码，则必须对资源进行编码，以使所有变体的长度相同，子曲目数量相同。我们建议您使用最小段长度为一秒的编码模板。

第 1 步：创建来源位置

源位置代表存储内容的源服务器。它可以是 Amazon S3、标准 Web 服务器、内容分发网络 (CDN) 或打包来源，例如 AWS Elemental MediaPackage。

MediaTailor 从您的源位置获取内容清单，并使用它们来组装一个引用基础内容段的实时滑动清单窗口。

要创建源位置，请执行以下步骤。

创建来源位置

1. 在处打开 MediaTailor 控制台https://console.aws.amazon.com/mediatailor/。
2. 在导航窗格中，选择 “频道汇编” > “来源位置” 。
3. 在导航栏上，选择创建来源位置。
4. 在 “来源位置配置” 下，输入标识符和源内容的位置：
   - 名称：您的来源位置的标识符，例如 my- origin。
   - 基本 URL：托管内容的原始服务器的基本 URL，例如https://111111111111.cloudfront.net。该网址必须采用标准的 HTTP 网址格式，以 http://或 https://为前缀。
5. 选择 “创建来源位置” 。

第 2 步：将 VOD 源添加到您的源位置

现在您已经为频道定义了一个或多个来源位置，您可以添加一个或多个 VOD 来源。每个 VOD 来源代表单个内容，例如一部电影、一集电视节目或精彩片段。

您必须为 VOD 源创建至少一个包配置。每个包配置都包含 VOD 源的打包格式和清单设置。然后，您将套餐配置添加到您的频道以创建输出。

您可以使用多个封装配置来创建不同的声道输出。例如，如果您的 VOD 源同时打包为 HLS 和 DASH，则可以为每种格式创建两个包配置。然后，您可以使用软件包配置的源组来创建两个通道输出：一个用于 HLS，一个用于 DASH。

添加 VOD 源和创建包配置

1. 在处打开 MediaTailor 控制台https://console.aws.amazon.com/mediatailor/。
2. 在导航窗格中，选择 “频道汇编” > “来源位置” 。
3. 在来源位置窗格中，选择您在创建来源位置过程中创建的源位置。
4. 选择 “添加 VOD 来源”。
5. 在 VOD 来源详细信息下，输入您的 VOD 来源的名称，例如 my-example-video。
6. 在 “包配置***source-group-name***” > 下输入有关软件包配置的信息：

   > **Note**
   >
   > 源代码的包配置都必须具有相同的持续时间，具体取决于源代码的清单。而且，包配置中的所有源必须具有相同数量的子流。为了满足这些要求，我们建议您为资产使用编码模板。我们建议您使用最小段长度为一秒的编码模板。 MediaTailor 不支持每部影片或自动自适应比特率直播 (ABR)，因为这些编码方法违反了这些要求。

   - 源组：输入描述此包配置的源组名称，例如 HLS-4k。记下这个名字；你将在创建频道的输出时引用它。有关更多信息，请参阅 将来源组与频道的输出配合使用。
   - 类型：选择此配置的打包格式。 MediaTailor支持 HLS 和 DASH。
   - 相对路径：从源位置的基本 HTTP 网址到清单的相对路径。例如，/my/path/index .m3u8。
7. 选择 添加源。
8. 重复此过程中的第 4-7 步，为您的广告列表添加 VOD 来源。

第 3 步：创建频道

频道将您的源代码汇编成线性直播。每个频道都包含一个或多个与您的 VOD 源的包配置相对应的输出。

首先，你创建一个频道，然后通过创建节目将你的VOD源添加到频道的日程表中。

创建通道

1. 在处打开 MediaTailor 控制台https://console.aws.amazon.com/mediatailor/。
2. 在导航窗格中，选择 “频道汇编” > “频道” 。
3. 在导航栏上，选择创建频道。
4. 在 “频道详情” 下，输入有关您的频道的详细信息：
   - 名称：输入您的频道的名称。
   - 回放模式：确定允许哪种程序过渡以及程序完成后会发生什么。使用默认循环模式。
5. 选择下一步。
6. 在 “输出详细信息” 下，定义此输出的设置：
   - 清单名称：输入清单名称，例如***index***。 MediaTailor 将附加格式扩展名，例如 HLS 的 .m3u8。

     > **Note**
     >
     > 您必须为每个通道输出输入一个唯一的清单名称。

   - 格式类型：选择频道的直播格式。支持 DASH 和 HLS。选择与您在中创建的包配置相对应的格式第 1 步：创建来源位置。
   - 来源组：输入您在中创建的源组的名称第 1 步：创建来源位置。
7. 在 “清单设置” 下，输入有关您的清单设置的更多信息：
   - 清单窗口（秒）：每个清单中包含的时间窗口（以秒为单位）。最小值为 30 秒，最大值为 3600 秒。
8. 选择下一步。
9. 在 “渠道政策” 下，选择 “不附加频道政策” 。此选项仅允许有权访问您的 AWS 账户 凭证的用户播放。
10. 选择下一步。

11. 在“查看并创建”窗格中查看您的设置。
12. 选择 Create channel (创建通道)。

> **Note**
>
> 频道是在停止状态下创建的。在您启动之前，您的频道不会处于活动状态。

第 4 步：将节目添加到频道的时间表中

现在您有了一个频道，您将在该频道的日程表中添加节目。每个程序都包含来自您账户中某个源位置的 VOD 来源。频道时间表决定了您的节目在频道的直播中播放的顺序。

每个程序可以有一个或多个广告插播时间。您可以通过指定用作广告列表的 VOD 来源来插入广告插播时间。广告时段的持续时间由广告牌的持续时间决定。您可以选择使用服务器端广告插入服务器（例如广告插入）来对 MediaTailor 广告插播时间进行个性化设置。

向频道的日程表中添加节目

1. 在处打开 MediaTailor 控制台https://console.aws.amazon.com/mediatailor/。
2. 在导航窗格中，选择“频道汇编”>“频道”。
3. 在“频道”窗格中，选择您在该第 3 步：创建频道过程中创建的频道。
4. 在“计划详情”下，输入有关您的计划的详细信息：
   - 名称：这是要添加到您的频道日程表中的节目的名称。
   - 来源位置名称：选择“选择现有来源位置”，然后第 1 步：创建来源位置从“选择来源位置”下拉菜单中选择您在中创建的来源位置。
   - VOD 源名称：选择“选择现有 VOD 来源”，然后选择您在本教程前面创建的 VOD 来源。
5. 在“播放配置”下，定义在频道时间表中插入节目的方式和时间：
   - 过渡类型：此值固定为“相对”。相对过渡类型表示该程序是相对于程序列表中的其他程序发生的。
   - 相对位置：如果这是您频道节目表中的第一个节目，则可以跳过此设置。如果这不是您频道日程表中的第一个节目，请在节目列表中选择要追加该节目的位置。您可以选择“程序之前”或“程序之后”。

- 相对节目：如果这是您的日程安排中的第一个节目，则可以跳过此设置。如果这不是您频道日程表中的第一个节目，请选择“使用现有节目”，选择您在其中创建的节目名称向频道的日程表中添加节目。

6. 选择添加广告插播时间。在“广告插播时间”下，配置广告插播时间的设置：

   - Slate 源位置名称：选择“选择现有源位置”，然后选择您在本教程前面创建的 Slate 存储源位置。
   - VOD 源名称：选择 “选择现有的 VOD 来源”，然后选择您在本教程前面添加的 Slate 中使用的 VOD 来源。广告板的持续时间决定了广告间隔的持续时间。
   - 对于以毫秒为单位的偏移量：该值以毫秒为单位确定广告时段开始时间，即相对于节目开始时的偏移量。输入任何小于 VOD 源时长且与节目 VOD 源（所有音频、视频和隐藏字幕轨道）内所有曲目（所有音频、视频和隐藏字幕轨道）的片段边界对齐的值，否则广告插播将被跳过。例如，如果您输入 0，则会创建在节目开始之前播放的片前广告插播时间。注意：。

7. 选择“添加程序”。

   有关程序的更多信息，请参阅Configuring ad breaks for your program。

   有关在线性直播中使用广告的更多高级信息，请参阅可选配置设置。

第 5 步 (optional)： MediaTailor 用于在您的直播中插入个性化广告

您现在有了一个包含节目的频道。如果你愿意，你可以使用 MediaTailor 在频道直播中将个性化广告插入节目的广告插播中。

先决条件

在继续操作之前，必须满足以下要求：

- 您必须拥有广告决策服务器 (ADS)。
- 您必须在此将节目添加到频道的日程安排中过程中配置广告插播时间段设置。

要在频道的直播中添加个性化广告，请使用 MediaTailor

1. 在处打开 MediaTailor 控制台https://console.aws.amazon.com/mediatailor/。
2. 在导航窗格中，选择配置。
3. 在“必填设置”下，输入有关您的配置的基本必填信息：

- 名称：您的配置的名称。
- 内容来源：输入频道输出中的播放网址，减去文件名和扩展名。有关 MediaTailor 配置的高级信息，请参阅必需的设置。
- 添加决策服务器：输入您的广告的网址。

4. 您可以选择配置别名、个性化详细信息和高级设置。有关这些设置的信息，请参阅可选配置设置。
5. 在导航栏上，选择创建配置。

有关使用 MediaTailor 广告插入的更多高级信息，请参阅使用 AWS Elemental MediaTailor 插入广告。

第 6 步：启动您的频道

你现在有了一个频道。但是在访问该频道的直播之前，你需要先启动你的频道。如果您尝试在频道处于活动状态之前访问该频道，则会 MediaTailor 返回 HTTP 4xx 错误代码。

开始你的频道

1. 在处打开 MediaTailor 控制台https://console.aws.amazon.com/mediatailor/。
2. 在导航窗格中，选择 “频道汇编” > “频道” 。
3. 在导航栏上，选择 “开始” 。

第 7 步：测试您的频道

要验证您的频道是否正常运行，请打开网络浏览器并输入频道输出中的网址。你应该看到你频道的直播。

在某些情况下，您可能需要清除缓存才能看到预期行为。

步骤 8：清除

完成为本教程创建的频道后，应通过删除该频道进行清理。

一旦频道状态变为已停止，您将停止为该频道支付费用。要保留您的频道以备日后使用，但不产生费用，您可以立即停止该频道，然后再开始播放。

删除您的频道

1. 在处打开 MediaTailor 控制台https://console.aws.amazon.com/mediatailor/。

2. 在导航窗格中，选择 “频道汇编” > “频道” 。
3. 选择要删除的频道。
4. 如果您的频道正在运行，请从 “操作” 下拉菜单中选择 “停止” 。必须先停止您的频道，然后才能将其删除。
5. 当您的频道停止时，从 “操作” 下拉菜单中选择 “删除” 。

使用 AWS Elemental MediaTailor 插入广告

配置是您在 AWS Elemental MediaTailor中与之交互的对象。该配置包含源服务器和广告决策服务器 (ADS) 的映射信息。您还可以定义默认播放时间， MediaTailor 以便在广告不可用或无法填满整个广告空闲时使用。

如果将内容分发网络 (CDN) 与一起使用 MediaTailor，则必须先在 CDN 中设置行为规则，然后才能将 CDN 信息添加到配置中。有关设置 CDN 的更多信息，请参阅使用 CDN 优化 MediaTailor 广告个性化和内容交付。

主题

- 支持的音频和视频编解码器
- 了解 AWS Elemental MediaTailor 广告插入行为
- MediaTailor 服务器引导的广告插入概述和实现
- MediaTailor 广告服务器集成要求
- 广告自助餐
- MediaTailor 回放配置管理
- 集成内容源以进行 MediaTailor 广告插入
- 集成 AWS Elemental MediaTailor 使用谷歌的广告平台
- 使用广告中断抑制功能自定义广告中断行为
- MediaTailor 保险杠广告插入
- MediaTailor 片前广告插播
- MediaTailor slate 广告插入
- 中的广告转码 AWS Elemental MediaTailor
- 预取广告
- 将预处理广告与 AWS Elemental MediaTailor
- 使用 MediaTailor 变现功能
- MediaTailor ADS 请求的动态广告变量
- MediaTailor 清单查询参数
- 报告广告跟踪数据
- 叠加广告
- 广告 ID 装饰

支持的音频和视频编解码器

MediaTailor 支持以下编解码器。

- 音频编解码器：mp4a、ac-3和 ec-3
- 视频编解码器：h.264(AVC)、h.265 (HEVC)、av01 () AV1

了解 AWS Elemental MediaTailor 广告插入行为

AWS Elemental MediaTailor 通过将广告替换或插入到来源清单中，将广告拼接到直播或视频点播 (VOD) 内容中。是插入还是替换广告取决于来源清单中广告插播时间的配置方式，以及内容是 VOD 还是直播。广告插播时间是节目中展示广告的时间段，而广告可用量是广告时段内可以充斥广告的具体广告时间单位。

- 使用广告替换， MediaTailor 将内容区段替换为广告。
- 通过广告插入， MediaTailor 可以在不存在区段的地方插入广告内容。

要了解如何将广告 MediaTailor 拼接到直播和 VOD 内容中，请选择适用的主题。

主题

- VOD 的广告拼接行为
- 直播广告拼接行为

VOD 的广告拼接行为

MediaTailor 根据来源清单中广告标记的配置方式以及广告决策服务器 (ADS) 是否发送 VMAP 响应，在 VOD 内容中插入或替换广告。

对于按标记配置显示的广告行为，请参阅以下各节。

如果存在广告标记

AWS Elemental MediaTailor 在来源清单中存在 SCTE-35 广告标记的地方插入广告。时0长EXT-X-CUE-OUT值为的广告标记表示广告的插入。

HLS 广告标记指南

请遵循以下指导方针进行后贴片和广告 Pod SCTE 信号：

片前广告

对于 HLS 后置式广告，CUE-OUT/IN 标记必须位于最后一个内容分段之前。这是因为 HLS 规范要求必须在分段之前明确声明标记装饰器。

例如，考虑以下声明：

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

AWS Elemental MediaTailor 插入如下所示的帖子。

```
#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: 广告面板

CUE-OUT/IN标签必须明确附加到区段。您不能连续使用多个 CUE-OUT/IN 标记来模拟广告 pod 行为。

例如，以下声明是描绘广告窗格的有效用法。CUE-OUT/IN

```
#EXT-X-CUE-OUT: 0
#EXT-X-CUE-IN
#EXTINF:4.000,
Somecontent1.ts
#EXT-X-CUE-OUT: 0
#EXT-X-CUE-IN
#EXTINF:4.000,
Somecontent2.ts
#EXT-X-CUE-OUT: 0
```

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

上述声明会导致如下所示的输出。

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

以下声明无效。

```
#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
```

如果没有广告标记

广告标记是在清单中发出广告中断信号的推荐方式。但是，不需要广告标记。如果清单中不包含 DASH 或 HLS 的广告标 MediaTailor 记，则只需调用 ADS 并根据响应创建广告插播时间：

- 如果 ADS 发送了 VAST 响应，则 MediaTailor 会在清单开头的广告插播时段中插入响应中的所有广告。这是一种前置式广告。
- 如果 ADS 发送 VMAP 响应，则 MediaTailor 使用广告中断时间偏移来创建广告片段，并在指定时间（前置片段、中片或片尾片段）将其插入到整个清单中。 MediaTailor 使用清单中每个广告时段的 VMAP 响应中每个广告时段的所有广告。

> **Note**
>
> 对于 VOD 内容，当片段与具有 VMAP 的插入点重叠时， MediaTailor 会向下舍入到最近的插入点。

> **Tip**
>
> 如果您希望创建中置式广告时段，但您的 ADS 不支持 VMAP，请确保清单中有广告标记。MediaTailor 将在标记处插入广告，如以下各个部分中所述。

> **Note**
>
> 对于服务器引导的广告插入方法， MediaTailor 请在清单的顶部插入前置广告，然后玩家在其他广告类型之前播放这些广告。

直播广告拼接行为

在直播中， AWS Elemental MediaTailor 始终执行广告替换，尽可能保持广告标记之间的总时间。当广告标记包含该`DURATION`属性时， MediaTailor 使用该值来确定广告时段的持续时间。在实时工作流程中，每个`CUE-OUT`和`CUE-IN`指标都必须有持续时间或匹配的指标。

MediaTailor 替换 HLS 和 DASH 直播内容的广告。有关如何 MediaTailor 计算广告插播时间和时间的信息，请参阅the section called “广告标记”和the section called “广告标记”。

广告选择和替换

AWS Elemental MediaTailor 包括来自广告决策服务器 (ADS) VAST 响应的广告，如下所示：

- 如果指定了持续时间，则 MediaTailor 选择一组适合时长的广告并将其包括在内。
- 如果未指定持续时间，则会尽可能多地 MediaTailor 播放广告，直到遇到表示返回主内容的广告标记为止。

AWS Elemental MediaTailor 在直播广告替换期间，请遵守以下准则：

- MediaTailor 尝试播放完整的广告，而不进行剪辑或截断。
- 每当 MediaTailor 遇到表明广告插播结束的广告标记时，它都会返回到底层内容。这可能意味着缩短当前播放的广告。
- 在持续时间结束时， MediaTailor 返回到基础内容。

- 如果 MediaTailor 在广告时段内没有广告可供播放，则它要么播放 slate（如果已配置），要么恢复底层内容流的播放。这种情况通常出现在没有足够的转码广告来填补广告时段的持续时间时。

> **Tip**
>
> 您可以使用个性化的阈值配置设置，定义广告时段中允许的有限未填充广告时间。有关更多信息，请参阅PlaybackConfiguration 参考资料。

服务器引导式广告插入的实时预告片

与服务器端广告插入相比，对于服务器引导的广告插入方法，Live preroll 的工作方式有所不同：

服务器端广告插入（拼接模式）

每位观众会话开始时，前置广告会取代部分直播内容。每位观众根据加入直播的时间在不同的时间看到预告片。

服务器引导的广告插入方法

MediaTailor 在所有带有属性的媒体清单的顶部放置一个 preroll daterange 标签。`CUE="PRE,ONCE"`这会导致玩家在播放开始时请求和播放前贴片广告一次，尽管他们共享的是相同的非个性化清单。

配置要求：

- 直播前片广告决策服务器：为前置广告配置 VAST 端点（可能与插播广告不同）
- 直播前片允许的最大持续时间：设置前置广告的最大持续时间（可选，如果省略，则将使用所有返回的广告）

技术实现：

- Preroll 日期范围标签使用`START-DATE="1970-01-01T00:00:00.000Z"`（Unix 时代）
- 素材资源列表请求使用配置的预贴广告决策服务器，而不是普通的 ADS
- 玩家通过资产列表`availId="aws-mediatailor-preroll-1"`中的数据识别预览请求

> **Important**
>
> 对于直播，前置广告会覆盖内容，而不是延迟内容。未来的版本可能会通过其他配置选项支持内容延迟模式。

> **Note**
>
> 对于服务器引导的广告插入，直播内容和 VOD 内容的预贴行为各不相同。直播内容需要明确的预贴片配置，而 VOD 内容默认包含使用常规广告决策服务器的预贴片。

示例

- 如果广告时段的持续时间设置为 70 秒，并且 ADS 响应包含两个 40 秒的广告，则会 AWS Elemental MediaTailor 播放其中一个 40 秒的广告。在剩下的时间内，它将切换到配置的画面或基础内容。在此过程中的任何时间点，如果 MediaTailor 遇到广告插入指示器，则它将立即切换到基础内容。
- 如果广告时段的持续时间设置为 30 秒，而 ADS 响应提供的最短广告为 40 秒，则不 MediaTailor 播放任何广告。如果配置了广告名单，则会 MediaTailor 播放 30 秒钟或直到遇到提示指示器。否则，MediaTailor 播放基础内容。

MediaTailor 服务器引导的广告插入概述和实现

AWS Elemental MediaTailor 服务器引导式广告插入 (SGAI) 通过将广告引用为单独的播放列表，而不是将其直接拼接到媒体播放列表中，从而为服务器端广告插入提供了一种替代方案。这种方法通过可缓存的清单提高了性能，并实现了更好的可扩展性。

有关如何使用服务器引导式广告插入功能的信息 MediaTailor，请从以下列表中选择适用的主题。

在播放配置中启用

要允许玩家使用服务器引导的广告插入，您必须在 MediaTailor播放配置PLAYER_SELECT中`Insertion Mode`将其设置为。这允许玩家在会话初始化时选择拼接广告或引导式广告插入。

创建服务器引导式会话

创建播放会话时，请选择引导模式。如何做到这一点取决于你的玩家是使用隐式会话还是显式会话。

隐式创建的服务器引导会话

追加到 aws.insertionMode=GUIDED HLS 多变体播放列表请求中。示例：

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

其中：

- playback-endpoint 是 AWS Elemental MediaTailor 在创建配置后生成的唯一播放终端节点。

  示例

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

- hashed-account-id是你的 AWS 账户 身份证。

  示例

  ```
  777788889999
  ```

- origin-id 是您在创建配置时提供的名称。

  示例

  ```
  myOrigin
  ```

- index.m3u8或者是测试流中清单的名称及其文件扩展名。定义此项，从而在将它附加到您在the section called “第 4 步：创建配置”中配置的视频内容源时获得完全标识的清单。

使用前面示例中的值，完整 URLs 内容如下。

- 示例：

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

明确创建的服务器引导会话

将玩家通过 HTTP POST 发送`insertionMode=GUIDED`到 MediaTailor 配置的会话初始化前缀端点的 JSON 元数据中添加。

以下示例显示了 JSON 元数据的结构：

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

使用此初始化元数据，播放会话将使用服务器引导的广告插入。

使用 SGAI 进行广告跟踪

SGAI 支持服务器端和客户端广告跟踪。您在会话初始化时设置报告模式。模式在会话期间无法更改。

服务器端跟踪（默认）

MediaTailor 当玩家请求广告区段时，会自动触发 VAST 信标。资产列表 URIs 中的广告包含加密的信标元数据（`awsBeaconData`、`awsBeaconDomain`、`awsConfigurationName`）。玩家必须支持 HLS `#EXT-X-DEFINE:QUERYPARAM` 变量替换。资产列表响应不包含任何`TRACKING`部分。

有关 SGAI 服务器端信标的工作原理的详细信息，请参阅。Server-side 使用服务器引导的广告插入 (SGAI) 进行跟踪

客户端跟踪

`aws.reportingMode=CLIENT`添加到您的会话初始化请求中。素材资源列表响应包括一个带有信标的`TRACKING`部分 URLs ，玩家可以在广告播放期间触发该信标。`GetTracking`API 端点不用于 SGAI 会话。相反，每个资产清单响应都直接包含跟踪数据。跟踪数据使用与服务器端广告插入 (SSAI) 跟踪响应相同的 JSON 架构。

有关更多信息，请参阅 Server-guided 广告插入。

引导式预取，带有明显的心跳声

对于 SGAI 直播，您可以通过在会话初始化请求中添加`aws.guidedPrefetchMode=MANIFEST`以下内容来启用基于清单的广告预取：

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

启用后， MediaTailor 将会话标识符 (`?aws.sessionId=<id>`) 作为查询参数附加到多变体播放列表中的每个插页式媒体清单 (`/v1/i-media`) 网址。每次播放器刷新 i-media 清单时，请求都会到达 MediaTailor 会话 ID，该 MediaTailor 会话 ID 用于为即将到来的广告中断排队预取请求。

- 该`aws.guidedPrefetchMode`参数接受两个值：`MANIFEST`（启用）和`OFF`（禁用，默认）。
- 引导式预取模式仅对 SGAI 会话有效。在拼接会话中使用它会返回错误。
- DASH 尚不支持引导式预取模式。
- 引导式预取与报告模式无关 — 信标在播放时触发，而不是在预取时触发。
- 使用引导式预取时，请勿在 CDN 中缓存 i-media 清单。预取机制取决于玩家 MediaTailor 直接收到的清单刷新请求。如果您的 CDN 缓存了`/v1/i-media`响应，则 MediaTailor 不会收到心跳请求，也无法触发预取。

MediaTailor 服务器引导的广告插入功能兼容性矩阵

AWS Elemental MediaTailor 提供了两种具有不同功能兼容性的广告插入方法。服务器引导式广告插入与服务器端广告插入的工作方式不同，后者会影响与某些 MediaTailor 功能的兼容性。使用此表来了解每种广告插入方法都有哪些功能。

按广告插入方法划分的功能兼容性

功能	服务器端广告插入 (SSAI)	服务器引导式广告插入 (SGAI)
广告预取	✓ 支持	暂不支持
广告抑制	✓ 支持	不适用
片前广告行为	由 MediaTailor 配置控制	由 MediaTailor 配置控制
客户端广告跟踪	使用 GetTracking API	资产列表中的 `TRACKING` “使用” 部分（未使用

功能	服务器端广告插入 (SSAI)	服务器引导式广告插入 (SGAI)
		GetTracking API ）
服务器端广告跟踪	✓ 支持 — 信标根据使用会话 ID 的/v1/segment 请求触发	✓ 支持（仅限 HLS ）— 使用无会话信标，在广告中嵌入加密信标数据。 URIs #EXT-X-DEFINE:QUERYPARAM 需要 HLS v11 或更高版本。尚不支持达世币。
广告标识装饰	✓ 支持	✗ 不兼容

兼容性详情

广告预取

目前不支持广告预取。

广告抑制

服务器引导的广告插入方法不支持广告抑制，因为玩家只能获取他们要播放的广告。

片前广告行为

不同的插入方法不同，片前广告的时机有所不同：

- 服务器端广告插入：根据 MediaTailor 配置设置控制何时播放前置广告
- 服务器引导的广告插入：在清单的顶部插入 MediaTailor 片前广告。你的玩家会先展示这些广告，然后开始播放你的内容

广告跟踪

根据广告插入方法，客户端跟踪使用不同的机制：

- 服务器端广告插入 (SSAI)：使用 API 端点 GetTracking

- 服务器引导式广告插入 (SGAI)：在每个资产列表响应的TRACKING部分 MediaTailor 提供跟踪信息。未使用 GetTracking API 端点。会话初始化响应不包括trackingUrl。

服务器端跟踪也因方法而异：

- 服务器端广告插入 (SSAI)：当玩家使用会话 ID 获取拼接的广告段时，会 MediaTailor 触发信标。/v1/segment/
- 服务器引导广告插入 (SGAI)： MediaTailor 使用无会话信标。 MediaTailor 在广告清单中嵌入加密的信标数据 (awsBeaconDataawsBeaconDomain、、awsConfigurationName) URIs ，然后在资产列表中返回这些数据。广告清单使用#EXT-X-DEFINE:QUERYPARAM标签，因此玩家可以将这些值替换为细分 URLs。当玩家请求每个广告段时， MediaTailor 会解密数据，触发相应的信标，然后重定向到内容区段。当服务器端报告处于活动状态时， MediaTailor 在资产列表响应中省略该TRACKING部分。有关更多信息，请参阅 Server-side 使用服务器引导的广告插入 (SGAI) 进行跟踪。

广告标识装饰

广告-ID 装饰与服务器引导的广告插入不兼容，因为填充 X-AD-CREATIVE-SIGNALING标题的字段只有在获取素材资源列表时才会知道，而不是在写入清单时才知道。

MediaTailor 服务器引导的直播广告插入配置

AWS Elemental MediaTailor 服务器引导的直播内容广告插入通过可缓存的清单带来显著的性能优势。为直播内容配置 SGAI 使用的核心参数与 VOD 相同，但需要特别考虑直播特性和实时处理。

直播 SGAI 的要求

在为直播内容启用 SGAI 之前，请确保具备以下条件：

- 您的直播包含格式正确的 DATERANGE 标记
- 广告时长一致且可预测
- 您的 CDN 已配置为适当地缓存 SGAI 清单
- 玩家支持服务器引导的广告插入工作流程
- 您的广告决策服务器可以处理直播内容的实时请求

玩家要求

必须将玩家配置为正确处理 SGAI 直播清单：

- Support 支持服务器引导的广告插入工作流程
- 能够处理清单中的广告插入指导
- 正确处理直播定时和同步
- 对于 HLS 内容：支持 HLS 版本 8 并 EXT-X-DATERANGE带有 CLASS 属性。服务器端信标的版本 11。
- 对于 HLS 内容：支持 EXT-X-DEFINE变量替换

直播回放配置

要为直播内容启用 SGAI，请创建具有以下设置的播放配置：

Example SGAI 直播回放配置

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

以下是实时 SGAI 配置的关键注意事项：

`VideoContentSourceUrl`

必须指向带有格式正确的 SCTE-35 DATERANGE 标记的直播 HLS 直播。直播应保持一致的片段持续时间和比特率变体。

SGAI 实时清单请求

SGAI 直播清单使用与传统广告插入相同的网址模式：

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

实时 SGAI 的基于清单的预取

对于实时 SGAI 工作流程，您可以启用基于清单的预取检测信号，以减少广告填充延迟。在清单`aws.guidedPrefetchMode=MANIFEST`请求中添加：

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

启用后， MediaTailor 将会话标识符 (`?aws.sessionId=<id>`) 作为查询参数附加到多变体播放列表中的每个插页式媒体清单 (`/v1/i-media`) 网址。每次玩家刷新 i-media 清单时，请求都会到达 MediaTailor会话 ID，该 MediaTailor 会话 ID 用于识别会话，并将预取请求排队等候即将到来的广告时段。

> **Important**
>
> 使用引导式预取时，请勿在 CDN 中缓存 i-media 清单。预取心跳机制取决于玩家直接收到的清单刷新请求。 MediaTailor 如果您的 CDN 缓存并提供`/v1/i-media`响应，则 MediaTailor 不会收到心跳请求，也无法触发预取。将您的 CDN 配置为在使用 MediaTailor `aws.guidedPrefetchMode=MANIFEST`时传递`/v1/i-media/*`请求。

引导式预取与报告模式无关。无论您使用服务器端（默认）还是客户端（`aws.reportingMode=CLIENT`）跟踪，信标都会在播放时触发，而不是在预取广告时触发。有关广告预取工作原理的一般信息 MediaTailor，请参阅。预取广告

测试 SGAI 实时配置

使用以下验证步骤验证您的 SGAI 实时设置：

1. 生成测试清单

   请求 SGAI Live 清单网址，并验证其返回的可缓存内容以及正确的广告插入指导。

2. 验证 CDN 缓存

   检查您的 CDN 是否正在根据配置的 TTL 值缓存 SGAI 清单。

3. 测试广告插入

   根据 SGAI 清单中提供的指南，确认玩家可以成功插入广告。

4. 监控性能

   使用 CloudWatch 指标来验证源负载减少和缓存命中率的提高。

MediaTailor VOD 内容的服务器引导式广告插入配置

AWS Elemental MediaTailor 通过高度可缓存的清单和减少服务器处理量，VOD 内容的服务器引导式广告插入可显著提高性能。为 VOD 内容配置 SGAI 可以利用 video-on-demand资产的静态特性来最大限度地提高缓存效率并最大限度地减少源请求，因此非常适合具有重复观看模式的大型内容库。

VOD SGAI 的要求

在为 VOD 内容启用 SGAI 之前，请确保满足以下条件：

- 您的 VOD 内容包含格式正确的广告标记（SCTE-35 或定时元数据）
- 内容存储在可靠的来源中，且可用性始终如一
- 您的 CDN 已配置为缓存带有相应的 TTL 值的 SGAI 清单
- 玩家支持服务器引导的广告插入工作流程
- 您的广告决策服务器可以处理特定于 VOD 的元数据和定位

玩家要求

必须将玩家配置为处理 SGAI VOD 清单和广告插入：

- Support 支持服务器引导的广告插入工作流程
- 能够处理 VOD 清单中的广告插入指导
- Support 支持在 VOD 播放期间插入客户端广告
- 正确处理广告时段内的搜索操作
- Support 支持内容时长和位置跟踪

VOD 播放配置

要为 VOD 内容启用 SGAI，请创建具有以下设置的播放配置：

Example SGAI VOD 播放配置

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

以下是 VOD SGAI 配置的关键注意事项：

`VideoContentSourceUrl`

应使用一致的网址模式指向你的 VOD 内容库。确保源站能够处理预期的请求量并提供可靠的内容传输。

`ConfigurationAliases`

包括特定于 VOD 的参数，例如内容时长、流派或系列信息，这些参数可用于广告定位，而不会影响清单的可缓存性。

`ManifestProcessingRules`

启用广告标记直通以保留原创内容的时序信息，这对于具有预定义广告插播位置的 VOD 内容尤其重要。

SGAI VOD 清单请求

SGAI VOD 清单使用与传统 VOD 广告插入相同的网址模式。

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

特定于 VOD 的广告定位

VOD 内容提供了独特的广告定位机会：

内容元数据定位

利用 VOD 内容元数据改善广告定位：

- 类型和类别：根据内容类型（剧情、喜剧、纪录片）定位广告
- 内容分级：确保广告内容与年龄相符（G、PG、R 评级）
- 系列和季节：针对连续剧或相关内容投放广告
- 发布日期：基于内容年龄（新版本与目录内容）进行定位
- 内容时长：根据内容总长度调整广告加载量

查看情境定位

VOD 观看模式支持情境广告定位：

- 一天中的时间：根据观看内容的时间来定位广告
- 狂看：调整观看多集的用户的广告频率
- 完成率：根据用户的历史内容完成模式进行定位
- 设备类型：优化用于观看设备（电视、手机、平板电脑）的广告格式

测试 SGAI VOD 配置

使用以下验证步骤验证您的 SGAI VOD 设置：

1. 生成测试清单

   请求不同内容类型的 SGAI VOD 清单 URLs ，并通过适当的广告插入指导验证它们返回可缓存的内容。

2. 验证 CDN 缓存

   检查您的 CDN 是否根据配置的 TTL 值缓存 SGAI 清单并实现较高的缓存命中率。

3. 测试广告插播情况

   根据各种 VOD 内容的 SGAI 清单中提供的指南，确认玩家可以成功插入广告。

4. 测试寻道操作

   验证在 VOD 内容中搜索是否能在广告间隔中正常运行，并保持正确的播放位置。

5. 监控性能

   使用 CloudWatch 指标来验证源站负载减少、缓存命中率提高以及广告成功插入率。

关键测试场景

测试以下特定的 VOD 场景：

- 热门内容：验证经常访问的 VOD 资产的高缓存命中率
- 长篇内容：测试带有多个广告插播时间的电影或长剧集中的广告插入
- 系列内容：验证系列中各剧集的广告定位一致性
- 不同类型：根据内容元数据和流派测试广告定位

VOD SGAI 优化最佳实践

优化 SGAI VOD 实现以获得最佳性能：

缓存优化

- 最大化 TTL 值：由于内容不会改变，因此对 VOD 清单使用更长的缓存持续时间
- 最小化缓存密钥：减少缓存密钥的变化以提高命中率
- 预热热门内容：缓存热门或精选 VOD 内容的清单
- 监控缓存性能：跟踪缓存命中率并根据使用模式进行优化

内容分发优化

- 一致的 URL 模式：使用可预测的 URL 结构实现更好的缓存
- 元数据标准化：确保内容元数据一致，实现可靠的广告定位
- 广告插播定位：优化广告插播位置，实现自然的内容转换
- 质量变体：确保 SGAI 适用于您的 VOD 内容的所有比特率变体

MediaTailor 广告服务器集成要求

要将您的广告服务器与集成 AWS Elemental MediaTailor，您的广告服务器必须发送符合支持版本的 VAST 和 VMAP 的 IAB 规范的 XML。您可以使用公有 VAST 验证器确保标签的格式正确。

AWS Elemental MediaTailor 支持来自广告决策服务器的 VAST 和 VMAP 响应。 AWS Elemental MediaTailor 还支持通过我们的客户端报告 API 代理 VPAID 元数据，用于客户端广告插入。有关客户端报告的信息，请参阅Client-side 广告追踪。

MediaTailor 支持以下版本的 VAST、VMAP 和 VPAID：

- 最高 V AST 4.3

  MediaTailor 接受 VAST 4.3 以上的响应版本，但不支持 VAST 4.0 及更高版本的某些高级功能。
- VMAP 1.0
- VPAID 2.0

大量的要求

广告服务器的 VAST 响应必须包含符合 IAB 的 `TrackingEvents` 元素和标准事件类型，例如 `impression`。如果您包含非标准的跟踪事件，则 AWS Elemental MediaTailor 会拒绝 VAST 响应，并且不提供广告以供使用。

VAST 3.0 引入了对广告连播的支持，广告连播是一组线性广告序列的投放。如果广告栏中的特定广告不可用， AWS Elemental MediaTailor 则会在广告的互动日志中记录错误。 CloudWatch然后，它会尝试在连播中插入下一个广告。通过这种方式， MediaTailor 遍历窗格中的广告，直到找到可以使用的广告。

设置目标

要为您的广告定位特定的玩家，您可以为广告代码创建模板，然后 URLs。有关更多信息，请参阅 MediaTailor ADS 请求的动态广告变量。

AWS Elemental MediaTailor 当玩家发送广告服务器 VAST `x-forwarded-for` 请求`user-agent`以及进行服务器端跟踪调用时，代理播放器的标题和标头。确保您的广告服务器可以处理这些标头。或者，您也可以使用 `[session.user_agent]` 或 `[session.client_ip]` 并在广告标签或广告 URL 上的查询字符串中传递这些值。有关更多信息，请参阅 MediaTailor ADS 请求的会话变量。

广告电话

AWS Elemental MediaTailor 按照您的配置中的定义调用您的 VAST 广告网址。在进行广告调用时，它会替换任何特定于玩家或会话特定的参数。 MediaTailor 在 VAST 响应中遵循多达七个级别的 VAST 包装器和重定向。在直播场景中 MediaTailor ，连接的玩家在广告开始时同时拨打广告电话。实际上，

由于抖动，这些广告调用可能会相差几秒钟。请确保您的广告服务器能够处理此类呼叫所需的并发连接数。 MediaTailor 支持为实时工作流程预取 VAST 响应。有关更多信息，请参阅 预取广告。

创造性的操控

当 AWS Elemental MediaTailor 收到 ADS VAST 响应时，它会为每个广告素材识别`MediaFile`出转码的最高比特率并将其用作来源。它将此文件发送到 on-the-fly转码器，以便转换为适合玩家多变体播放列表比特率和分辨率的格式副本。为了获得最佳效果，请确保比特率最高的媒体文件是具有有效清单预设的高质量 MP4 资产。当清单预设无效时，转码任务将失败，从而导致广告无法显示。无效的预设示例包括不支持的输入文件格式（例如）和某些格式副本规范 ProRes，例如分辨率 855X481。

有关媒体文件输入支持的格式列表，请参阅《AWS Elemental MediaConvert 用户指南》中的 “支持的输入格式” MP4行。

创意索引

AWS Elemental MediaTailor 根据`<Creative>`元素中提供的`id`属性的值对每个广告素材进行唯一索引。如果未指定广告素材的 ID，则 MediaTailor 使用媒体文件网址作为索引。

以下示例声明显示了广告素材 ID。

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

如果您定义自己的广告素材 IDs，请为每个广告素材使用一个新的、唯一的 ID。不要重复使用创意 IDs。 AWS Elemental MediaTailor 存储创意内容以供重复使用，并按其索引 ID 查找每个内容。当新广告素材传入时，该服务首先会对照索引检查其 ID。如果 ID 存在， MediaTailor 将使用存储的内容，而不是重新处理传入的内容。如果您重复使用广告素材 ID，则会 MediaTailor 使用存储的较旧广告，并且不会播放您的新广告。

广告服务合作伙伴提供的 VAST 扩展程序

为了帮助防止与广告素材发生冲突 IDs，您可以使用广告投放合作伙伴提供的扩展程序来响应 VAST 响应。 MediaTailor 支持来自 SpringServe、Publica 和 FreeWheel的扩展。启用 VAST 扩展覆盖后，会将默认的广告素材 ID MediaTailor 替换为扩展值。

要启用此功能，请提交 Su AWS pport 工单，请求启用基于 VAST 扩展的广告素 IDs 材。在 Support 请求中包含以下信息：

- AWS 区域

- AWS 账号
- MediaTailor 播放配置名称

要验证您的账户 IDs 是否启用了基于 VAST 扩展的广告素材，我们建议您同时请求在暂存或测试播放配置上启用`RAW_ADS_RESPONSE`日志记录。通过日志记录，您可以查看 ADS 收到的原始 VAST 响应，并确认使用了正确的广告 IDs 素材。

VPAID 要求

VPAID 允许发布商提供高度互动的视频广告并在其货币化流上提供可见性指标。有关 VPAID 的信息，请参阅 V PAID 规范。

AWS Elemental MediaTailor 在相同的广告可用性中支持 V server-side-stitched AST MP4 线性广告 client-side-inserted和 VPAID 互动广告的组合。它保留了它们在 VAST 响应中出现的顺序。MediaTailor 通过最多七个级别的包装器跟踪 VPAID 重定向。客户端报告响应包含未打包的 VPAID 元数据。

要使用 VPAID，请遵循以下准则：

- 为您的 VP MP4 AID 广告素材配置名单。 AWS Elemental MediaTailor 使用您配置的广告位填充 VPAID 广告位，并提供 VPAID 广告元数据供客户玩家投放互动广告。如果您尚未配置画面，则在 VPAID 广告出现时， MediaTailor 会像往常一样通过客户端报告提供广告元数据。它还会记录 CloudWatch 有关丢失的板块的错误。有关更多信息，请参阅MediaTailor slate 广告插入和创建 MediaTailor 播放配置。
- 使用客户端报告。 AWS Elemental MediaTailor 通过我们的客户端报告 API 支持 VPAID。有关更多信息，请参阅 Client-side 广告追踪。

  理论上，可以将默认服务器端报告模式与 VPAID 一起使用。但是，如果您使用服务器端报告，则会丢失有关 VPAID 广告是否存在及其相关元数据的任何信息，因为这些信息只能通过客户端 API 获得。
- 在直播场景中，请确保您的广告可用时间（用表示）足够长`EXT-X-CUE-OUT: Duration`，足以容纳 VPAID 上的任何用户互动。例如，如果 VAST XML 指定的 VPAID 广告时长为 30 秒，则可以考虑将广告投放时间配置为超过 30 秒。这段额外的时间为用户提供了更多与广告互动的机会。如果您不添加时间，则可能会丢失 VPAID 元数据，因为广告可用中的剩余时长不足以容纳 VPAID 广告。

广告自助餐

Interactive Advertising Bureau(IAB) Video Ad Serving Template (VAST) 规范定义了广告窗格和广告自助餐。此机制支持有序广告插入，并在广告插入失败时自动回退。在广告窗格中，按顺序播放广告。独立广告充当自助餐，当顺序广告失败时，可以从 MediaTailor中选择替代广告。

AWS Elemental MediaTailor 通过中的`AdSequencingMode`设置支持此规范`PlaybackConfiguration`。`AdDecisionServerConfiguration`如果设置为`FOLLOW_AD_SEQUENCE`，则按顺序 MediaTailor 插入顺序广告，并且仅在序列广告无法插入时使用独立广告作为替代品。

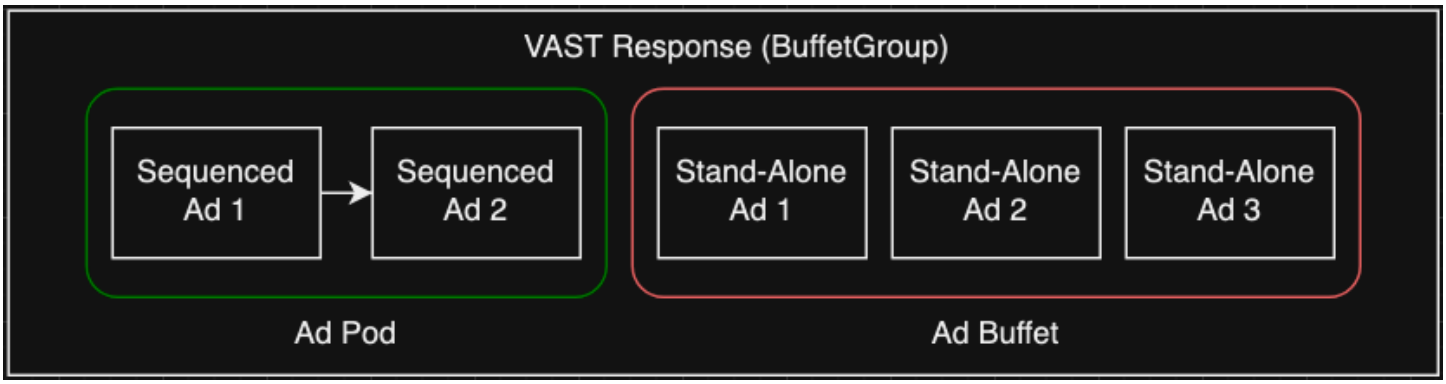
VAST 响应示例

以下示例显示了包含序列广告和独立广告的 VAST 响应：

```
<VAST>
  <Ad sequence="1" id="sequencedAd1">...</Ad>    <!-- Sequenced Ad 1 -->
  <Ad sequence="2" id="sequencedAd2">...</Ad>    <!-- Sequenced Ad 2 -->
  <Ad id="standAloneAd1">...</Ad>                <!-- Standalone Ad 1 -->
  <Ad id="standAloneAd2">...</Ad>                <!-- Standalone Ad 2 -->
  <Ad id="standAloneAd3">...</Ad>                <!-- Standalone Ad 3 -->
</VAST>
```

在此示例中，两个按顺序排列的广告形成一个广告窗格。这三个独立的广告构成了广告自助餐。它们共同构成了一个自助餐组。如果`sequencedAd1`失败，请 MediaTailor 先尝试将其替换为来自同一自助餐组的符合条件的独立广告。如果该群组中没有符合条件的独立广告，请从其他自助餐群组的独立广告中进行 MediaTailor 选择。

下图显示了 VAST 响应，其中包含一个广告窗格（序列广告）和一个广告自助餐（独立广告），它们共同构成一个自助餐组。



重要概念

以下术语描述了广告自助餐的核心概念：

- 序列广告 -<Ad> 元素中带有sequence属性的广告。它们形成一个有序的广告窗格，并按递增的序列顺序插入。
- 独立（自助餐）广告 -没有sequence属性的广告。只有在序列广告无法插入时，它们才用作替代品。
- 自助餐群组 — 在单个 VAST 响应中将广告窗格与其独立广告配 MediaTailor 对。当按顺序排列的广告作为包装时，其重定向的 VAST 会形成自己的独立自助餐组。替换失败的顺序广告时，MediaTailor 优先考虑来自同一自助餐组的独立广告。如果同一自助餐组中不存在符合条件的独立广告， MediaTailor 则退回到其他自助餐组的独立广告。

> **Note**
>
> sequence指的是<Ad>元素级属性，而不是元素中的sequence<Creative>属性。

广告自助餐的好处

如果没有广告自助餐，则 MediaTailor 对所有广告一视同仁，这可能会导致以下问题：

- VOD — 即使没有出现故障，也会插入独立广告，从而破坏商定的广告插播位置并过度填充广告间隔。
- 直播 —— Lower-revenue 独立广告可以在序列广告之前插入，占用有效时长，没有空间投放收入更高的序列广告。

启用广告自助服务后，仅 MediaTailor 插入按顺序排列的广告，并且仅在序列广告无法插入时使用独立广告作为替代品。这样可以确保首先播放收入最高的广告。

> **Warning**
>
> 如果您的广告决策服务器 (ADS) 响应不包含任何按顺序排列的广告（没有任何<Ad>元素具有sequence属性），则启用广告自助餐不会导致广告插入。

启用广告自助餐

广告自助餐是一项可选功能，您可以使用 PutPlaybackConfiguration API 为每种播放配置进行配置。

AdSequencingMode 价值观

下表描述了可用AdSequencingMode值。

值	说明
IGNORE_AD_SEQUENCE （默认值）	MediaTailor 无论序列属性如何，均按照广告在 VAST 响应中出现的顺序插入广告。
FOLLOW_AD_SEQUENCE	MediaTailor 为直播和 VOD 工作流程插入按顺序排列的广告。失败的顺序广告将被独立广告所取代。
FOLLOW_AD_SEQUENCE_ONLY_LIVE	广告自助餐行为仅适用于实时工作流程。
FOLLOW_AD_SEQUENCE_ONLY_VOD	广告自助餐行为仅在 VOD 工作流程中启用。

> **Note**
>
> 当您启用自助广告时，应 MediaTailor 遵循Wrapper.fallbackOnNoAd和Wrapper.allowMultipleAds属性。

Midroll 配置

以下示例为直播和 VOD 工作流程启用了自助广告：

```
{
    "PlaybackConfiguration": {
        "AdDecisionServerConfiguration": {
            "VastResponse": {
                "AdSequencingMode": "FOLLOW_AD_SEQUENCE"
            }
```

```
        }
    }
}
```

以下示例仅为实时工作流程启用广告自助餐：

```
{
    "PlaybackConfiguration": {
        "AdDecisionServerConfiguration": {
            "VastResponse": {
                "AdSequencingMode": "FOLLOW_AD_SEQUENCE_ONLY_LIVE"
            }
        }
    }
}
```

以下示例仅为 VOD 工作流程启用广告自助餐：

```
{
    "PlaybackConfiguration": {
        "AdDecisionServerConfiguration": {
            "VastResponse": {
                "AdSequencingMode": "FOLLOW_AD_SEQUENCE_ONLY_VOD"
            }
        }
    }
}
```

直播预播配置

以下示例为实时预投广告启用了广告自助餐：

```
{
    "PlaybackConfiguration": {
        "LivePrerollConfiguration": {
            "AdDecisionServerConfiguration": {
                "VastResponse": {
                    "AdSequencingMode": "FOLLOW_AD_SEQUENCE"
                }
            }
        }
    }
```

```
}
```

> **Note**
>
> LivePrerollConfiguration仅支持FOLLOW_AD_SEQUENCE和IGNORE_AD_SEQUENCE。通过 midroll 配置为 VOD 预投配置广告自助餐。

广告自助餐的运作方式

广告替换条件

如果按顺序排列的广告遇到以下失败案例之一，则 MediaTailor尝试使用来自同一自助餐组的符合条件的独立广告替换排序后的广告。如果同一自助餐组中没有独立广告，则 MediaTailor 使用其他自助餐组中的独立广告（如果有）。

失败案例	说明
VAST 解析失败	由于缺少必需的 VAST 元素（例如、、、缺失或Impression 无效）MediaFiles Creatives ，按顺序排列的广告无法解析。MediaFile VASTAdTagURI
VAST 封装器解析失败	重定向超时、失败或返回空响应。
VPAID 广告已下线	按顺序排列的广告是 VPAID，会话使用服务器端报告或未配置 Slate。
广告转码尚未准备就绪	顺序广告的转码状态不是 “成功”（例如，IN_PROGRESS 或 ERROR）。
已超过可用时长	顺序广告的持续时间超过剩余的使用时长（实时工作流程）。

独立广告资格

要获得替代资格，独立广告必须满足以下标准：

- 不要成为 VPAID 广告
- 完成转码
- 持续时间与剩余可用持续时间相符

独立广告精选

当按顺序排列的广告失败时，使用以下优先级从符合条件的独立广告 MediaTailor 中选择替代品：

1. 首先是相同的自助餐群组 — MediaTailor 首选失败广告自己的自助餐组中的独立广告。
2. 其他自助餐群组 -仅在同一自助餐组中不存在符合条件的广告时使用。

如果未找到符合条件的独立广告，则该职位将空缺并发布相应SkippedReason的广告。

VAST 包装器行为

对包装器重定向响应的处理 MediaTailor 取决于两个包装器变量的值：Wrapper.fallbackOnNoAd和。Wrapper.allowMultipleAds

- fallbackOnNoAd当重定向响应为空、超时或在 ADS 请求期间发生错误时生效。
- allowMultipleAds如果 ADS 响应包含一个或多个广告，则生效。

VAST 为空、请求错误或请求超时

下表描述了重定向响应为空、超时或发生错误时的 MediaTailor 行为。

后退 OnNoAd	MediaTailor 行为
true（默认值）	MediaTailor 选择独立广告来替换失效的包装纸，首选相同的自助餐组。如果没有符合条件的同组广告，请从其他自助餐群组中 MediaTailor 选择。
false	MediaTailor 删除广告位置并移至窗格中的下一个广告。不进行更换。

Example fall OnNoAd back =真（重定向失败）

家长 VAST：

```
<VAST>
  <Ad sequence="1" id="seqAd1">...</Ad>
  <Ad sequence="2">
    <Wrapper fallbackOnNoAd="true">
      <VASTAdTagURI>https://ads.example.com/vast</VASTAdTagURI>
```

```
    </Wrapper>
  </Ad>
  <Ad id="standAloneAd1">...</Ad>
  <Ad id="standAloneAd2">...</Ad>
</VAST>
```

包装器重定向响应：VAST 为空、超时或错误。

已解决 VAST：

```
<VAST>
  <Ad sequence="1" id="seqAd1">...</Ad>
  <!-- standAloneAd1 replaced failed Wrapper Ad and now has the Wrapper's sequence (2)
 -->
  <Ad sequence="2" id="standAloneAd1">...</Ad>
  <Ad id="standAloneAd2">...</Ad>
</VAST>
```

播放顺序：

1. seqad1
2. 展台 AloneAd 1

Example fall OnNoAd back =false（重定向失败）

家长 VAST：

```
<VAST>
  <Ad sequence="1" id="seqAd1">...</Ad>
  <Ad sequence="2">
    <Wrapper fallbackOnNoAd="false">
      <VASTAdTagURI>https://ads.example.com/vast</VASTAdTagURI>
    </Wrapper>
  </Ad>
  <Ad id="standAloneAd1">...</Ad>
  <Ad id="standAloneAd2">...</Ad>
</VAST>
```

包装器重定向响应：VAST 为空、超时或错误。

已解决 VAST：

```
<VAST>
  <Ad sequence="1" id="seqAd1">...</Ad>
  <!-- sequence 2 is not replaced -->
  <Ad id="standAloneAd1">...</Ad>
  <Ad id="standAloneAd2">...</Ad>
</VAST>
```

播放顺序：

1. seqad1

回复包含一个或多个广告

下表描述了重定向响应包含一个或多个广告时的 MediaTailor 行为。

允许 MultipleAds	MediaTailor 行为
`false`（默认值）	MediaTailor 从重定向响应中选择一个广告并丢弃剩余的广告。所选广告继承父包装广告的序列值。选择策略： MediaTailor 选择重定向响应中顺序最低的广告。如果没有按顺序排列的广告可用，则 MediaTailor 选择独立广告（如果有）。
`true`	当包装器具有序列属性时， MediaTailor 会在包装器的位置插入重定向广告。当包装器没有序列属性时，在父级的排序广告之后 MediaTailor 插入重定向广告。已解析的广告构成了自己的自助餐组。

Example允许 MultipleAds =false，包装器的序列为 2

家长 VAST：

```
<VAST>
  <Ad sequence="1" id="seqAd1">...</Ad>
  <Ad sequence="2">
    <Wrapper allowMultipleAds="false">
      <VASTAdTagURI>https://ads.example.com/vast</VASTAdTagURI>
    </Wrapper>
  </Ad>
  <Ad sequence="3" id="seqAd3">...</Ad>
  <Ad id="standAloneAd1">...</Ad>
```

```
  <Ad id="standAloneAd2">...</Ad>
</VAST>
```

包装器重定向响应：

```
<VAST>
  <Ad sequence="1" id="redirectSeqAd1">...</Ad>
  <Ad sequence="2" id="redirectSeqAd2">...</Ad>
  <Ad id="redirectStandAloneAd1">...</Ad>
</VAST>
```

已解决 VAST：

```
<VAST>
  <Ad sequence="1" id="seqAd1">...</Ad>
  <Ad sequence="2" id="redirectSeqAd1">...</Ad> <!-- redirectSeqAd1 selected, inherits
 Wrapper's sequence (2) -->
  <Ad sequence="3" id="seqAd3">...</Ad>
  <Ad id="standAloneAd1">...</Ad>
  <Ad id="standAloneAd2">...</Ad>
</VAST>
```

播放顺序：

1. seqad1
2. 重定向 SeqAd 1
3. SeqAd3

Example允许 MultipleAds =false，包装器没有序列

家长 VAST：

```
<VAST>
  <Ad sequence="1" id="seqAd1">...</Ad>
  <Ad>
    <Wrapper allowMultipleAds="false">
      <VASTAdTagURI>https://ads.example.com/vast</VASTAdTagURI>
    </Wrapper>
  </Ad>
  <Ad sequence="3" id="seqAd3">...</Ad>
```

```
  <Ad id="standAloneAd1">...</Ad>
  <Ad id="standAloneAd2">...</Ad>
</VAST>
```

包装器重定向响应：

```
<VAST>
  <Ad sequence="1" id="redirectSeqAd1">...</Ad>
  <Ad sequence="2" id="redirectSeqAd2">...</Ad>
  <Ad id="redirectStandAloneAd1">...</Ad>
</VAST>
```

已解决 VAST：

```
<VAST>
  <Ad sequence="1" id="seqAd1">...</Ad>
  <Ad id="redirectSeqAd1">...</Ad> <!-- redirectSeqAd1 selected, inherits Wrapper's
 sequence (UNDEFINED) -->
  <Ad sequence="3" id="seqAd3">...</Ad>
  <Ad id="standAloneAd1">...</Ad>
  <Ad id="standAloneAd2">...</Ad>
</VAST>
```

播放顺序：

1. seqad1
2. SeqAd3

> **Note**
>
> 重定向 SeqAd 1 无法播放，因为它没有定义序列。

Example允许 MultipleAds =true，包装器的序列=2

家长 VAST：

```
<VAST>
  <Ad sequence="1" id="seqAd1">...</Ad>
  <Ad sequence="2">
```

```
    <Wrapper allowMultipleAds="true">
      <VASTAdTagURI>https://ads.example.com/vast</VASTAdTagURI>
    </Wrapper>
  </Ad>
  <Ad sequence="3" id="seqAd3">...</Ad>
  <Ad id="standAloneAd1">...</Ad>
  <Ad id="standAloneAd2">...</Ad>
</VAST>
```

包装器重定向响应：

```
<VAST>
  <Ad sequence="1" id="redirectSeqAd1">...</Ad>
  <Ad sequence="2" id="redirectSeqAd2">...</Ad>
  <Ad id="redirectStandAloneAd1">...</Ad>
</VAST>
```

已解决 VAST：

```
<VAST>
  <Ad sequence="1" id="seqAd1">...</Ad>
  <!-- Redirect response merged at sequence 2 -->
  <Ad sequence="1" id="redirectSeqAd1">...</Ad>
  <Ad sequence="2" id="redirectSeqAd2">...</Ad>
  <Ad id="redirectStandAloneAd1">...</Ad>
  <Ad sequence="3" id="seqAd3">...</Ad>
  <Ad id="standAloneAd1">...</Ad>
  <Ad id="standAloneAd2">...</Ad>
</VAST>
```

播放顺序：

1. seqad1
2. 重定向 SeqAd 1
3. 重定向 SeqAd 2
4. SeqAd3

Example MultipleAdsallow =true，包装器没有序列

家长 VAST：

```
<VAST>
  <Ad sequence="1" id="seqAd1">...</Ad>
  <Ad>
    <Wrapper allowMultipleAds="true">
      <VASTAdTagURI>https://ads.example.com/vast</VASTAdTagURI>
    </Wrapper>
  </Ad>
  <Ad sequence="3" id="seqAd3">...</Ad>
  <Ad id="standAloneAd1">...</Ad>
  <Ad id="standAloneAd2">...</Ad>
</VAST>
```

包装器重定向响应：

```
<VAST>
  <Ad sequence="1" id="redirectSeqAd1">...</Ad>
  <Ad sequence="2" id="redirectSeqAd2">...</Ad>
  <Ad id="redirectStandAloneAd1">...</Ad>
</VAST>
```

已解决 VAST：

```
<VAST>
  <Ad sequence="1" id="seqAd1">...</Ad>
  <Ad sequence="3" id="seqAd3">...</Ad>
  <!-- Redirect response merged after sequenced ads -->
  <Ad sequence="1" id="redirectSeqAd1">...</Ad>
  <Ad sequence="2" id="redirectSeqAd2">...</Ad>
  <Ad id="redirectStandAloneAd1">...</Ad>
  <Ad id="standAloneAd1">...</Ad>
  <Ad id="standAloneAd2">...</Ad>
</VAST>
```

播放顺序：

1. seqad1
2. SeqAd3
3. 重定向 SeqAd 1
4. 重定向 SeqAd 2

指标和日志监控

对现有指标的更改

启用广告自助餐后，以下现有指标会改变行为。

指标	广告自助餐的行为
AdDecisionServer.Ads	从 ADS 响应中解析出的顺序广告的数量。
AdDecisionServer.Duration	仅限顺序广告的总时长。
AdDecisionServer.FillRate	填充率仅使用顺序广告计算。

新指标

启用广告自助餐后，以下新指标可用。

指标	说明
AdNotReady.SequencedAd	尚未准备好插入的序列广告的数量。
AdNotReady.StandaloneAd	尚未准备好插入的独立广告的数量（不包括跳过的StandaloneAdNotNeeded 广告）。
Avail.FillRate.SequencedAd	按顺序排列的广告填补的收益百分比。
Avail.FillRate.StandaloneAd	独立广告填补的收益百分比。
Avail.FilledDuration.SequencedAd	无效期限充斥着按顺序排列的广告。

指标	说明
Avail.FilledDuration.StandaloneAd	使用期限充斥着独立广告。
AdsBilled.SequencedAd	按顺序计费的广告数量。
AdsBilled.StandaloneAd	独立付费广告的数量。
SkippedReason.StandaloneAdNotNeeded	由于所有序列广告均已成功插入，因此不需要独立广告时发出。

新跳过广告的原因

启用广告自助餐后，以下新的跳过广告原因可用。

Reason	说明
STANDALONE_AD_NOT_NEEDED	独立广告符合条件，但不是必需的，因为所有按顺序排列的广告均已成功插入。未触发错误信标。

对于不符合插入资格的独立广告（例如，VPAID 或转码未准备就绪），实际失败原因将以（例如，TRANSCODE_IN_PROGRESS或VPAID_AD）的形式发布。SkippedReason

信标时出错

MediaTailor 按如下方式处理广告自助餐的错误信标：

- 失败的顺序广告——触发错误信标（现有行为）。
- 尚未准备就绪的独立广告（广告转码状态不成功）——会触发错误信标（与序列广告相同）。
- 符合插入条件但不需要的未使用独立广告——不会触发错误信标。

ADS 活动变更

启用广告自助餐后， MediaTailor 将以下字段添加到两个 ADS 事件中：

VAST_RESPONSE 事件

VAST_RESPONSE事件VastAd中的每个字段都包含以下字段。

字段	Type	说明
buffetGroup	字符串	MediaTailor 生成的字符串，用于定义广告属于哪个自助餐组。
sequence	Int	Ad元素的序列值。

FILLED_AVAIL 事件

活动中creativeAds和FILLED_AVAIL活动skippedAds中的每个条目都包含以下字段。

字段	Type	说明
buffetGroup	字符串	MediaTailor 生成的字符串，用于定义广告属于哪个自助餐组。
sequence	Int	Ad元素的序列值。

这些字段允许您确定：

- 哪些广告是按顺序解析的，哪些是独立的
- 插入了哪些广告及其类型
- 为什么跳过了序列广告，以及是否由独立广告取而代之
- 为什么独立广告不可用

另请参阅

有关 PutPlaybackConfiguration API 和 IAB VAST 规范的更多信息，请参阅以下内容：

- PutPlaybackConfiguration API 参考 — 用于配置AdSequencingMode和相关设置的完整参数参考。

- IAB VAST 4.x SpecificationIAB Tech Lab网站上 — 定义广告窗格、序列广告和独立广告的行业标准。
- 监控 AWS Elemental MediaTailor 使用亚马逊 CloudWatch 指标— 所有 MediaTailor CloudWatch 指标的完整参考，包括新的广告自助餐指标。

MediaTailor 回放配置管理

本节介绍管理 AWS Elemental MediaTailor 回放配置的关键任务。您可以学习如何创建新配置来设置内容流并为播放设备提供访问权限，查看现有配置的详细信息，编辑配置以更新原始服务器和广告决策服务器等设置，以及删除不再需要的配置。

主题

- 创建 MediaTailor 播放配置
- 查看 MediaTailor 配置详情
- 编辑 MediaTailor 配置设置
- 删除 MediaTailor 配置

创建 MediaTailor 播放配置

本主题介绍如何创建用于开始接收内容流的配置 AWS Elemental MediaTailor。它还展示了如何为下游播放设备提供请求内容的接入点。

您可以使用 AWS Elemental MediaTailor 控制台、 AWS Command Line Interface (AWS CLI) > 或 MediaTailor API 来创建配置。有关通过 AWS CLI 或 MediaTailor API 创建配置的信息，请参阅 AWS Elemental MediaTailor API 参考。

创建配置时，不要将敏感的识别信息放入自由格式字段，例如配置名称字段。识别信息可能包括客户账号之类的信息。此外，在 MediaTailor控制台、REST API、或 AWS SDK 中工作时，请勿使用识别信息。 AWS CLI您输入的任何数据都 MediaTailor 可能会被提取并包含在诊断日志或亚马逊 CloudWatch 活动中。

添加配置（控制台）

1. 在处打开 MediaTailor 控制台https://console.aws.amazon.com/mediatailor/。
2. 在 Configurations (配置) 页面上，选择 Create configuration (创建配置)。
3. 填写配置和其他配置字段，如以下主题所述：

- 必需的设置
- 可选配置设置

4. 选择创建配置。

   AWS Elemental MediaTailor 在 “配置” 页面的表格中显示新配置。

5. （推荐）使用 AWS Elemental MediaTailor 为清单和报告请求设置 CDN。您可以使用配置播放 URL 进行 CDN 设置。有关为清单和报告请求设置 CDN 的信息，请参阅使用 CDN 优化 MediaTailor 广告个性化和内容交付。

必需的设置

当您创建配置时，必须包括以下必需的设置。

Name

输入描述配置的唯一名称。此名称是配置的主要标识符。允许的最大长度为 512 个字符。

内容来源

输入此流的清单的 URL 前缀（去掉资产 ID）。最大长度为 512 个字符。

例如，网址前缀对`http://origin-server.com/a/`的 HLS 多变量播放列表网址`http://origin-server.com/a/main.m3u8`和的 DASH MPD 网址有效。`http://origin-server.com/a/dash.mpd`或者，您可以输入一个较短的前缀（如 `http://origin-server.com`），但 `/a/` 必须包含在播放器内容请求的资产 ID 中。

> **Note**
>
> 如果您的内容源使用 HTTPS，则其证书必须来自知名的证书颁发机构。它不能是自签名证书。如果你使用自签名证书， AWS Elemental MediaTailor 则无法连接到内容来源，也无法提供清单来响应玩家的请求。

广告决策服务器

输入您的广告决策服务器 (ADS) 的 URL。这是包含变量的 URL（如第 3 步：配置 ADS 请求 URL 和查询参数中所述），或用于测试用途的静态 VAST URL。最大长度为 25000 个字符。

> **Note**
>
> 如果您的 ADS 使用 HTTPS，则其证书必须来自知名的证书颁发机构。它不能是自签名证书。这同样也适用于 ADS 返回的夹层广告 URL。如果您使用自签名证书，则 AWS Elemental MediaTailor 无法从内容来源检索广告并将其拼接到清单中。

可选配置设置

您可以选择在 MediaTailor 控制台、 MediaTailor API 或 AWS Command Line Interface (AWS CLI) 中配置配置别名、个性化详细信息和高级设置。

配置别名

以下是可选的配置别名，您可以在 MediaTailor 控制台中或使用 MediaTailor API 进行配置。

玩家参数变量

要在会话初始化期间进行动态域配置，请添加一个或多个玩家参数变量。

有关使用玩家参数变量动态配置域名的更多信息，请参阅MediaTailor 多个内容源的域变量。

日志配置

以下是日志配置设置。

已启用百分比

设置 MediaTailor 写入日志的回放配置会话 CloudWatch 日志的百分比。例如，如果您的播放配置有 1000 个会话，并且您将启用百分比设置为 60，则会将 600 个会话日志 MediaTailor 写入 CloudWatch 日志。

启用此选项后， MediaTailor 将自动创建一个服务相关角色， MediaTailor 允许在您的日志账户中写入和管理会话 CloudWatch 日志。有关更多信息，请参阅 将服务相关角色用于 MediaTailor。

记录策略

表示用于收集 MediaTailor发出的日志的方法。要将日志直接发送到 CloudWatch 日志，请选择`LEGACY_CLOUDWATCH`。要向 Logs 发送 CloudWatch 日志，然后由日志将日志发送到您选择的目的地，请`VENDED_LOGS`选择。支持的目标是 CloudWatch 日志组、Amazon S3 存储桶和亚马逊数据消防队数据流。

出售的日志需要进行额外设置。有关设置，请参阅使用已售日志。

ADS 互动日志选择加入事件

表示 MediaTailor 将为使用此配置初始化的会话发出RAW_ADS_RESPONSE日志。

RAW_ADS_RESPONSE日志事件包含来自 ADS 的整个 VAST 或 VMAP 响应。因此，日志可能很多，可能会增加您的日志记录成本。

ADS 互动日志排除事件

表示 MediaTailor 不会在日志中发出描述与 ADS 交互的选定事件。

有关 ADS 日志事件的描述，请参阅ADS 日志。

清单服务交互日志排除事件

表示 MediaTailor 不会在日志中发出描述与清单服务交互的选定事件。

有关清单服务日志事件的描述，请参阅清单日志。

广告调节

以下内容决定了在将广告拼接到内容流中之前 MediaTailor 需要采取哪些措施来对其进行修改。

流媒体文件调节

确定在决定拼接哪些广告时 MediaTailor 使用的逻辑。

- 当流媒体文件条件设置为 Transcode 时，会在 MediaTailor 传progressive送时对媒体文件进行转码，并将它们拼接到清单中。如果没有足够的带有progressive投放媒体文件的广告来填补空缺，则进行 MediaTailor 转码并使用已streaming投放的媒体文件。
- 当流媒体文件条件设置为 “无” 时，会将带有streaming投放 MediaTailor 媒体文件的广告拼接到清单中，而无需对其进行转码。如果没有足够的带有streaming投放媒体文件的广告来填补空缺，则进行 MediaTailor 转码并使用已progressive投放的媒体文件。

个性化细节

以下是您可以在 MediaTailor 控制台或 MediaTailor API 中配置的个性化详细信息。

画面广告

输入用于填充未填充广告时间的高质量 MP4 素材资源的网址。 MediaTailor 对于大多数工作流程，Slate 配置是可选的，但对于 VPAID 广告来说是强制性的。有关 Slate 行为、配置要求以及何时显示 slate 的全面信息，请参阅。MediaTailor slate 广告插入

起动保险杠

起始缓冲区资产位置的网址。Bumpers 是在广告插播开始或结束时播放的简短视频或音频片段。它们可以存储在亚马逊的S3或其他存储服务上。要了解有关保险杠的更多信息，请参阅。MediaTailor 保险杠广告插入

端部保险杠

末端缓冲区资产位置的网址。Bumpers 是在广告插播开始或结束时播放的简短视频或音频片段。它们可以存储在亚马逊的S3或其他存储服务上。要了解有关保险杠的更多信息，请参阅。MediaTailor 保险杠广告插入

个性化阈值

仅在与 Slate 广告配合使用时适用。定义在 MediaTailor 放弃个性化设置和展示基础内容之前，广告插播时间内允许的最大填充广告时长（以秒为单位）。此功能适用于直播和 VOD 流中的广告替换 而不是广告插入，因为它依赖于基础内容流。有关详细的行为场景和全面的信息，请参阅个性化阈值的工作原理。

实时前置式广告决策服务器

要在主要内容开始播放之前在实时流的开头插入广告，请从广告决策服务器 (ADS) 输入前置式广告的 URL。这是包含变量的 URL（如第 3 步：配置 ADS 请求 URL 和查询参数中所述），或用于测试用途的静态 VAST URL。最大长度为 25000 个字符。

> **Note**
>
> 如果您的 ADS 使用 HTTPS，则其证书必须来自知名的证书颁发机构。它不能是自签名证书。这同样也适用于 ADS 返回的夹层广告 URL。如果您使用自签名证书，则 AWS Elemental MediaTailor 无法从内容来源检索广告并将其拼接到清单中。

有关前置式广告工作方式的信息，请参阅MediaTailor 片前广告插播。

实时前置式广告最大允许持续时间

当您在直播流的开头插入广告时，请输入前置式广告效用的最大允许持续时间。 MediaTailor 插入广告时不会超过此持续时间。如果广告的回复中包含的广告数量超过了该时段的容纳量，则在不超

过持续时间的情况下使用尽可能多的广告来 MediaTailor 填补空缺。有关如何使用 MediaTailor 填充的更多详细信息，请参阅直播广告拼接行为。

可用抑制模式

设置效用禁止模式，也称为广告禁止。默认情况下，广告抑制处于关闭状态，并用广告或广告牌 MediaTailor 填充所有内容。当模式设置为时BEHIND_LIVE_EDGE，广告抑制处于活动状态，MediaTailor 不会填充清单回顾窗口中处于空闲抑制值时段或之后的广告插播时间。当模式设置为时AFTER_LIVE_EDGE，广告抑制处于活动状态。 MediaTailor在无用广告抑制期内或之后填补广告间隔，即实时优势加上无效抑制值加上缓冲时间。

可用抑制值

空闲抑制值是实时边缘偏移时间HH:MM:SS。 MediaTailor 这次不会在清单回顾窗口中填补广告插播时间段或落后时间。

插入模式

插入模式控制玩家是否可以使用拼接广告或引导式广告插入。默认值强制所有玩家会话使用拼接（服务器端）广告插入。STITCHED_ONLY设置 InsertionMode 为PLAYER_SELECT允许玩家在会话初始化时选择拼接广告或引导式广告插入。对于未指定插入模式的玩家，默认设置是拼接的。

高级设置

以下是可选的高级设置。你可以在 MediaTailor 控制台中使用 AWS Command Line Interface (AWS CLI) 或使用 MediaTailor API 进行配置。

CDN 内容分段前缀

AWS Elemental MediaTailor 允许使用内容分段的 CDN 路径的 URL 创建清单。在执行此步骤之前，请在 CDN 中设置规则以从来源服务器中拉取分段。对于 CDN content segment prefix (CDN 内容分段前缀)，输入 CDN 前缀路径。

有关 MediaTailor 与 CDN 集成的更多信息，请参阅使用 CDN 优化 MediaTailor 广告个性化和内容交付。

CDN 广告分段前缀

AWS Elemental MediaTailor 允许使用您自己的 CDN 路径的网址为广告区段创建清单。默认情况下，使用默认缓存设置从亚马逊内部 CloudFront 分配中 MediaTailor提供广告片段。您必须先在 CDN 中设置规则以从以下来源拉取广告分段，然后才能填写 CDN ad segment prefix (CDN 广告分段前缀) 字段，如下例所示。

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

对于 CDN ad segment prefix (CDN 广告分段前缀)，在配置中输入您的 CDN 前缀的名称。

有关 MediaTailor 与 CDN 集成的更多信息，请参阅使用 CDN 优化 MediaTailor 广告个性化和内容交付。

DASH 原始清单类型

如果您的源服务器生成单周期 DASH 清单，请打开下拉列表并选择 SINGLE_PERIOD。默认情况下， MediaTailor 会将 DASH 清单作为多周期清单处理。有关更多信息，请参阅 the section called “整合 MPEG-DASH 来源”。

DASH mpd 位置

（根据 DASH 的需要可选）媒体演示描述 (mpd) 的位置。对于以下情况，请选择 “禁用”：

- 您可以设置用于访问 MediaTailor 清单的 CDN 路由规则。
- 你使用客户端报告，或者你的玩家支持粘性 HTTP 重定向。

有关 Location (位置) 功能的更多信息，请参阅 the section called “位置功能”。

对配置文件名称进行转码

将此配置与自定义转码配置文件关联的名称。此名称覆盖的动态转码默认值。 MediaTailor仅当您已经在 AWS Support 的帮助下设置了自定义配置文件时才填写此字段。

广告标记直通

对于 HLS，启用或禁用广告标记直通。启用广告标记直 MediaTailor 通EXT-X-CUE-IN功能后，会将EXT-X-SPLICEPOINT-SCTE35广告标记从源清单传递到 MediaTailor 个性化清单。EXT-X-CUE-OUT不对广告标记值应用任何逻辑；它们按原样从原始清单传递到个性化清单。例如，如果EXT-X-CUE-OUT来源清单60中的值为，但未投放任何广告，则 MediaTailor 不会将该值更改为个性化清单0中的值。

个人 ADS 请求超时

在直播或 VOD 播放期间解决广告决策时 MediaTailor 等待单个 HTTP 响应的最长时间（以毫秒为单位）。默认值：3000。有效范围：250— 6000。

有关如何在直播和 VOD 工作流程中 MediaTailor 处理广告拼接的更多信息，请参阅直播广告拼接行为和。VOD 的广告拼接行为

此设置对应于PutPlaybackConfigurationAdsPersonalizationTimeouts参数AdsRequestTimeoutMilliseconds的成员。

最大并发 ADS 请求数

MediaTailor 可以向 ADS 发出的同时请求的数量。默认值：1。有效范围：1— 100。

此设置对应于PutPlaybackConfigurationAdsPersonalizationConcurrency参数MaxConcurrentAdsRequests的成员。

直播最长 ADS 个性化时间

实时清单个性化期间所有 ADS 活动的总时间预算，以毫秒为单位。该预算将在单个清单请求中的所有广告时段中共享。默认值：10000。有效范围：250— 10000。

有关直播广告拼接行为的更多信息，请参阅直播广告拼接行为。

此设置对应于PutPlaybackConfigurationAdsPersonalizationTimeouts参数LiveMaximumAdsPersonalizationTimeMilliseconds的成员。

VOD 最长 ADS 个性化时间

VOD 清单个性化期间所有 ADS 活动的总时间预算，以毫秒为单位。该预算将在单个清单请求中的所有广告时段中共享。默认值：10000。有效范围：250— 10000。

有关 VOD 广告拼接行为的更多信息，请参阅。VOD 的广告拼接行为

此设置对应于PutPlaybackConfigurationAdsPersonalizationTimeouts参数VodMaximumAdsPersonalizationTimeMilliseconds的成员。

启用 VOD VAST 并行化

开启此设置后，将同时 MediaTailor 处理返回 VAST 响应的 VOD 工作流程的 ADS 请求。此设置仅适用于 VOD 工作流程。 MediaTailor 默认情况下关闭此设置。要生效，请将最大并发 ADS 请求数设置为大于的值1。

> **⚠ 并发请求的频率上限**
>
> 当您启用 VAST 并行化时，某些广告服务器可能不会对并发广告请求强制执行频率上限限制。请咨询您的广告服务器提供商，确认并行请求条件下是否支持频率上限。如果您的广告服务器不支持并发请求的频率上限，请考虑切换到 VMAP，它可以正确处理并行广告请求的频率上限。

此设置对应于PutPlaybackConfigurationAdsPersonalizationConcurrency参数EnableVodVastParallelization的成员。

预取 ADS 请求超时

在预取检索期间解析广告决策时 MediaTailor 等待单个 HTTP 响应的最长时间（以毫秒为单位）。默认：个人 ADS 请求超时值。有效范围：250— 10000。

有关预取的更多信息，请参阅预取广告。

此设置对应于PutPlaybackConfigurationAdsPersonalizationTimeouts参数PrefetchAdsRequestTimeoutMilliseconds的成员。

预取最长 ADS 个性化时间

预取检索期间所有 ADS 活动的总时间预算，以毫秒为单位。默认情况下，此超时没有上限。有效范围：250— 10000。

有关预取的更多信息，请参阅预取广告。

此设置对应于PutPlaybackConfigurationAdsPersonalizationTimeouts参数PrefetchMaximumAdsPersonalizationTimeMilliseconds的成员。

> **一起设置超时**
>
> 对于每个工作流程，必须同时设置个人超时和总时间预算。个人超时时间不能超过总预算。直播和 VOD 工作流程共享一个个人 ADS 请求超时时间。将该超时时间与直播和 VOD 总时长预算一起设置。同样，同时设置 Prefetch ADS 请求超时时间和 Prefetch 最长 ADS 个性化时间，不能同时设置其中一个。

查看 MediaTailor 配置详情

除了在创建配置时提供的值之外， AWS Elemental MediaTailor 还显示配置的名称、播放终端节点和相关访问 URL。要查看配置，请使用以下步骤。

查看配置

1. 在处打开 MediaTailor 控制台https://console.aws.amazon.com/mediatailor/。
2. 在 Configurations (配置) 页面上，选择要查看的配置的 Configuration name (配置名称)。

编辑 MediaTailor 配置设置

您可以编辑配置以更新源服务器和广告决策服务器 (ADS) 映射，或更改与内容分发网络 (CDN) 的 AWS Elemental MediaTailor 交互方式。

编辑配置

1. 在处打开 MediaTailor 控制台https://console.aws.amazon.com/mediatailor/。
2. 在 Configurations (配置) 页面上，选择要编辑的配置的名称。
3. 在配置详细信息页面上，选择 Edit (编辑)，然后根据需要修改配置设置。您无法编辑配置名称。有关配置属性的信息，请参阅创建 MediaTailor 播放配置。
4. 选择保存。

删除 MediaTailor 配置

您可以删除配置，使其无法用于播放 AWS Elemental MediaTailor。

删除配置

1. 在处打开 MediaTailor 控制台https://console.aws.amazon.com/mediatailor/。
2. 在 Configurations (配置) 页面上，执行下列操作之一：
   - 选择要删除的配置的名称。
   - 在 Configuration name (配置名称) 列中，选择名称旁的选项，然后选择 Delete (删除)。
3. 在 Delete (删除) 确认框中，输入 **`Delete`**，然后选择 Delete (删除)。

集成内容源以进行 MediaTailor 广告插入

本主题介绍如何将不同类型的视频内容源与集成 MediaTailor。 MediaTailor支持直播和点播内容的 HLS 和 DASH 流媒体协议。该服务可以在指定的广告时段进行广告插入或替换，并且对输入视频清单的结构和格式有特定要求才能启用这些功能。以下主题详细介绍了输入来源要求以及集成 HLS 和 DASH 内容 MediaTailor 以实现个性化广告体验的步骤。

主题

- MediaTailor 广告插入的输入来源要求

- 集成 HLS 源代码
- 整合 MPEG-DASH 来源
- 保护 AWS Elemental MediaTailor 与 SigV4 的原始交互

MediaTailor 广告插入的输入来源要求

输入源必须满足以下要求才能使用 MediaTailor：

- 使用 Apple HLS（HTTP 直播流）或 MPEG DASH（基于 HTTP 的动态自适应流）
- 使用直播或视频点播 (VOD)
- 可通过公共 Internet 访问并具有公有 IP 地址
- 使用标准 HTTP 端口（端口 80）或 HTTPS 端口（端口 443）。 MediaTailor 不支持用于源服务器通信的自定义端口。
- 包含广告插入入门教程中描述的其中一种格式的 MediaTailor 广告标记

集成 HLS 源代码

AWS Elemental MediaTailor 支持等于3或更高的 `.m3u8` HLS 清单，用于直播和视频点播 (VOD)。`EXT-X-VERSION`当 MediaTailor 遇到广告插播时，它会根据内容的类型尝试插入或替换广告。如果没有足够的广告来填补持续时间，则在广告休息的剩余时间内， MediaTailor 将显示基础内容流或配置的列表。有关基于内容类型的 HLS 广告行为的更多信息，请参阅了解 AWS Elemental MediaTailor 广告插入行为。

以下部分提供了有关如何 MediaTailor 处理 HLS 清单的更多信息。

主题

- HLS 支持的广告标记
- 启用广告标记直通
- HLS 清单标签处理
- HLS 清单示例

HLS 支持的广告标记

AWS Elemental MediaTailor 通过解析输入清单中是否存在支持的广告标记，识别 HLS 清单中的广告可用性边界。以下各节介绍 MediaTailor 使用什么样的标记。

EXT-X-ASSET

标签 EXT-X-ASSET 包含广告决策服务器 (ADS) 用于为查看者个性化内容的元数据。EXT-X-ASSET 参数是以逗号分隔的键值对。

要使用该标签，必须满足以下要求：

- 你必须 URL-encode 使用原始清单中的*EXT-X-ASSET*值。以下示例显示了带有键和 URL-encoded 值的EXT-X-ASSET标签。

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

- 您必须在 MediaTailor ADS 配置中包含动态*[asset.]*变量和密钥。以下示例显示了使用动态[asset.]变量和密钥的 MediaTailor ADS 配置。

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

VAST 请求示例

以下示例显示了一个针对 ADS 的 VAST GET 请求。

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

EXT-X-CUE-OUT 和 EXT-X-CUE-IN

这种广告标记类型最为常见。以下示例显示了这些提示标记的选项。

```
#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

借助 EXT-X-DATERANGE 广告标记标签，您可以使用 SCTE35-OUT 属性来指定广告效用的计时。

> **Note**
>
> AWS Elemental MediaTailor 忽略为EXT-X-DATERANGE广告标记提供的任何START-DATE属性。

您可以通过以下方式之一指定广告效用：

- 符合 SCTE35-OUT 和 DURATION 规范的 EXT-X-DATERANGE 标签。

  示例

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

- 成对的 EXT-X-DATERANGE 标签，第一个符合 SCTE35-OUT 规范，第二个符合 SCTE35-IN 规范。

  示例

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

- 先前选项的组合。使用 SCTE35-OUT 和 DURATION 规范指定 EXT-X-DATERANGE 标签，然后使用 SCTE35-IN 规范指定 EXT-X-DATERANGE 标签。在这种情况下， MediaTailor 使用两个规格中最早的提示设置。

示例

```
#EXT-X-DATERANGE:ID="splice-6FFFFFF0",START-DATE="2019-01T00:15:00Z
\",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

您在EXT-X-SPLICEPOINT-SCTE35广告标记标签上附加一个 base64 编码的二进制 SCTE-35文件中的有效负载。解码后的二进制文件必须提供 SCTE-35splice_info_section包含提示出线的标记0x34（用于提供商投放机会的开始）和提示标记（提供商投放机会0x35结束）。

以下示例显示了带有 base64 编码的二进制有效负载的拼接点规范，这些有效负载指定了广告去除和广告插入标记。

```
    #EXT-X-SPLICEPOINT-SCTE35:/DA9AAAAAAAAAP/wBQb+uYbZqwAnAiVDVUVJAAAKqX//
AAEjW4AMEU1EU05CMDAxMTMyMjE5M19ONAAAmXz5JA==
    ...
    #EXT-X-SPLICEPOINT-SCTE35:/DA4AAAAAAAAAP/wBQb+tTeaawAiAiBDVUVJAAAKqH+/
DBFNRFNOQjAwMTEzMjIxOTJfTjUAAIiGK1s=
```

启用广告标记直通

默认情况下，HLS 的 MediaTailor 个性化清单不包含原始清单中的 SCTE-35 广告标记。启用广告标记直通功能后，将以下广告标记从原始清单 MediaTailor传递到个性化清单中：

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

广告标记直通是一项可选设置。如果您想将 SCTE 广告标记包含在 MediaTailor 个性化清单中，请使用广告标记直通。常见用例包括：

- 内容替换-执行内容替换或内容限制。
- 广告跟踪-根据一个或多个广告标记的存在与否触发广告跟踪信息。
- 玩家设置-根据是否存在广告标记，在玩家界面中启用清理或倒数计时器功能。

> **Note**
>
> MediaTailor 不会更改这些标记的值。例如，如果EXT-X-CUE-OUT来源清单60中的值为，但未投放任何广告，则 MediaTailor 不会将该值更改为个性化清单0中的值。

启用广告标记直通

您可以使用 AWS 管理控制台 或 AWS Command Line Interface (AWS CLI) 启用广告标记直通。

使用控制台启用广告标记直通

1. 在处打开 MediaTailor 控制台https://console.aws.amazon.com/mediatailor/。
2. 选择“新建配置”或“编辑配置”。
3. 在“高级设置”部分，从下拉菜单中选择“启用”。

要启用广告标记直通，请使用 AWS Command Line Interface (AWS CLI)

使用 put-playback-Configuration 命令。

HLS 清单标签处理

本节介绍如何 AWS Elemental MediaTailor 管理个性化输出清单中的标签。

EXT-X-CUE 标签

MediaTailor 将输入清单中的EXT-X-CUE-OUTEXT-X-CUE-OUT-CONT、和EXT-X-CUE-IN标签替换为输出清单中的EXT-X-DISCONTINUITY标签。DISCONTINUITY 标记标出以下边界：

- 主要内容过渡到广告的边界
- 一个广告转换到另一个广告的位置
- 广告过渡回主内容的边界

EXT-X-DATERANGE 标签

MediaTailor 将EXT-X-DATERANGE标签从输入清单传递到输出清单。 MediaTailor 还会插入与EXT-X-DISCONTINUITY标签相对应的DATERANGE标签。DISCONTINUITY 标记标出以下边界：

- 主要内容过渡到广告的边界
- 一个广告转换到另一个广告的位置
- 广告过渡回主内容的边界

EXT-X-KEY 标签

MediaTailor 传递输入清单中的EXT-X-KEY标签。这些标记指示主要内容已加密。由于广告未加密，因此 MediaTailor 可在广告效用的开头插入 EXT-X-KEY:METHOD=NONE。当播放返回到主要内容时，通过插入带有定义为加密类型的METHOD值的EXT-X-KEY标签来 MediaTailor 重新启用加密。

无法识别的标签

MediaTailor 将所有未知和自定义标签从输入清单传递到输出清单。

HLS 清单示例

以下部分提供了 HLS 源清单和个性化清单的示例。了解这些示例可以帮助您配置 MediaTailor工作流程并对其进行故障排除。

有关如何将查询参数应用于 HLS 清单和区段的信息，请参阅。MediaTailor HLS 隐式会话初始化

了解 HLS 播放列表类型

HTTP 直播 (HLS) 使用两种主要的播放列表类型：

多变体播放列表

多变体播放列表是列出内容所有可用格式副本的顶级索引文件。它包含对媒体播放列表的引用，但本身不包含任何媒体片段。该播放列表允许玩家根据网络条件、设备功能或用户偏好选择最合适的节目。

这种播放列表类型在各种上下文中也有其他几个名称，包括主播放列表、主清单、主播放列表、主播放列表、索引文件或主 M3U8。

在 MediaTailor 工作流程中，多变体播放列表是播放请求的入口点，也是广告个性化的起点。

媒体播放列表

媒体播放列表包含特定内容再现（质量级别）的实际媒体片段信息。它包括播放单个格式副本所需的时间信息 URLs、片段和其他元数据。

这种播放列表类型也称为媒体播放列表、子清单、区块列表、媒体 M3U8 或格式副本播放列表。

在 MediaTailor 工作流程中，媒体播放列表会进行个性化设置，以按适当顺序包括内容细分和广告细分。

有关 HLS 播放列表类型的更多详细信息，请参阅HLS 播放列表类型。

HLS 起源清单示例

以下示例显示了 HLS 从内容来源 AWS Elemental MediaTailor 收到的 HLS 多变体播放列表。

```
#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_
```

在这个多变体播放列表示例中：

- 这些#EXT-X-STREAM-INF标签定义了具有不同分辨率和比特率的不同格式副本
- 每个格式副本都引用一个媒体播放列表（例如index_1.m3u8）
- 该#EXT-X-MEDIA标签定义了字幕曲目

以下示例显示了 HLS 从内容来源 AWS Elemental MediaTailor 收到的 HLS 媒体播放列表。本示例使用 EXT-X-CUE-OUT 和 EXT-X-CUE-IN 标签来描述广告效用机会。

```
#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:/DAlAAAAAsvhAP/wFAXwAAAGf+/+AdLfiP4AG3dAAAEBAQAAXytxmQ==
    #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=/DAlAAAAAsvhAP/wFAXwAAAGf
+/+AdLfiP4AG3dAAAEBAQAAXytxmQ==
    #EXTINF:6.006,
    index_1_8779961.ts?m=1566416212
    #EXT-X-CUE-OUT-CONT:ElapsedTime=6.640,Duration=21,SCTE35=/DAlAAAAAsvhAP/wFAXwAAAGf
+/+AdLfiP4AG3dAAAEBAQAAXytxmQ==
    #EXTINF:6.006,
    index_1_8779962.ts?m=1566416212
    #EXT-X-CUE-OUT-CONT:ElapsedTime=12.646,Duration=21,SCTE35=/DAlAAAAAsvhAP/wFAXwAAAGf
+/+AdLfiP4AG3dAAAEBAQAAXytxmQ==
    #EXTINF:6.006,
    index_1_8779963.ts?m=1566416212
    #EXT-X-CUE-OUT-CONT:ElapsedTime=18.652,Duration=21,SCTE35=/DAlAAAAAsvhAP/wFAXwAAAGf
+/+AdLfiP4AG3dAAAEBAQAAXytxmQ==
    #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
```

在此媒体播放列表示例中：

- #EXTINF标签指定每个片段的持续时间
- 该#EXT-X-CUE-OUT标签标志着广告插播的开始
- 这些#EXT-X-CUE-OUT-CONT标签提供有关正在进行的广告中断的信息
- 该#EXT-X-CUE-IN标签标志着广告插播的结束

HLS 个性化清单示例

以下示例显示了个性化的 HLS 多变体播放列表。 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
```

请注意如何 MediaTailor 修改媒体播放列表 URLs 以包含特定会话的信息，从而实现个性化广告插入。

以下示例显示了 AWS Elemental MediaTailor 个性化的媒体播放列表。

```
#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/43f3e412052f2808dd84ea1da90e92e914edddee/external-canary-hls/
ee1696a8-4f7f-4c4c-99de-9821131847e8/0/8779960
    #EXTINF:3.0,
    ../../../../segment/43f3e412052f2808dd84ea1da90e92e914edddee/external-canary-hls/
ee1696a8-4f7f-4c4c-99de-9821131847e8/0/8779961
    #EXTINF:3.0,
    ../../../../segment/43f3e412052f2808dd84ea1da90e92e914edddee/external-canary-hls/
ee1696a8-4f7f-4c4c-99de-9821131847e8/0/8779962
    #EXTINF:3.0,
    ../../../../segment/43f3e412052f2808dd84ea1da90e92e914edddee/external-canary-hls/
ee1696a8-4f7f-4c4c-99de-9821131847e8/0/8779963
    #EXTINF:2.966667,
    ../../../../segment/43f3e412052f2808dd84ea1da90e92e914edddee/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
```

在此个性化媒体播放列表示例中：

- MediaTailor 在内容区段之间插入了广告区段
- 这些#EXT-X-DISCONTINUITY标签标记了内容和广告之间的过渡
- 内容分段由源服务器提供（MediaPackage 在本例中）
- 广告区段是从 MediaTailor的广告细分存储空间中提供的

个性化清单的主要区别

MediaTailor 对HLS清单进行个性化设置时，它会进行一些重要的更改：

多变体播放列表更改

- 变体播放列表 URLs 被重写为指向包含会话信息的 MediaTailor-man URLs aged
- 为了获得最佳播放效果，可以重新组织标签的顺序

变体播放列表更改

- 广告标记 (EXT-X-CUE-OUT,EXT-X-CUE-IN) 将替换为实际的广告细分
- 在边界处 content/ad 添加不连续标记 (EXT-X-DISCONTINUITY)
- 内容段 URLs 被重写为指向源或 CDN
- 广告区段 URLs 已添加到 point to MediaTailor 的广告区段存储空间

了解这些更改可以帮助您解决 MediaTailor 工作流程中的问题，并确保正确配置 CDN 和播放器。

相关主题

有关 HLS 清单和的更多信息 MediaTailor，请参阅以下主题：

- HLS 播放列表类型-HLS 播放列表类型的详细说明
- 使用 CDN 优化 MediaTailor 广告个性化和内容交付-有关将 CDN 与以下内容一起使用的信息 MediaTailor
- MediaTailor 广告插入的工作原理- MediaTailor 广告插入工作原理概述

整合 MPEG-DASH 来源

AWS Elemental MediaTailor 支持遵循 DASH 动态配置文件指南的.mpd直播和视频点播 (VOD) 清单。MediaTailor 接受多周期和单周期 DASH-compliant 清单输入，并提供多周期 DASH-compliant清单输出。

输入清单必须具有以下内容：

- SCTE-35 带有splice insert 或 time signal拼接信息设置的事件流。可以使用清除 XML 或 base64 编码的二进制提供设置。
- 带有 segment timelines 的 Segment templates。

对于已发布的清单， MediaTailor 要求源服务器的更新保持以下内容不变：

- 在 start 属性中指定的时段开始时间。
- 时段表示形式的分段模板中的 presentationTimeOffset 值。

最佳做法是，为广告提供与内容流时段相同的AdaptationSet和Representation设置。 AWS Elemental MediaTailor 使用这些设置对广告进行转码以匹配内容流，从而在两者之间顺畅切换。

以下部分提供了有关如何 MediaTailor 处理 DASH 清单中的广告的更多信息。

主题

- DASH 广告标记
- DASH 广告使用时长
- SCTE-35 DASH 中的标记和周期边界
- DASH 清单区段编号
- DASH MPD 示例
- DASH 定位功能

DASH 广告标记

MediaTailor

AWS Elemental MediaTailor 使用 SCTE-35 提示标记使用以下逻辑识别 DASH 清单中的可用广告：

- Multi-period DASH：Event在每个包含SpliceInsert或Period提TimeSignal示标记的广告中 MediaTailor 插入第一个广告。 MediaTailor忽略中的其他Event标记。Period
- Single-period DASH：Event在每个广告中 MediaTailor 插入Period包含SpliceInsert或TimeSignal提示标记的广告。

默认情况下，将 DASH 清单作为多周期清单进行 AWS Elemental MediaTailor 管理。您可以更改配置以处理源服务器中的单周期 DASH 清单。有关信息，请参阅the section called “创建配置”。

以下部分提供了有关 DASH 广告标记处理的更多详细信息，并提供了来自原始内容的装饰清单。

DASH 原始清单 XML 要求

必须正确格式化来源的 DASH 清单中的广告标记， MediaTailor 才能识别广告插播时间。以下主题以明确 XML 描述了这些格式要求。

SpliceInsert使用清晰的 XML

SpliceInsert透明 XML 中的广告标记必须包含以下内容：

- EventStream必须具有以下属性：schemeIdUri=urn:scte:scte35:2013:xml
- Event必须坚持 scte35:SpliceInfoSection
- scte35:SpliceInfoSection必须坚持 scte35:SpliceInsert
- scte35:SpliceInsert必须具有以下属性：outOfNetworkIndicator="true"

Example **SpliceInsert**在 XML 中

在以下示例中，所需的 SCTE 标记以粗体显示。

```
<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使用清晰的 XML

TimeSignal透明 XML 中的广告标记必须包含以下内容：

- EventStream必须具有以下属性：schemeIdUri=urn:scte:scte35:2013:xml
- Event必须坚持 scte35:SpliceInfoSection
- scte35:SpliceInfoSection必须坚持 scte35:TimeSignal
- scte35:SpliceInfoSection还必须保持 scte35:SegmentationDescriptor
- scte35:SegmentationDescriptor必须具有以下属性，其中值是有效的Cue-out 数字：segmentationTypeId="*xx*"

- scte35:SegmentationDescriptor必须坚持 scte35:SegmentationUpid

Cue-out 数字

以下是支持的提示数字。segmentationTypeId

分段消息	segmentationTypeId 值	十六进制值
分销商广告结束	51	0x51
分销商广告开始	50	0x32
分销商安置机会已结束	55	0x37
分销商安置机会开始	54	0x36
结束休息	35	0x23
提供商广告结束	49	0x31
提供商广告开始	48	0x30
提供商叠加投放机会已结束	57	0x39

分段消息	segmentationTypeId 值	十六进制值
提供商叠加投放机会开始	56	0x38
提供商安置机会已结束	53	0x35
提供商安置机会开始	52	0x34
开始休息	34	0x22

Example **`TimeSignal`**在 XML 中

在以下示例中，所需的 SCTE 标记以粗体显示。

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

DASH 起源清单 base64 编码的二进制要求

必须正确格式化来源的 DASH 清单中的广告标记， MediaTailor 才能识别广告插播时间。以下主题以 base64 编码的二进制文件描述了这些格式要求。

两者TimeSignal和 base64 编码清单中的SpliceInsert广告标记都必须包含以下内容：

- EventStream必须具有以下属性：urn:scte:scte35:2014:xml+bin
- Event必须坚持 scte35:Signal
- scte35:Signal必须scte35:Binary包含包含 base64 编码的二进制文件。

解码后的二进制文件必须提供splice_info_section与清晰 XML 广告标记所需信息相同的信息。

- 命令类型必须为splice_insert()或 time_signal()
- 其他设置必须符合TimeSignal使用清晰的 XML和中所述的设置SpliceInsert使用清晰的 XML。

解码后的二进制必须提供 splice_info_section，且其信息集必须与清除 XML 在 scte35:SpliceInfoSection 元素中提供的相同。命令类型必须为 splice_insert() 或 time_signal()，且其他设置必须符合先前针对清除 XML 交付的描述。

以下示例显示了此选项，所需标记以粗体显示。

```
<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>/DAhAAAAAAAAAP/wEAUAAAHAf+9/fgAg9YDAAAAAAAA25aoh</
scte35:Binary>
        </scte35:Signal>
      </Event>
      <Event presentationTime="1541436360" duration="24" id="30">
        <scte35:Signal xmlns="http://www.scte.org/schemas/35/2016">

 <scte35:Binary>QW5vdGhlciB0ZXN0IHN0cmluZyBmb3IgZW5jb2RpbmcgdG8gQmFzZTY0IGVuY29kZWQgYmluYXJ5Lg=
scte35:Binary>
        </scte35:Signal>
      </Event>
  .
  .
  .
</Period>
```

以下是先前示例中列出的第一个事件的解码二进制。splice_command_type 的设置为 5，这表示 splice_insert。

```
{
        "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 广告使用时长

在播放过程中，当 AWS Elemental MediaTailor 遇到广告可用性时，它会将部分或全部广告替换为广告。 MediaTailor 在广告开始时开始替换广告，包括如下广告：

- 如果广告效果指定了持续时间， MediaTailor 则在持续时间边界内添加尽可能多的广告，而不会覆盖随后的内容。
- 如果未提供持续时间 MediaTailor ，则在广告有效期结束之前添加广告。对于多周期清单来说，这就是周期的结束。对于单周期清单，活动到此结束。 MediaTailor 在广告结束后不会播放广告，并且在广告结束时会截断当前广告，而不是覆盖随后的内容。

操作方法 AWS Elemental MediaTailor 寻找广告的有效时长

AWS Elemental MediaTailor 按以下顺序搜索持续时间设置：

1. `Event duration`
2. 对于拼接插入，搜索 `scte35:BreakDuration duration`
3. 对于时间信号，搜索 `scte35:SegmentationDescriptor segmentationDuration`

如果找 AWS Elemental MediaTailor 不到任何这些设置，它会无期限地管理广告收录。

以下示例显示了具有 `duration` 的 `Event`。

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

```
              </scte35:SpliceInsert>
            </scte35:SpliceInfoSection>
          </Event>
            ...
```

以下示例显示未指定持续时间的广告效用。Event 没有 duration，scte35:SpliceInsert 元素不包含 scte35:BreakDuration 子元素。

```
  <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>
            ...
```

SCTE-35 DASH 中的标记和周期边界

MediaTailor 支持提供单周期或多周期清单的 DASH 来源。对于单周期清单，使用来自的时间信息将清单 MediaTailor 转换为多周期格式EventStream以定义周期边界。转换后，单周期和多周期清单都要经过相同的广告个性化工作流程。

当在一段时间内 MediaTailor 检测到线性广告插播事件时，它会将整个时段视为广告插播时间段，并相应地对其进行个性化设置。 MediaTailor 使用时段开始时间 (Period@start) 作为广告时段开始时间，无论事件本身内有何时间偏移信号。

关键行为

MediaTailor 在 DASH 周期内处理 SCTE-35 事件时应用以下规则：

- 对于线性广告时段，如果事件的时间表明广告时段在时段边界之后开始，则 MediaTailor 忽略信号的开始时间并开始个性化设置。Period@start对于非线性（叠加）事件， MediaTailor 尊重信号presentationTime。
- 如果新的 SCTE 事件出现在现有线性广告插播时段的中间，则将其 MediaTailor 忽略。

- 如果活跃的非线性（叠加）事件已存在于原始周期，则 MediaTailor 忽略该周期内的任何新线性事件。

中期事件信号

使用 AWS Elemental MediaPackage v2 时，即使未在 SCTE-35 标记边界创建新周期，也会 MediaTailor 接收标记。这可能会导致 SCTE 标记出现在内容周期的中间。

当一个新 SCTE-35 事件在内容周期中间发出线性广告插播信号时，会将其 MediaTailor 识别为广告中断机会，但在发出信号的事件时`Period@start`而不是在发出信号的事件时开始个性化设置。这种不匹配可能会导致以下问题：

- 永远不会向观众展示的广告
- 插入石板是为了填补时差造成的剩余时段

从时段开始计算事件偏移量

使用以下公式来确定事件与其周期起点相抵消的距离：

```
Offset = (presentationTime - presentationTimeOffset) / timescale
```

以下示例显示`EventStream`了自周期`presentationTime`起点偏移的 a：

```
<Period id="1741948755359" start="PT10522H39M15.359S">
  <EventStream schemeIdUri="urn:scte:scte35:2013:xml" timescale="90000"
               presentationTimeOffset="156775387982400">
    <Event presentationTime="156775484008800" duration="36900000">
      <scte35:SpliceInfoSection protocolVersion="0" ptsAdjustment="0" tier="4095">
        <scte35:TimeSignal>
          <scte35:SpliceTime ptsTime="587770208"/>
        </scte35:TimeSignal>
        <scte35:SegmentationDescriptor segmentationEventId="168602131"
            segmentationEventCancelIndicator="false"
            segmentationDuration="36900000"
            segmentationTypeId="34"
            segmentNum="0" segmentsExpected="1">
        </scte35:SegmentationDescriptor>
      </scte35:SpliceInfoSection>
    </Event>
  </EventStream>
</Period>
```

在此示例中，偏移量计算为：

```
156775484008800 - 156775387982400 = 96026400 ticks
```

```
96026400 / 90000 = 1066.96 seconds (approximately 17 minutes 46 seconds)
```

公布的休息时间为：

```
36900000 / 90000 = 410 seconds (approximately 6 minutes 50 seconds)
```

由于活动不是在时段边界处 MediaTailor 开始，因此开始个性化设置`Period@start`，这比预定休息时间提前 17 分钟以上。这会导致广告与实际的分段时段不一致。如果配置了 slate，则 MediaTailor 插入清单周期以填补空白。

早期提醒

当广告中断时间在最初的信号持续时间之前结束时，就会出现早期提示。在多周期清单中，这会创建一个新的时段，提前结束最初的广告时段。

不同 AWS Elemental MediaPackage 版本的行为有所不同：

- MediaPackage v1 — 创建新时段并删除其中的所有事件信息。
- MediaPackage v2 — 在中包含两个广告标记`EventStream`：原始提示和提示输入。要发出提示信号，请将设置为`outOfNetworkIndicator`。`false`

MediaTailor 通过检查值来识别早期提示点。`outOfNetworkIndicator`当在同一时段内 MediaTailor 找到与提示出线相匹配的提示时，它不会将该提示视为新的广告插播机会。

以下示例显示了既`EventStream`有提示出线又有早期提示的：

```
<EventStream schemeIdUri="urn:scte:scte35:2013:xml" timescale="90000"
             presentationTimeOffset="276012000">
  <Event presentationTime="273308400" duration="5400000">
    <scte35:SpliceInfoSection duration="5400000" protocolVersion="0"
                              ptsAdjustment="183600" tier="4095">
      <scte35:SpliceInsert spliceEventId="222" spliceEventCancelIndicator="false"
          outOfNetworkIndicator="true" spliceImmediateFlag="false"
          uniqueProgramId="1" availNum="1" availsExpected="1">
        <scte35:Program>
          <scte35:SpliceTime ptsTime="273124800"/>
        </scte35:Program>
```

```
        <scte35:BreakDuration autoReturn="true" duration="5400000"/>
      </scte35:SpliceInsert>
    </scte35:SpliceInfoSection>
  </Event>
  <Event presentationTime="276012000">
    <scte35:SpliceInfoSection protocolVersion="0" ptsAdjustment="183600" tier="4095">
      <scte35:SpliceInsert spliceEventId="1" spliceEventCancelIndicator="false"
          outOfNetworkIndicator="false" spliceImmediateFlag="false"
          uniqueProgramId="1" availNum="1" availsExpected="1">
        <scte35:Program>
          <scte35:SpliceTime ptsTime="275828400"/>
        </scte35:Program>
      </scte35:SpliceInsert>
    </scte35:SpliceInfoSection>
  </Event>
</EventStream>
```

第一个事件 (outOfNetworkIndicator="true") 表示广告插播时间开始。第二个事件 (outOfNetworkIndicator="false") 表示休息结束的提前提示。

缓解期中期 SCTE-35 标记问题

如果您因 SCTE-35 标记在 DASH 直播中段出现而出现意想不到的广告个性化行为，则 MediaTailor 可能会抑制或传递不适用于您的广告个性化工作流程的特定标记。

启用标记抑制后， MediaTailor 执行以下操作：

- 抑制标记 — MediaTailor 忽略指定的标记，并从生成的清单中移除相应的事件。如果没有其他事件， MediaTailor 也将空白部分EventStream从时间段中移除。
- 直通标记 — 出于个性化目的 MediaTailor忽略指定的标记，但标记仍保留在生成的清单中。

支持以下中断类型进行抑制或直通：

- BREAK
- PROVIDER_ADVERTISEMENT
- DISTRIBUTOR_ADVERTISEMENT
- PROVIDER_PLACEMENT_OPPORTUNITY
- DISTRIBUTOR_PLACEMENT_OPPORTUNITY
- PROVIDER_OVERLAY_PLACEMENT_OPPORTUNITY

MediaTailor 记录每个被抑制或传递的标记，并在清单响应中抑制一个或多个标记时发出计数指标。

> **Important**
>
> 此配置仅适用于直播模式下的 DASH 清单。要为您的配置配置标记抑制或直通，请联系 AWS 支持部门。联系支持部门时，请指定要隐藏或传递哪些分段类型。

DASH 清单区段编号

MediaTailor 支持使用<SegmentTimeline>和media属性定义的媒体片段。<SegmentTemplate>您可以使用 $Number$ 标识符或 $Time$ 标识符在 media 属性中指定媒体段列表。

以下示例显示了具有 media 属性设置的 SegmentTemplate，该设置使用 $Number$ 标识符。

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

以下示例显示了具有 media 属性设置的 SegmentTemplate，该设置使用 $Time$ 标识符。

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

DASH MPD 示例

以下各节提供了 DASH 起源 MPDs 和个性化的示例 MPDs。了解这些示例可以帮助您配置 MediaTailor工作流程并对其进行故障排除。

有关如何将查询参数应用于 DASH 清单和区段的信息，请参阅MediaTailor DASH 隐式会话初始。

了解 DASH MPD 结构

基于 HTTP 的动态自适应流式传输 (DASH) 使用媒体演示描述 (MPD) 清单来提供流媒体内容。MPD 是一个 XML 文档，描述了媒体内容的结构和可用性。

MPD（媒体演示说明）

MPD 是 DASH 流媒体中的主要清单文件，用于描述媒体内容的结构和可用性。它包含有关构成流媒体内容的时段、改编集、表示形式和片段的信息。

这种清单类型在各种上下文中也有其他几个名称，包括 DASH 清单、DASH MPD、主清单（与 HLS 比较时）或演示文稿清单。

在 MediaTailor 工作流程中，MPD 是播放请求的入口点，也是广告个性化的起点。

周期

周期是 DASH 演示文稿中的一个时间部分。每个 Period 都包含一个或多个改编集，代表媒体时间跨度。在广告插入工作流程中，通常使用单独的时间段来划分内容和广告。

在 MediaTailor 工作流程中，使用时间段将主要内容与广告内容分开，每个广告通常由自己的时间段表示。

AdaptationSet

对一个或多个媒体内容组件的一 AdaptationSet 组可互换编码版本进行分组。例如，一个 AdaptationSet 可能包含多个视频质量级别，而另一个可能包含多个音频语言选项。

在 MediaTailor 工作流程中， AdaptationSets 在广告插入期间保留，以保持内容和广告之间媒体类型的一致性。

表示

表示形式是其中媒体内容的特定编码版本。 AdaptationSet每种表示形式通常在比特率、分辨率或其他编码参数上有所不同，允许客户端根据网络条件和设备功能选择最合适的版本。

在 MediaTailor 工作流程中，广告周期中的表现尽可能与内容时段中的表现形式紧密匹配，以确保流畅的观看体验。

有关 DASH 清单类型的更多详细信息，请参阅DASH 清单类型。

有关 DASH 清单结构和配置的信息 AWS Elemental MediaPackage，请参阅 DASH 概述中的 MediaPackage 用户指南部分。

直播 DASH MPD 示例

本节提供实时 DASH 的示例 MPDs。每个示例都列出了从源站服务器收到的 MPD，然后使用广告对 MPD MediaTailor 进行了个性化设置。

DASH MPD 拼接插入示例

接头插入的 DASH 原点 MPD 示例

以下来自 MPD 的示例显示了 DASH 从内容来源收到的清单中的广告可用性。此示例使用`SpliceInsert`标记来表示广告可用。

```
<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:SegmentationTypeID segmentationType="52"/>
          <scte35:SegmentNum segmentNum="1"/>
          <scte35:SegmentsExpected segmentsExpected="1"/>
        </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>
```

在这个 Origin MPD 示例中：

- 该<EventStream>元素包含 SCTE-35 标记，用于指示广告可用
- 该<scte35:SpliceInsert>元素提供有关广告可用性的详细信息
- <scte35:BreakDuration>元素指定广告时段的持续时间
- 这些<AdaptationSet>元素定义了可用的视频和音频流

用于拼接插件的 DASH 个性化 MPD 示例

AWS Elemental MediaTailor 根据广告规格对广告进行个性化设置。个性化设置反映了从播放器和当前正在进行的广告活动接收的查看者数据。

以下示例显示了对广告进行 AWS Elemental MediaTailor 个性化设置后的可用性。

```
<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>
```

在这个个性化 MPD 示例中：

- MediaTailor 为广告内容创建了新的时段
- 该<BaseURL>元素指向广告内容的位置
- 这些<AdaptationSet>元素保持与内容相似的结构
- 这些<Representation>元素为广告内容提供了不同的质量等级

DASH MPD 时间信号示例

DASH 原点 MPD 时间信号示例

以下来自 MPD 的示例显示了 DASH 从内容来源收到的清单中的广告可用性。此示例使用TimeSignal标记来表示广告可用。

```
<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:SegmentationTypeID segmentationType="52"/>
          <scte35:SegmentNum segmentNum="1"/>
          <scte35:SegmentsExpected segmentsExpected="1"/>
        </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>
```

在这个 Origin MPD 示例中：

- 使用该<scte35:TimeSignal>元素代替 <scte35:SpliceInsert>
- <scte35:SegmentationDescriptor>提供了有关广告可用性的更多信息

DASH 个性化 MPD 时间信号示例

AWS Elemental MediaTailor 根据广告规格对广告进行个性化设置。个性化设置反映了从播放器和当前正在进行的广告活动接收的查看者数据。

以下示例显示了对广告进行 AWS Elemental MediaTailor 个性化设置后的可用性。

```
<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>
```

时间信号的个性化 MPD 与拼接插入的 MPD 类似，它为广告内容 MediaTailor 创建了一个新的周期。

VOD DASH MPD 示例

本节提供视频点播 (VOD) DASH MPDs 的示例。每个示例都列出了从源站服务器收到的 MPD，然后使用广告对 MPD MediaTailor 进行了个性化设置。

VOD DASH MPDs 遵循与直播相同的结构 MPDs，但它们通常在 MPD 元素中有一个`type="static"`属性，并且可能包含不同内容片段的多个时段。

有关 VOD DASH 的示例 MPDs，请参阅中的 MediaTailor 文档。DASH 广告标记

个性化的主要区别 MPDs

对 DASH MediaTailor 进行个性化设置时 MPDs，它会进行一些重要的更改：

经期处理

- 为广告内容创建了新的时段
- 调整时段开始时间以保持时间表的连续性
- EventStream 带有 SCTE-35 标记的元素会被处理和移除

AdaptationSet 和陈述处理

- AdaptationSets 在广告中，创建时段是为了匹配内容而创建的 AdaptationSets
- 为广告内容的不同质量水平创建了陈述
- SegmentTemplate 元素已更新为指向广告内容

了解这些更改可以帮助您解决 MediaTailor 工作流程中的问题，并确保正确配置 CDN 和播放器。

相关主题

有关 DASH MPDs 和的更多信息 MediaTailor，请参阅以下主题：

- DASH 清单类型-详细说明 DASH 清单类型
- 使用 CDN 优化 MediaTailor 广告个性化和内容交付-有关将 CDN 与以下内容一起使用的信息 MediaTailor
- MediaTailor 广告插入的工作原理- MediaTailor 广告插入工作原理概述
- 有关 DASH 清单结构和 MediaPackage 配置的全面信息，请参阅 DASH 概述中的 MediaPackage 用户指南部分。

DASH 定位功能

本节提供了有关 DASH 的位置功能（在 AWS Elemental MediaTailor中默认启用）的信息。如果您为访问 MediaTailor 清单创建内容分发网络 (CDN) 路由规则，请阅读本节。如果您对不支持粘性 HTTP 重定向的播放器使用服务器端报告，也请参阅本节。

什么是位置功能？

位置功能允许不支持粘性 HTTP 重定向的播放器在其清单更新请求中提供粘性行为。

AWS Elemental MediaTailor 使用无会话初始化，它需要玩家采取粘性的 HTTP 重定向行为。通过服务器端报告，当玩家请求更新清单时 MediaTailor，该服务会发出 302 临时重定向，将玩家定向到个性化清单的终端节点。 MediaTailor 在响应中包含会话 ID，作为查询参数。其目的是使播放器遵循整个会话的 URL，但不支持粘性 HTTP 重定向的播放器会删除重定向并返回原始 URL。当玩家返回到原始网址时，每个新请求都会 MediaTailor 创建一个新会话，而不是继续使用原始会话。这可能会导致清单损坏。

DASH 规范在定位功能中为这个问题提供了解决方案，该功能在 AWS Elemental MediaTailor 配置中默认处于启用状态。启用此功能后， MediaTailor 将绝对 URL 放入清单<Location>标签中。不支持粘性 HTTP 重定向的播放器可以使用 <Location> 中提供的 URL 来请求对清单的更新。

我是否需要在我的配置中禁用位置功能？

位置功能将覆盖您设置的用于访问 AWS Elemental MediaTailor 清单的所有 CDN 路由规则，因此，您可能需要禁用该功能。位置功能不会影响内容或广告分段的 CDN 缓存。

在以下列表中查找您的情况，以确定您是否需要为您的配置禁用位置功能以及如何处理该功能：

- 如果您尚未设置用于访问 AWS Elemental MediaTailor 清单的 CDN 路由规则，请将位置设置保持启用状态。
- 否则，请使用以下规则：
  - 如果您未使用服务器端报告或您所有的播放器都支持粘性 HTTP 重定向，请禁用位置功能。有关如何在控制台上执行此操作的信息，请参阅the section called “创建配置”。
  - 否则，请联系 AWS 支持。

我是否需要使用位置功能？

对于不支持粘性 HTTP 重定向的玩家，你需要使用定位功能。对您所有的清单更新请求使用 <Location> 标签中提供的 URL。

示例

示例 URL 和示例 <Location> 标签。

- Example示例：初始请求 URL

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

- Example示例：重定向的 302 响应

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

- Example示例：清单中的位置标签

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

保护 AWS Elemental MediaTailor 与 SigV4 的原始交互

签名版本 4 (SigV4) 是一种签名协议，用于对通过 HTTPS 向支持源发出的 MediaTailor 请求进行身份验证。 MediaTailor 仅支持 HTTPS 通信，不允许 HTTP 连接。使用 SigV4 签名时，在向 Chann MediaTailor el Assembly、Amazon S3 和 AWS Elemental MediaPackage 版本 2 发起的 HTTPS 原始请求中 MediaTailor 包含已签名的授权标头。

您可以在源头使用 SigV4 来确保清单请求仅在清单请求来自 MediaTailor 并包含签名授权标头时才能得到满足。这样，未经授权的 MediaTailor播放配置就无法访问您的原始内容。如果签名的授权标头有效，则您的源会满足请求。如果无效，则请求失败。

以下各节描述了对支持的源使用 MediaTailor SigV4 签名的要求。

MediaTailor 渠道组装要求

如果您使用 SigV4 来保护您的 MediaTailor 频道集合来源，则必须满足以下要求 MediaTailor 才能访问清单：

- 您的 MediaTailor 配置中的原始基本 URL 必须是采用以下格式的频道集合频道：`channel-assembly.mediatailor.region.amazonaws.com`

- 您的源必须配置为使用 HTTPS。 MediaTailor 仅支持 HTTPS 通信，不允许 HTTP 连接。如果未在源站启用 HTTPS，则 MediaTailor 不会对请求进行签名。
- 您的频道必须有包括以下内容的原始访问政策：
  - 主要访问权限 MediaTailor 以访问您的频道。授予访问 mediata ilor.amazonaws.com 的访问权限。
  - IAM 权限 mediatailor：读GetManifest 取配置引用的所有多变量播放列表。 MediaTailor

  有关在频道上设置政策的信息，请参阅使用 MediaTailor 控制台创建频道。

Example Channel Assembly 的原始访问政策，范围限于 MediaTailor 配置账户

```
{
    "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 Channel Assembly 的原始访问策略，范围限于 MediaTailor 播放配置

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

亚马逊 S3 要求

如果您使用 SigV4 保护您的 Amazon S3 来源，则必须满足以下要求 MediaTailor 才能访问清单：

- 您的 MediaTailor 配置中的原始基本 URL 必须是采用以下格式的 S3 存储桶：`s3.region.amazonaws.com`

- 您的源必须配置为使用 HTTPS。 MediaTailor 仅支持 HTTPS 通信，不允许 HTTP 连接。如果未在源站启用 HTTPS，则 MediaTailor 不会对请求进行签名。
- 您的频道必须有包括以下内容的原始访问政策：
  - 访问您的存储桶 MediaTailor 的委托人访问权限。授予访问 mediatailor.amazonaws.com 的访问权限。

    有关在 IAM 中配置访问权限的信息，请参阅 AWS 身份和访问管理用户指南中的访问管理。
  - IAM 权限 s3：GetObject 读取 MediaTailor 配置引用的所有顶级清单。

有关适用于 Amazon S3 的 SigV4 的一般信息，请参阅 Amazon S3 API 参考中的身份验证请求（AWS 签名版本 4）主题。

Example Amazon S3 的原始访问策略，范围限于账户 MediaTailor

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

Example Amazon S3 的原始访问策略，范围限于 MediaTailor 播放配置

```
{
    "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 要求

如果您使用 SigV4 保护您的 MediaPackage v2 来源，则必须满足以下要求 MediaTailor 才能访问清单：

- 您的 MediaTailor 配置中的原始基本 URL 必须是采用以下格式的 MediaPackage v2 端点：`mediapackagev2.region.amazonaws.com`
- 您的源必须配置为使用 HTTPS。 MediaTailor 仅支持 HTTPS 通信，不允许 HTTP 连接。如果未在源站启用 HTTPS，则 MediaTailor 不会对请求进行签名。
- 您的频道必须有包括以下内容的原始访问政策：
  - 访问您的终端节点的主 MediaTailor 访问权限。授予访问 mediatailor.amazonaws.com 的访问权限。
  - IAM 权限 mediapackagev2：读取配置GetObject 引用的所有多变量播放列表。 MediaTailor

有关 v2 版 SigV4 的一般信息，请参阅 MediaPackage v2 API 参考中的身份验证请求（AWS 签名版本 4）主题。 MediaPackage

Example MediaPackage v2 的原始访问政策，范围限于账户 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 MediaPackage v2 的原始访问策略，范围限于 MediaTailor 播放配置

```
{
    "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”"}
    }
}
```

集成 AWS Elemental MediaTailor 使用谷歌的广告平台

MediaTailor 自动使用谷歌广告管理器 (GAM)、谷歌活动管理器 (GCM) 和谷歌展示和视频 360 (DV360) 对服务器间连接进行身份验证。当 MediaTailor 检测到广告请求或跟踪事件发往 Google 广告服务器时，它会自动建立所需的安全、经过身份验证的连接。

您必须在相关的 Google 广告平台上注册一个帐户。 MediaTailor 支持以下集成类型：

- Server-side 集成： MediaTailor 自动保护向 Google 广告平台发出的服务器端广告请求和展示跟踪请求。这包括：
  - 谷歌广告管理器 (GAM)：向谷歌广告服务器发出的广告请求会自动受到保护，这是访问授权买家（谷歌的实时广告销售市场和广告交易平台）所必需的。
  - Google Campaign Manager (GCM)： Ad-serving 跟踪请求会自动通过 Google 经过身份验证的终端节点进行路由，并保证安全，为在这些平台上投放广告活动的广告商提供更准确的报告和更少的拒绝曝光量。
  - Display & Video 360 (DV360)：广告请求会自动通过 Google 经过身份验证的终端节点进行路由，并提供安全保护，为在这些平台上投放广告活动的广告商提供更准确的报告和更少的拒绝曝光量。
- Client-side 集成：使用谷歌编程访问库 (PAL) SDK 进行玩家集成。如果您想在 Google 广告管理器中使用公开竞价交易类型，则需要进行此集成。

所有其他广告请求继续运行，未经修改。

Ad Manager 对编程交易类型的支持因您使用的集成类型而异。有关可用选项的列表，请参阅交易类型或联系您的 Google 帐号团队。

主题

- Server-side AWS Elemental MediaTailor 与谷歌的广告平台整合
- Client-side AWS Elemental MediaTailor 与谷歌广告管理器集成
- 使用 Google 广告管理器优化广告填充率

Server-side AWS Elemental MediaTailor 与谷歌的广告平台整合

MediaTailor 使用谷歌颁发的 SSL 证书，自动对与谷歌广告管理器 (GAM)、谷歌广告管理器 (GCM) 和谷歌展示和视频 360 (DV360) 的服务器间连接进行身份验证。 AWS Elemental MediaTailor当 MediaTailor 检测到发往 Google 广告服务器的广告请求或跟踪事件时，它会自动建立所需的安全、经过身份验证的连接，无需额外设置 SSL 证书或支持票证。

前提条件：您必须拥有相关谷歌广告平台（谷歌广告管理器、谷歌广告管理器或展示与视频360）的活跃账号。

MediaTailor 根据以下目标 URI 规则应用 Google-issued SSL 证书身份验证：

谷歌广告管理器 (GAM)：服务器端广告请求

- 此规则会触发对的广告请求`serverside.doubleclick.net`，包括当您使用该网域配置 ADS 模板网址时，或者当重定向广告请求解析到该网域时。
- MediaTailor 如果`ssss=mediatailor`查询参数不存在，则添加该参数，然后使用 Google-issued SSL 证书保护连接。
- 这提供了访问授权买家的权限，即谷歌的实时广告销售市场和广告交易平台。

谷歌广告管理器 (GAM)：在 VMAP 中重定向的广告请求

- 如果 MediaTailor 之前在当前请求`serverside.doubleclick.net`中看到过 VAST `pubads.g.doubleclick.net` 重定向且之前的 ADS 响应是 VMAP 响应，则此规则将在重定向的广告请求上触发。
- MediaTailor 自动将域更改为，`serverside.doubleclick.net`并使用 Google-issued SSL 证书建立经过身份验证的连接。
- 这允许授权买家访问 VMAP-based 广告工作流程。

谷歌广告管理器 (GCM)：广告投放和跟踪请求

- 当 URI 包含`/ddm/pfadx/`或`/ddm/trackimp/`（已编码或未编码）时，此规则会触发`ad.doubleclick.net`广告投放和跟踪请求。
- MediaTailor 自动将域更改为，`serverside.doubleclick.net`并使用 Google-issued SSL 证书建立经过身份验证的连接。
- 这为广告商在这些平台上投放广告活动提供了更准确的报告和更少的拒绝曝光量。

Display & Video 360 (DV360)：广告请求

- 当 URI 包含`/dbm/vast/`（已编码`bid.g.doubleclick.net`或未编码）时，此规则会触发`pagead2.googlesyndication.com`广告请求。

- MediaTailor 自动将域更改为，`serverside.doubleclick.net`并使用 Google-issued SSL 证书建立经过身份验证的连接。
- 这样可以为展示和视频 360 广告系列提供经过身份验证的广告。

所有其他广告请求都是在没有 Google-issued SSL 证书身份验证的情况下处理的。

> **Note**
>
> SSL 证书身份验证现在是自动的。如果您之前提交了支持票证以启用 SSL 证书，则无需执行任何操作——您的配置将继续有效。

Client-side AWS Elemental MediaTailor 与谷歌广告管理器集成

要使用 Google Ad Manager 编程访问库 (PAL) SDK 和使用 Google Ad Manager 的公开竞价交易类型，需要进行 MediaTailor 客户端集成。

PAL SDK 提供有关播放会话的内容、设备和用户数据的信息。通过 PAL SDK，您可以将这些信息提供给 Google 广告管理器，然后谷歌广告管理器可以更好地确定要展示哪些定向广告。SDK 适用于安卓、iOS、HTML5 和 Cast。有关使用 PAL SDK 的信息，请参阅 Google Ad Manager PAL SDK 。

创建与 Ad Manager 的客户端集成

1. 使用 PAL SDK 生成随机数。有关设置随机数的信息，请参阅 Google Ad Manager PAL SDK 上的 SDK 文档。

   > **Note**
   >
   > nonce 是 PAL 为直播请求生成的加密字符串。每个请求都必须有一个唯一的随机数。

2. 使用您的 ADS 请求中的`givn`参数来传递随机数值。为此，请更新您的广告网址以包括`&givn=`*`[player_params.givn]`*。有关说明，请参阅启用客户端跟踪。

Datazoom 播放器 SDK

MediaTailor 已与Datazoom合作提供免费的玩家软件开发工具包，以简化与SDK的集成，例如谷歌广告管理器PAL中提供的软件开发工具包。有关 Datazoom 和 MediaTailor 合作伙伴关系的信息，请参

阅。Datazoom 免费播放器 SDK要访问Datazoom播放器软件开发工具包，请使用D atazoom网站上的联系信息。 AWS

使用 Google 广告管理器优化广告填充率

当您使用 Google 广告管理器 (GAM) 作为广告服务器时 MediaTailor，您的广告模板网址参数会直接影响广告填充率和收入。参数值不正确是填充率低的最常见原因之一。

推荐的 GAM 参数

在您的广告模板网址中加入以下参数，以确保 GAM 持续返回广告并填充广告间隔时间。

`correlator=[avail.random]`

确保 GAM 将每个广告插播时间视为一个独特的请求并返回新的广告。如果没有此参数，GAM 可能会在同一会话中重复广告插播时返回空响应。

`pmxd=[session.avail_duration_ms]`

广告窗格的最大持续时间（以毫秒为单位）。使用 MediaTailor 动态变量，`[session.avail_duration_ms]`使该值自动与 SCTE 广告标记的实际持续时间相匹配。请勿对该值进行硬编码——如果硬编码的时长短于实际的广告插播时间，则剩余时间将空白。

`pmad=-1`

每个 pod 的最大广告数量。将此设置`-1`为允许尽可能多的广告填补空白，或将其设置为所需的上限（例如`10`）。较低的值（例如 4）会人为地限制可以填补空白处的广告数量，即使有更多库存也是如此。

`vpos`

视频位置（片头、中片或片后）。将其设置为与您的实际广告位类型相匹配的值。请勿使用`[avail.type]` — MediaTailor 不支持此变量，它将解析为空值，这可能导致 GAM 拒绝请求或退回低价值库存。

GAM 的玩家参数

某些 GAM 参数需要发布者或玩家应用程序在会话初始化时提供的值。将这些 MediaTailor 作为玩家参数传递给，并在您的广告模板网址中引用它们。有关配置玩家参数的详细信息，请参阅MediaTailor ADS 请求的玩家变量。

`ppid=[player_params.ppid]`

编程 Publisher-Provided 标识符。提供这个由发布商分配的值，这样 GAM 就可以跟踪会话并在每次广告插播时继续返回广告。否则，GAM 可能会在会话的第一次中断后停止返回广告。

`rdid=[player_params.rdid]`（可重置的设备标识符）

设备广告 ID。对于 CTV 和智能电视设备，程序化广告购买者需要此标识符才能对库存进行出价。没有它，CTV 设备的填充率可能会大大降低。

`msid=[player_params.msid]`（应用程序 ID）

应用程序标识符。例如`rdid`，这是程序化买家定位 CTV 和智能电视设备所必需的。

有关 GAM 支持的设备定位参数的更多信息，请参阅 Google 网站上的 Google 广告管理器定位参数。

常见的配置错误

以下配置错误是 GAM 填充率低的常见原因：

- 硬编码 **pmxd** -使用固定值（例如，当实际广告间隔时间较长（例如 90 秒）时，剩余时间未填充。`pmxd=60000`请改用 `[session.avail_duration_ms]`。
- 低**pmad**值 —将每个 pod 的最大广告数设置得太低会阻止 GAM 即使有库存也无法填满整个休息时长。无限`-1`使用，或设置所需的上限。
- 缺失 **correlator** —如果没有此参数，GAM 可能会在同一会话中重复广告插播时返回空的 VAST 响应。
- 缺失 **ppid** —如果没有此玩家参数，GAM 可能会在会话中的第一个广告插播时间后停止返回广告。
- CTV 设备上缺少设备 ID — 如果没有`rdid`，程序化购买者就无法锁定设备，这会大大减少 CTV 库存的填充量。
- 使用 **vpos=[avail.type]** — MediaTailor 不支持该`[avail.type]`变量。这会解析为空字符串，这可能会导致请求失败或低价值的广告响应。
- 会话初始化时未传递玩家参数 -您的 ADS 模板网址引用玩家参数（例如`[player_params.ppid]`），但会话初始化调用未传递相应的值。GAM 收到该宏的空值或空值，这可能导致广告请求返回未填充值。

使用广告中断抑制功能自定义广告中断行为

在中创建配置时 AWS Elemental MediaTailor，您可以指定可选的广告插播配置设置来控制广告插播时间的行为，包括配置广告间隔抑制功能。这使您可以为视频内容量身定制广告插播体验，以满足您的特定要求。

兼容性限制

您不能对以下内容使用广告间隔抑制：

- VOD 和 Live-to-VOD 工作流程。仅支持实时工作流程。
- Server-guided 广告插入 (SGAI) 方法。 Server-guided 方法对广告决策的处理方式不同，不需要抑制配置。

主题

- 配置广告间隔抑制

配置广告间隔抑制

您可以配置 MediaTailor 为跳过直播内容的广告时段个性化设置。这被称为广告中断抑制或广告抑制。本主题向您展示了如何配置广告中断屏蔽功能，还说明了配置广告中断抑制的工作原理。

广告中断抑制可用于以下用例：

- 大清单回顾窗口 – 如果查看者在清单的直播边缘开始播放，但回顾窗口很大，您可能只希望在查看者开始观看之后插入广告。或者，在清单中为总回顾窗口的一部分插入广告。您可以配置广告抑制，以便在直播边缘或后面的指定时间范围内对广告时段进行 MediaTailor 个性化设置。
- Mid-break 加入 — 如果观众在广告中断期间开始观看直播视频，则该用户很可能会更改频道而不观看广告。借助广告抑制功能，如果广告插播是在观众加入直播之前开始的，则可以跳过广告时段个性化设置。

配置广告抑制

要使用广告抑制，您可以通过以下方式配置利用抑制模式、利用抑制值和利用抑制填充策略：

- 在 MediaTailor 控制台中
- 使用 AWS Command Line Interface (AWS CLI)

- 使用 MediaTailor API 或作为客户端播放会话请求中的参数

有关带参数的配置的信息，请参见配置广告禁止参数 – 播放会话请求。

广告抑制配置参数

您可以选择开启或关闭广告抑制。如果您开启广告抑制功能，则可以指定该抑制是在直播边缘之后还是在直播的直播播放边缘之前发生。无论哪种情况，您还要指定一个相对于直播边缘的时间，该时间 MediaTailor 不对广告进行个性化设置。当您开启广告抑制功能时，您可以指定一种利用抑制策略，该策略在会话中途开始时 MediaTailor 使用部分广告插播填充。

以下是广告禁止配置参数：

- Avail suppression mode (效用禁止模式) – 设置广告禁止模式。默认情况下，广告禁止功能关闭。可接受的值：OFFBEHIND_LIVE_EDGE、或AFTER_LIVE_EDGE。
  - OFF: 没有广告抑制功能，可对所有广告 MediaTailor插播进行个性化设置。
  - BEHIND_LIVE_EDGE: MediaTailor 不会对在直播边缘之前开始的广告时段进行个性化设置，减去 Avai l 抑制值。这会影响整个广告时段，而不仅仅是单个广告的可用性。
  - AFTER_LIVE_EDGE: MediaTailor 不对实时边缘内的广告插播进行个性化设置，再加上 Avai l 抑制值。可以将其配置为影响整个广告时段或允许部分填充可用的广告。
- Avail suppression value (效用禁止值) – 相对于直播中的实时边缘的时间。Accepted value (接受的值)：以 HH:MM:SS 格式表示的时间值。
- A@@ vail抑制填充策略-定义 MediaTailor 适用于Ava il抑制模式的策略。可接受的值：PARTIAL_AVAIL、FULL_AVAIL_ONLY。
  - BEHIND_LIVE_EDGE模式始终使用FULL_AVAIL_ONLY抑制策略。
  - AFTER_LIVE_EDGE模式可用于在会话中途开始时调用PARTIAL_AVAIL广告插播时间填充。

广告抑制设置示例

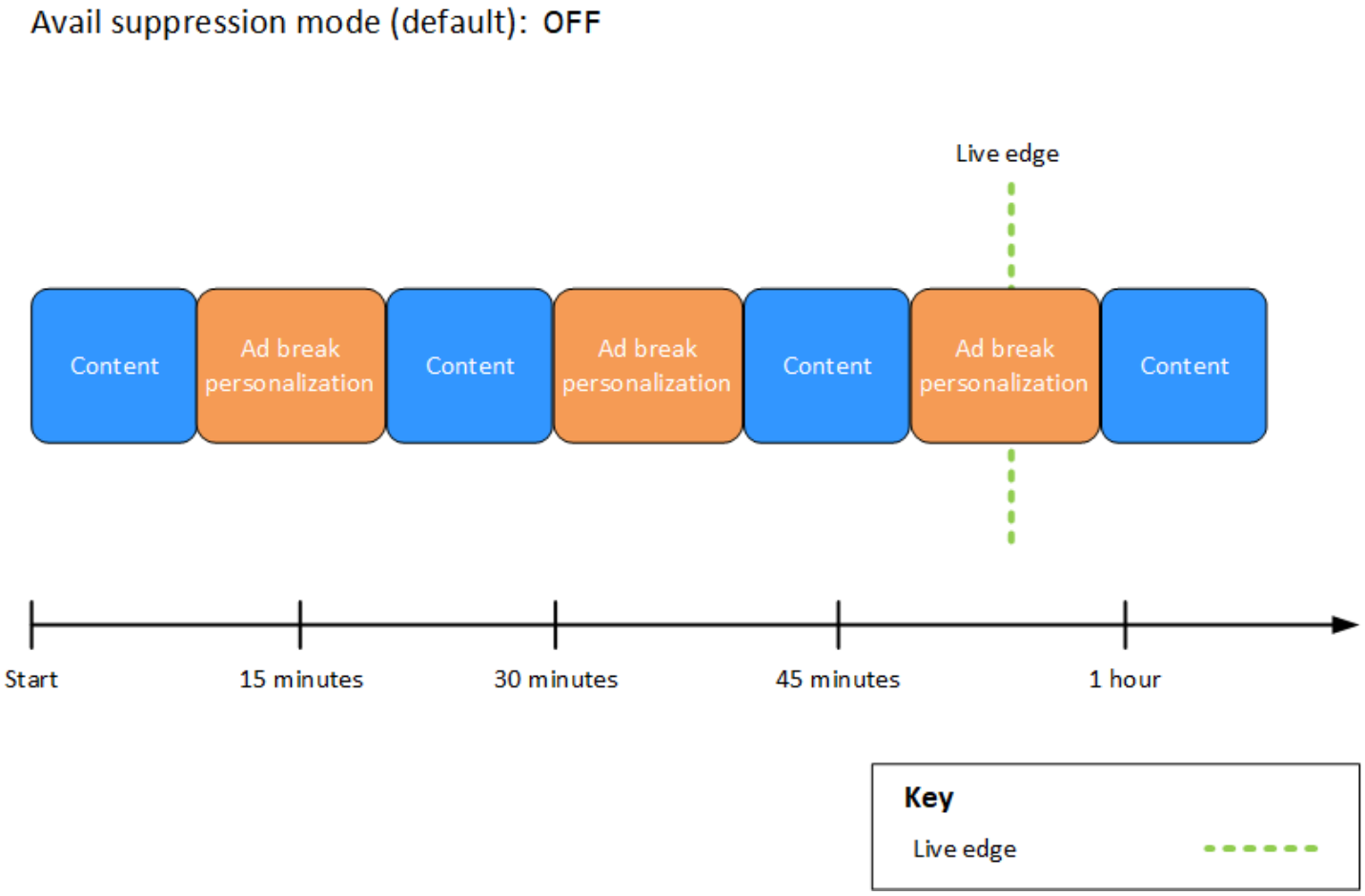
通过广告抑制配置参数相互交互的方式，您可以指定几种不同的方式来处理广告抑制和在直播前后、直播边缘或直播后进行广告抑制和可用性填充。本节提供了一些示例，向您展示了其中一些交互。使用这些示例来帮助您设置适合您特定情况的配置参数。

以下是广告抑制设置的示例：

Example 1：无广告禁止

当广告抑制模式为时OFF，没有广告抑制功能，并且会对所有广告 MediaTailor 插播进行个性化设置。

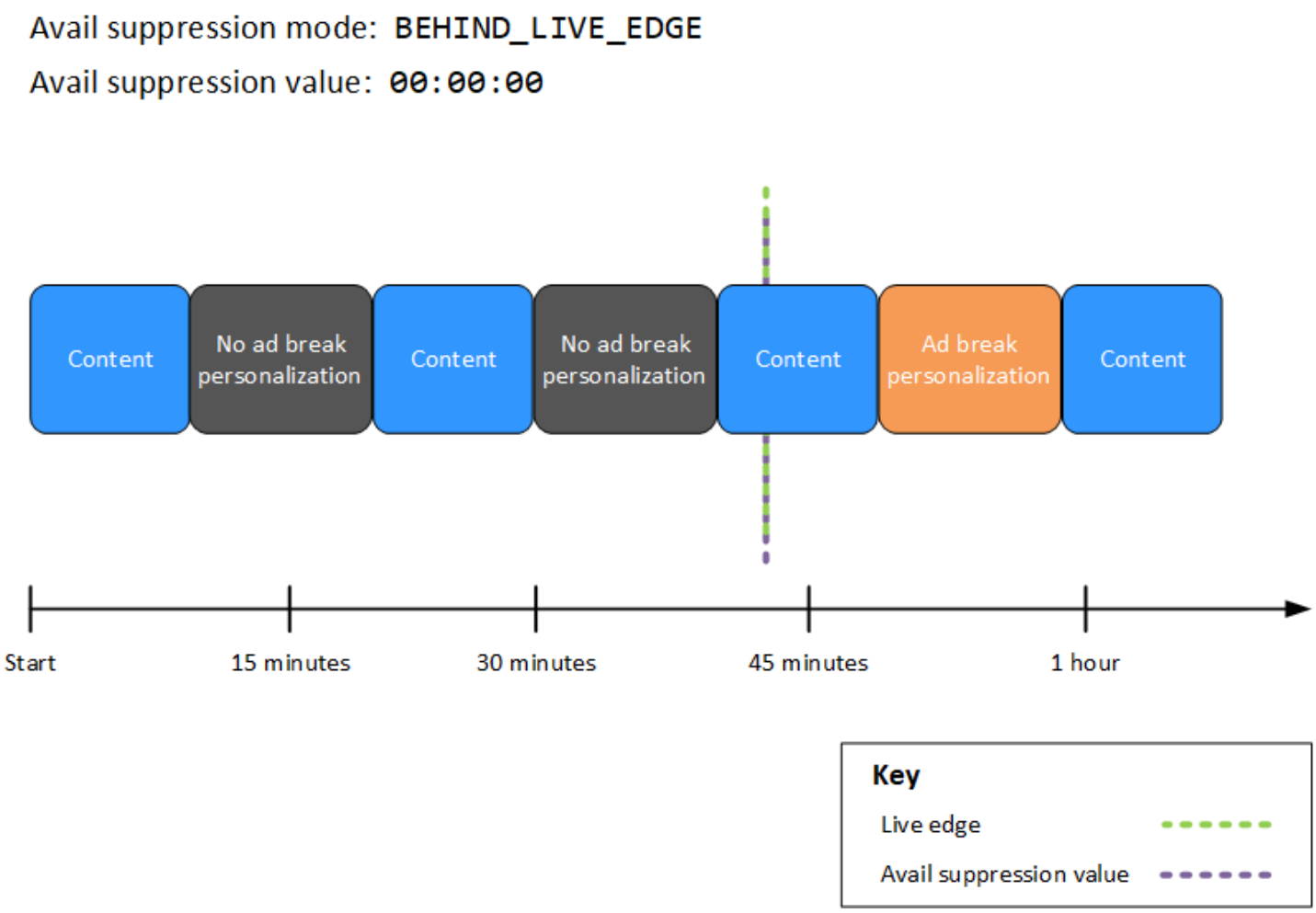
在下图中，各种方块沿着从左向右移动的时间轴水平排列。每个区块代表直播内容或个性化广告插播时间段的一段时间。虚线表示直播的当前直播边缘。在直播边缘之前有两个广告中断，另一个广告插播正在直播边缘进行中。如图所示，当广告抑制模式为时OFF，会对时间轴上直播边缘之前发生的所有广告中断进行 MediaTailor 个性化设置。 MediaTailor 还可以在直播边缘对正在进行的广告插播进行个性化设置。



Example 2：BEHIN **D_LIVE_EDGE ############**同步

当可用抑制模式设置为BEHIND_LIVE_EDGE且可用抑制值设置为时00:00:00，可用抑制值将与实时边缘同步。 MediaTailor 不会对任何在直播边缘或之前开始的广告时段进行个性化设置。

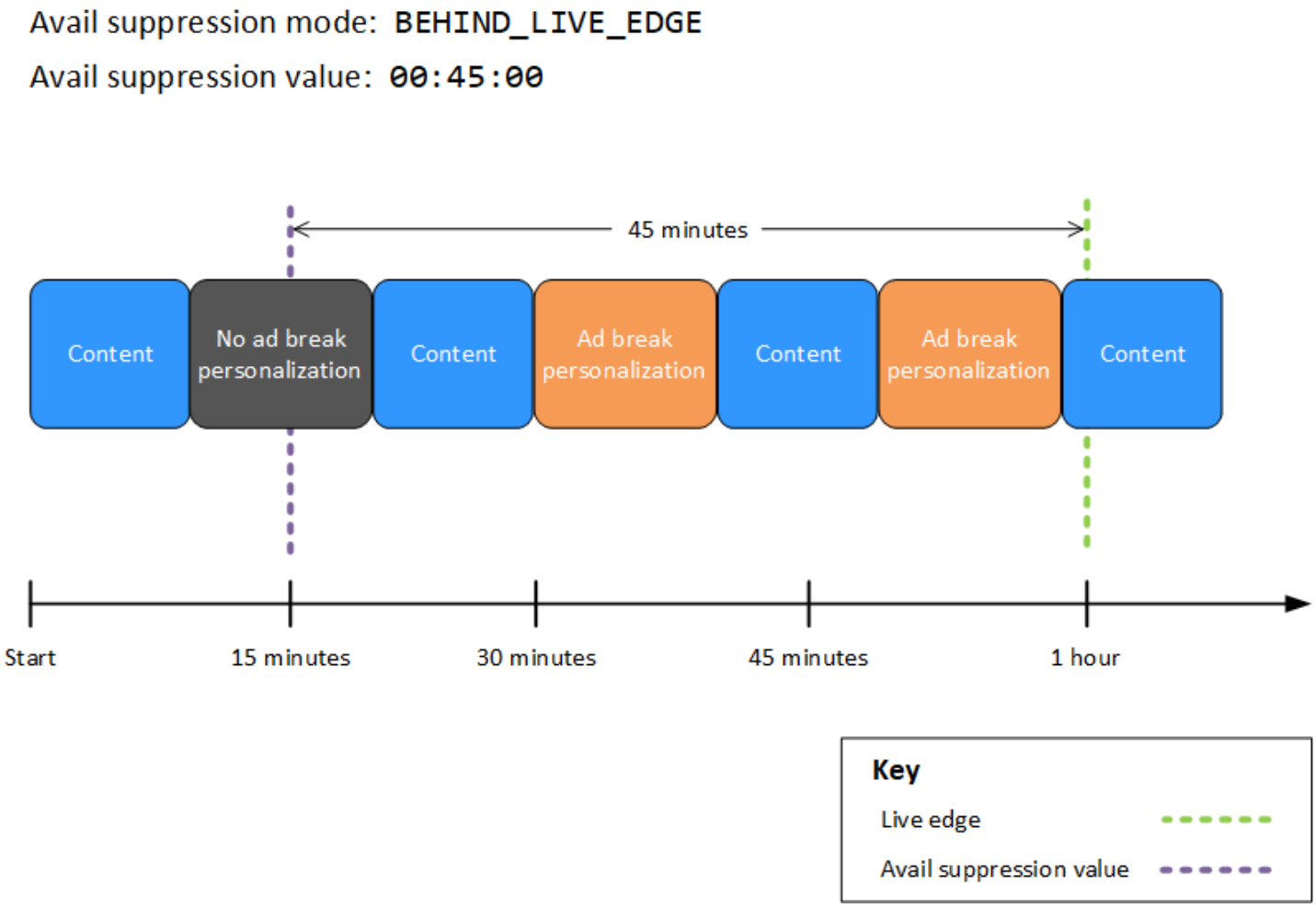
在下图中，各种方块沿着从左向右移动的时间轴水平排列。每个区块代表直播内容、个性化广告插播或非个性化广告插播时间段的一段时间。虚线表示直播的当前直播边缘。另一条虚线（表示设置为的可用抑制值）与实时边缘的虚线重叠。00:00:00两个广告中断发生在直播边缘之前，另一个广告中断发生在直播边缘之后。如图所示，如果将可用抑制模式设置为BEHIND_LIVE_EDGE，并且将可用抑制值设置为00:00:00使其与实时边缘同步，则 MediaTailor 不会对时间轴上直播边缘之前出现的任何广告中断进行个性化设置。 MediaTailor 对直播边缘之后出现的广告插播进行个性化设置。



Example 3：BEHIND_LIVE **_EDGE ###############**

当广告抑制模式设置为时BEHIND_LIVE_EDGE， MediaTailor 不会在该时间或之前对任何广告插播进行个性化设置。在此示例中，对比直播边缘落后 45 分钟的广告时段进行 MediaTailor 个性化设置。 MediaTailor 不会对比直播边缘落后超过 45 分钟的广告时段进行个性化设置。

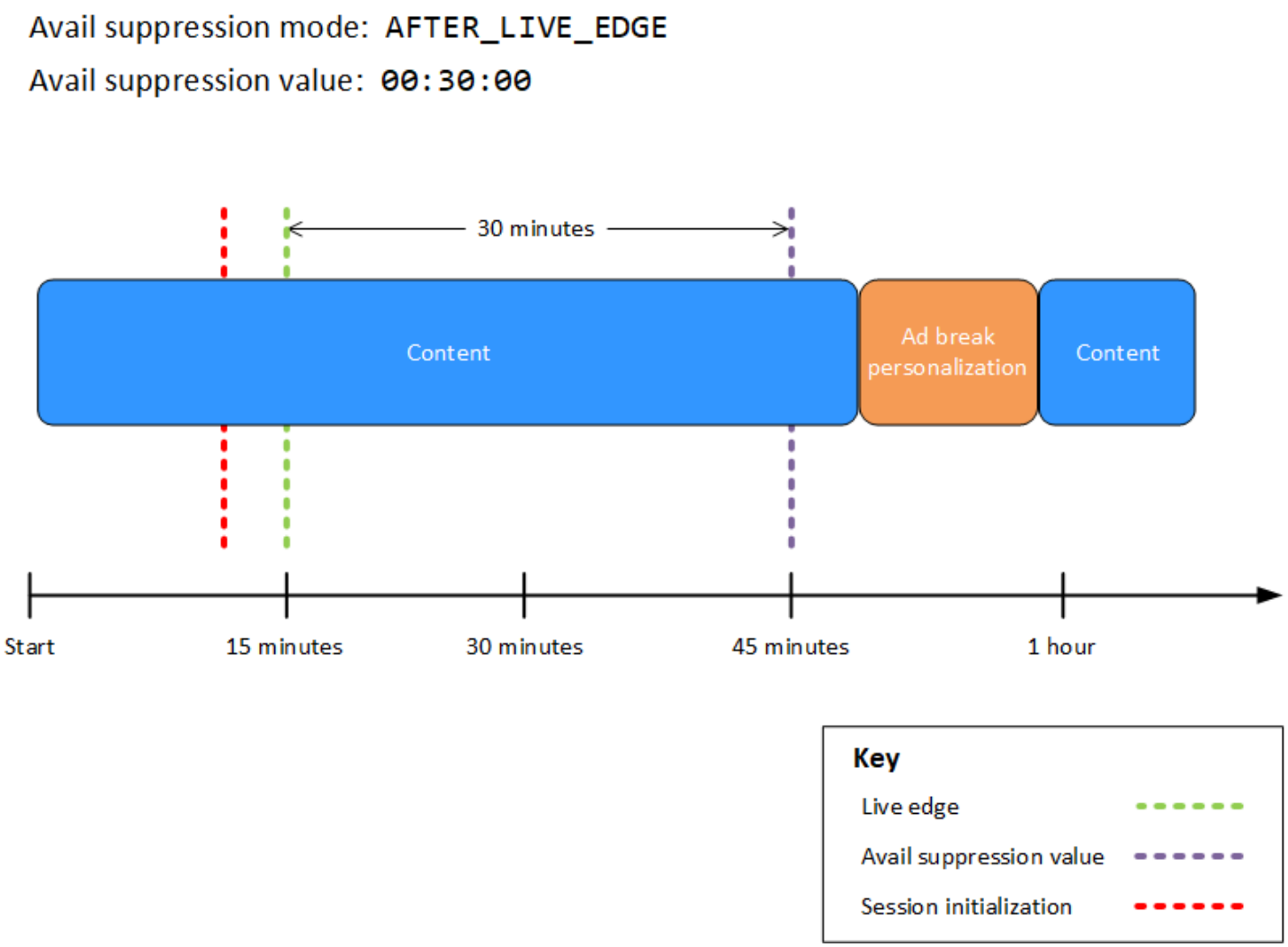
在下图中，各种方块沿着从左向右移动的时间轴水平排列。每个区块代表直播内容、个性化广告插播或非个性化广告插播时间段的一段时间。虚线表示直播的当前直播边缘。另一条虚线，表示设置为的可用抑制值00:45:00，出现在时间线前 45 分钟，相对于实时边缘的虚线。虚线之间的 45 分钟时间段表示可用性抑制期。在禁用期开始时，广告中断正在进行中。另外两次广告中断发生在禁止使用期内。如图所示，如果将可用抑制模式设置为BEHIND_LIVE_EDGE，并且将可用抑制值设置为在实时边缘00:45:00后面，则会对在可用抑制期内发生的任何广告中断进行 MediaTailor 个性化设置。 MediaTailor 在禁止使用期开始时，不会对正在进行的广告中断进行个性化设置。



Example 4：AF **TER_LIVE_EDGE** 广告被屏蔽，在广告屏蔽期间没有广告中断

当广告屏蔽模式设置为AFTER_LIVE_EDGE且可用抑制值大于零时，在会话经过的时间达到该值之前， MediaTailor 不会对任何广告中断进行个性化设置。

在下图中，各种方块沿着从左向右移动的时间轴水平排列。每个区块代表直播内容或个性化广告插播时间段的一段时间。虚线表示直播的当前直播边缘。另一条虚线，表示设置为的可用抑制值00:30:00，出现在时间轴中 30 分钟后，相对于实时边缘的虚线。第三条虚线代表会话初始化，相对于实时边缘的虚线，出现在时间轴的早期。实时边缘时间和可用抑制值时间之间的 30 分钟时间段表示可用抑制周期。广告中断发生在广告屏蔽期之后。如图所示，如果将可用抑制模式设置为AFTER_LIVE_EDGE，则将可用抑制值设置为在实时边缘00:30:00之后，并且会话初始化发生在实时边缘之前，会话初始化会话，从而对在可用抑制期之后出现的任何广告中断进行 MediaTailor 个性化设置。

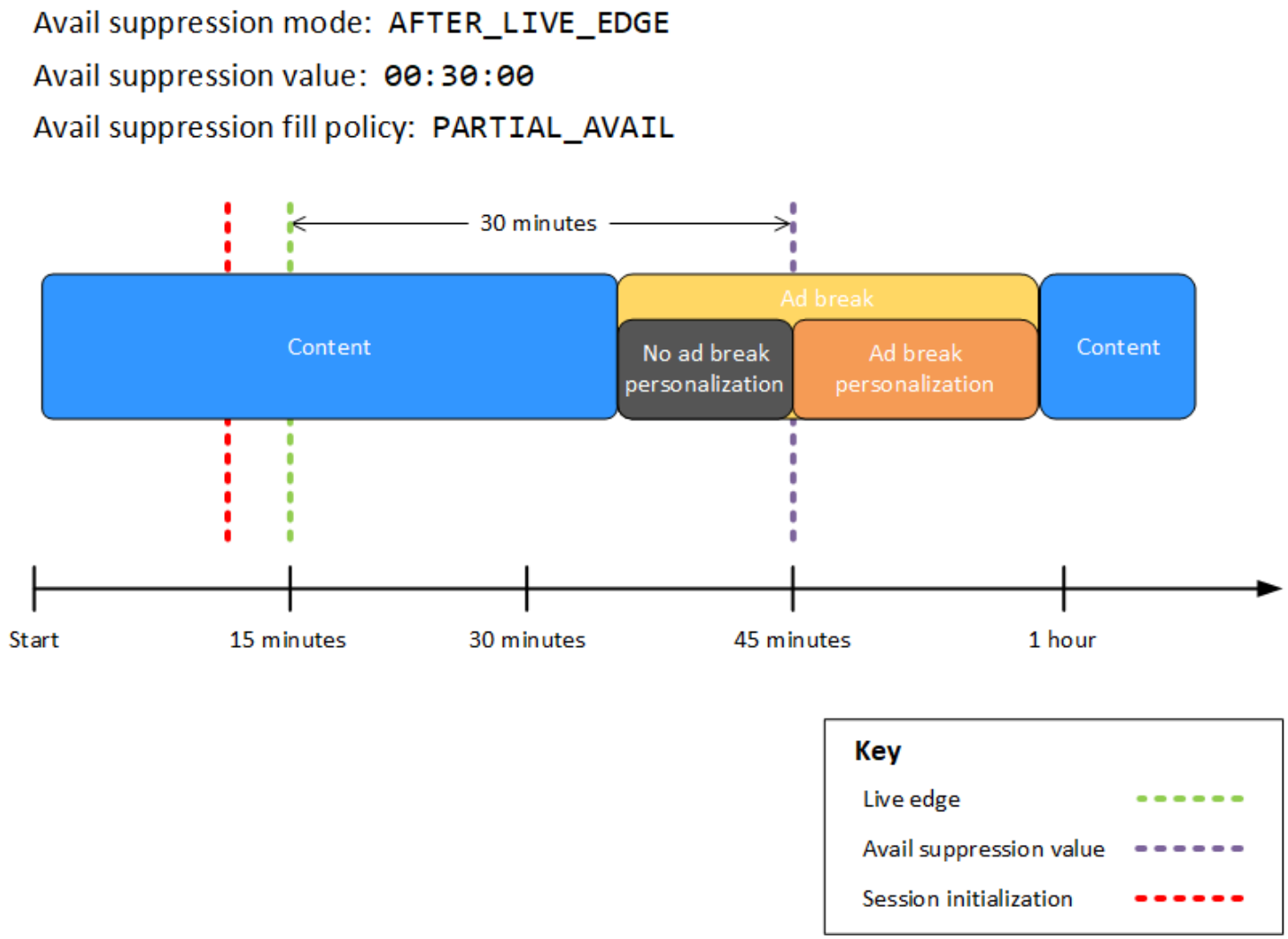


Example 5：AF **TER_LIVE_EDGE** 广告抑制，采用 **PARTIAL_AVAI** L 填充政策，在广告屏蔽期结束时正在进行广告中断

当广告屏蔽模式设置为AFTER_LIVE_EDGE且可用抑制值大于零时，在会话经过的时间达到该值之前， MediaTailor 不会对任何广告中断进行个性化设置。

在下图中，各种方块沿着从左向右移动的时间轴水平排列。每个区块代表直播内容、个性化广告插播或非个性化广告插播时间段的一段时间。虚线表示直播的当前直播边缘。另一条虚线，表示设置为的可用抑制值00:30:00，出现在时间轴中 30 分钟后，相对于实时边缘的虚线。第三条虚线代表会话初始化，相对于实时边缘的虚线，出现在时间轴的早期。实时边缘时间和可用抑制值时间之间的 30 分钟时间段表示可用抑制周期。在禁用期结束时，广告中断正在进行中。如图所示，当将可用抑制模式设置为AFTER_LIVE_EDGE，将可用抑制值设置为在实时边缘00:30:00之后，将可用抑制填充策略设置为PARTIAL_AVAIL，并且会话初始化发生在实时边缘之前，会话会对在可用抑制期之后出现的任何广告中断进行 MediaTailor 个性化

设置。对于在广告屏蔽期结束时正在进行的广告中断， MediaTailor 请对广告中断中在限制使用期之后发生的部分进行个性化设置，但不会对广告中出现的广告中断部分进行个性化设置。

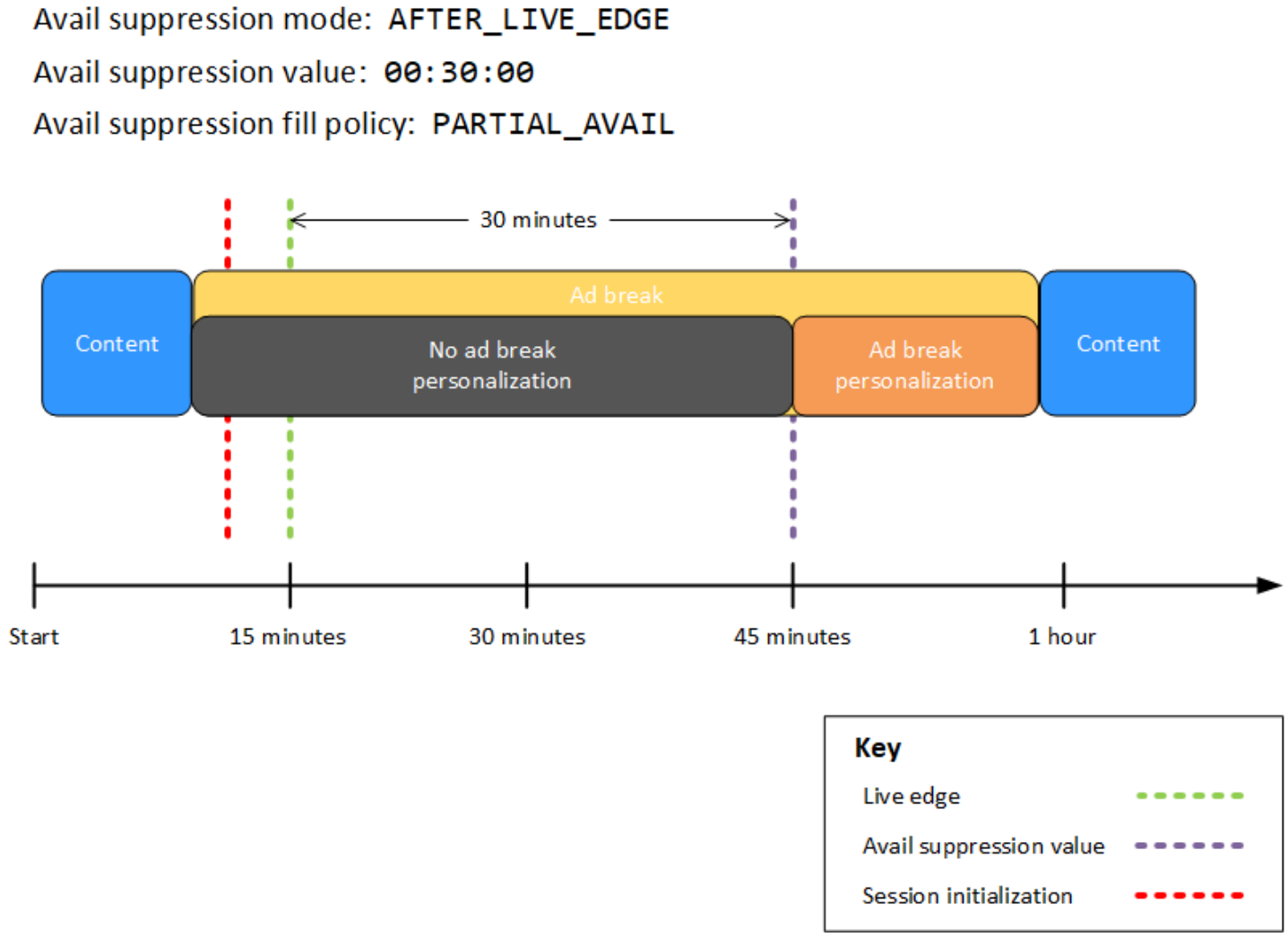


Example 6：AF **TER_LIVE_EDGE** 广告抑制，采用 **PARTIAL_AVAI** L 填充政策，从会话初始化之前到可用屏蔽期结束之后，广告中断正在进行中

当广告屏蔽模式设置为AFTER_LIVE_EDGE且可用抑制值大于零时，在会话经过的时间达到该值之前， MediaTailor 不会对任何广告中断进行个性化设置。

在下图中，各种方块沿着从左向右移动的时间轴水平排列。每个区块代表直播内容、个性化广告插播或非个性化广告插播时间段的一段时间。虚线表示直播的当前直播边缘。另一条虚线，表示设置为的可用抑制值00:30:00，出现在时间轴中 30 分钟后，相对于实时边缘的虚线。第三条虚线代表会话初始化，相对于实时边缘的虚线，出现在时间轴的早期。实时边缘时间和可用抑制值时间之间的 30 分钟时间段表示可用抑制周期。广告中断时间从会话初始化之前到可用性抑制期之后的时间进行。如图所示，当将可用抑制模式设置为AFTER_LIVE_EDGE，将可用抑制值设置为在实时边

缘00:30:00之后，将可用抑制填充策略设置为PARTIAL_AVAIL，并且会话初始化发生在实时边缘之前，会话会对在可用抑制期之后出现的任何广告中断进行 MediaTailor 个性化设置。对于在广告屏蔽期之前、期间和之后正在进行的广告中断， MediaTailor 请对广告中断中在可用性抑制期之后出现的部分进行个性化设置，但不会对广告插播时间中在限制使用期之前或期间发生的部分进行个性化设置。



Example 7：AF **TER_LIVE_EDGE** 广告被屏蔽且在广告屏蔽期开始时正在进行广告中断

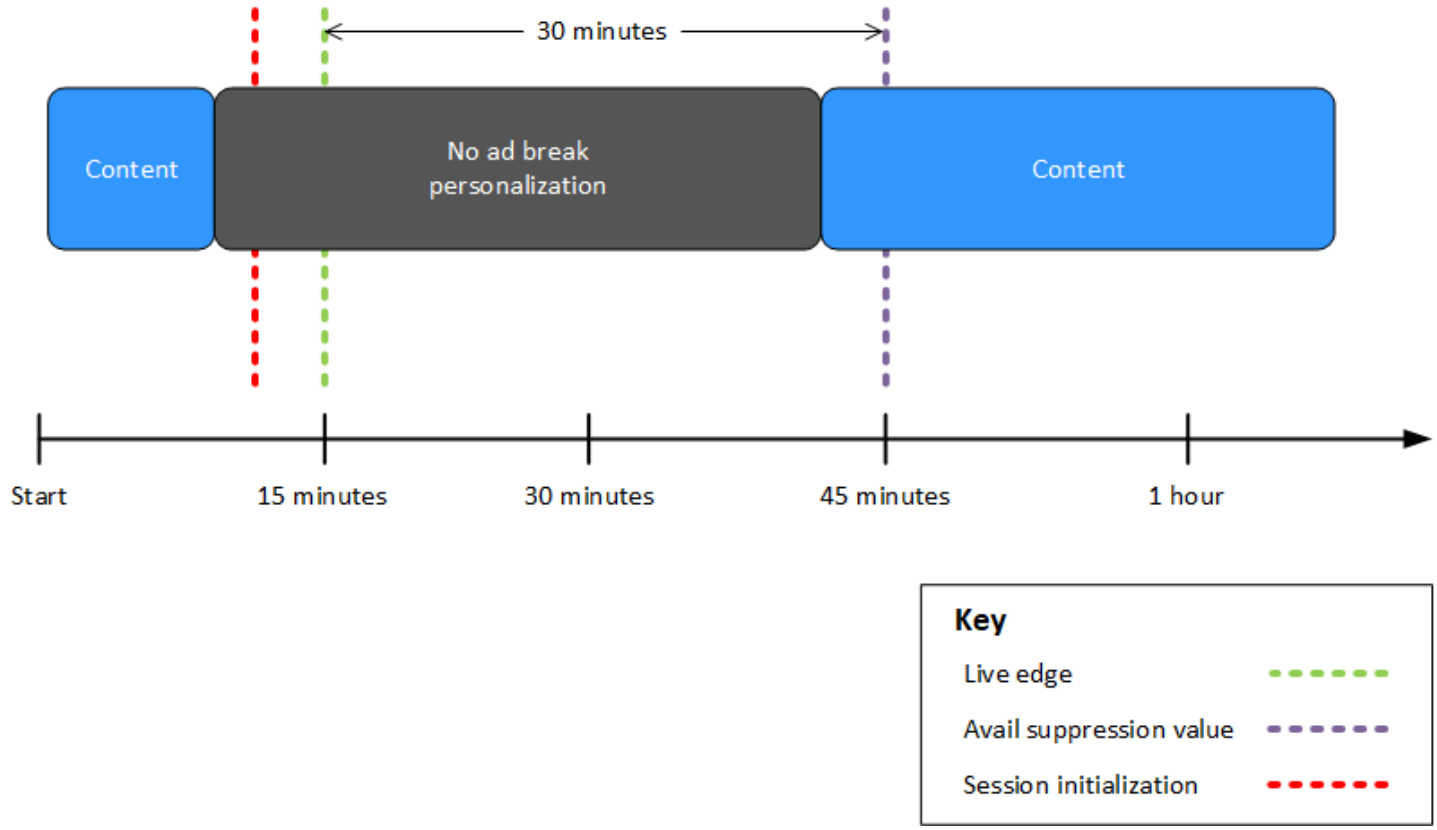
当广告屏蔽模式设置为AFTER_LIVE_EDGE且可用抑制值大于零时，在会话经过的时间达到该值之前， MediaTailor 不会对任何广告中断进行个性化设置。

在下图中，各种方块沿着从左向右移动的时间轴水平排列。每个区块代表直播内容或非个性化广告时段的播放时间段。虚线表示直播的当前直播边缘。另一条虚线，表示设置为的可用抑制值00:30:00，出现在时间轴中 30 分钟后，相对于实时边缘的虚线。第三条虚线代表会话初始化，相对于实时边缘的虚线，出现在时间轴的早期。实时边缘时间和可用抑制值时间之间的 30 分钟时间段表示可用抑制周期。广告插播时

间从会话初始化前一段时间一直持续到禁用期内。如图所示，当广告抑制模式设置为时AFTER_LIVE_EDGE，可用抑制值设置为在直播边缘00:30:00之后，并且会话初始化发生在直播边缘时间之前，但在广告插播开始之后， MediaTailor 不会对该广告插播进行个性化设置。

Avail suppression mode: AFTER_LIVE_EDGE

Avail suppression value: 00:30:00



配置广告禁止参数 – 播放会话请求

您可以在服务器端或客户端的初始播放会话请求中通过参数配置广告抑制设置。 MediaTailor如果您已经通过 MediaTailor 控制台或 AWS Elemental MediaTailor API 配置了广告抑制设置，则这些参数会覆盖这些设置。

广告禁止功能需要使用效用禁止模式和效用禁止值才能起作用。无法从不同的源配置这些参数。例如，您不能在 MediaTailor 控制台中配置一个参数，而使用查询参数配置另一个参数。

MediaTailor 支持以下广告抑制参数。

Name	说明	接受的值
availSuppressionMode	设置广告禁止模式。默认情况下，广告抑制为OFF。如果设置为BEHIND_LIVE_EDGE ，MediaTailor 则不会在aws.availSuppressionValue 时间之前或之后填充广告中断。如果设置为AFTER_LIVE_EDGE ，MediaTailor 则不会在广告屏蔽期内或之后填充广告插播时间。可用抑制周期从实时边缘时间跨度到时间，再加上额外的缓冲aws.availSuppressionValue 时间。	• OFF • BEHIND_LIVE_EDGE • AFTER_LIVE_EDGE
availSuppressionValue	相对于直播中的实时边缘的时间。	里面有 UTF-8 URL-encoded 时间码HH:MM:SS。例如，1 小时 30 分钟将表示为 01%3A30%3A00 。
availSuppressionFillPolicy	定义应用于无用抑制模式的策略。 BEHIND_LIVE_EDGE 始终使用全部可用性抑制策略。 AFTER_LIVE_EDGE 可用于在会话中途开始时调用部分广告插播时间填充。	• PARTIAL_AVAIL -不适用于BEFORE_LIVE_EDGE 抑制模式 • FULL_AVAIL_ONLY -AFTER_LIVE_EDGE 抑制模式的默认值

Server-side 配置

基本查询参数为 aws.availSuppression，后跟可选参数名称和值对。要构造查询，请在播放会话请求的末尾追加aws.availSuppression=到 MediaTailor，然后是参数名称和值。有关如何构建服务器端播放会话请求的更多信息，请参阅MediaTailor 服务器端广告跟踪和报告。

示例：HLS

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

Server-side 下表列出了查询语法。

查询字符串组件	说明
?	标记查询开始的受限字符。
aws.	基本查询，后跟由名称和值对构造的参数。有关所有可用参数的列表，请参阅配置广告禁止参数 – 播放会话请求。
=	将参数名称与值相关联。例如 aws.availSuppressionMode= *BEHIND_LIVE_EDGE* 。
&	连接查询参数。例如，aws.availSuppressionMode= *BEHIND_LIVE_EDGE* &aws.availSuppressionValue= *00:30:00*&aws.availSuppressionFillPolicy= *FULL_AVAIL_ONLY* >。

Client-side 配置

在客户的 POST 请求中加入availSuppression参数 MediaTailor。有关如何构建客户端播放会话请求的更多信息，请参阅Client-side 广告追踪。

示例：HLS

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

MediaTailor 保险杠广告插入

Bumpers 是简短的、不可跳过的视频或音频片段，在广告插播开始或结束之前播放。 AWS Elemental MediaTailor

以下条件适用于保险杠：

- 保险杠必须在 10 秒或更短的时间内。
- 保险杠可以在广告插播开始时、广告插播结束之前插入，或者两者兼而有之。
- 除非配置了前置片段，否则 Bumpers 会在播放会话中的每个广告时段播放。如果配置了前置滚动，则在前轮休息期间将不会播放保险杠。取而代之的是，他们将在预赛结束后的每个后续休息时间中进行比赛。
- 对于 HLS，您必须在每个 SCTE-35 `EXT-X-CUE-OUT`标签中包含该`duration`属性。
- 保险杠经过转码以匹配源内容。
- 保险杠不收费。

配置保险杠

要使用缓冲器，请使用 MediaTailor 控制台、 MediaTailor API 或 AWS Command Line Interface ()AWS CLI配置保险杠网址。您可以配置起动保险杠、末端保险杠或两者兼而有之。保险杠存储在服务器上，例如亚马逊简单存储服务 (Amazon S3) Service。保险杠网址表示存储的保险杠资产的位置。

起始和结束缓冲器 URL 示例：

起始保险杠网址：`https://s3.amazonaws.com/startbumperad`

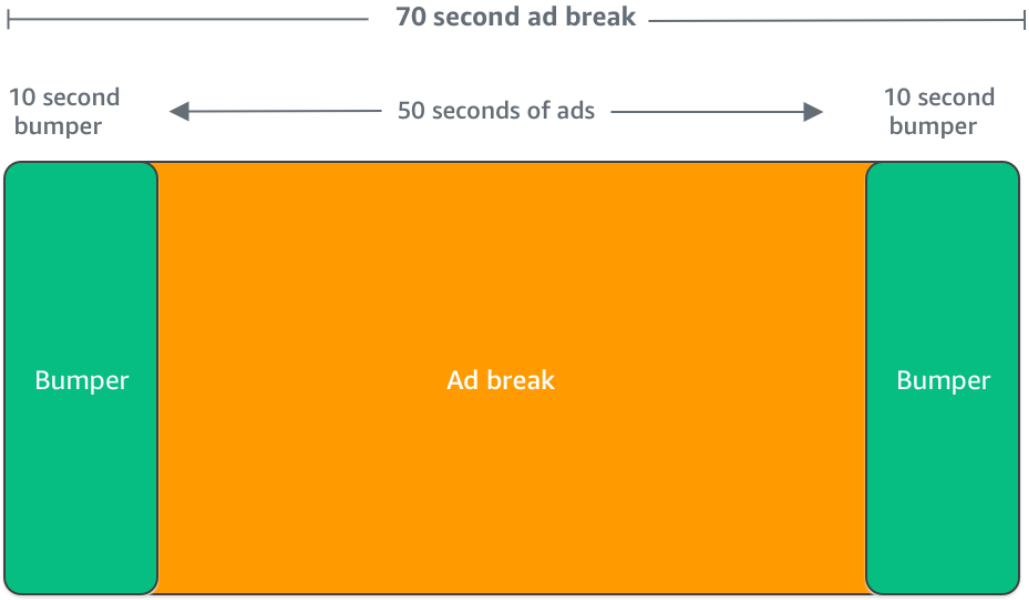
末端保险杠网址：`https://s3.amazonaws.com/endbumperad`

示例

以下是引擎广告行为的示例。

Example示例 1：起点和终点保险杠

在此示例中，启用了起点和终点保险杠。广告决策服务器有 50 秒的个性化广告来填补 70 秒的广告时段。10 秒的起始缓冲器在广告时段开始播放，50 秒的广告播放，然后播放 10 秒的结束缓冲器。



MediaTailor 片前广告插播

> **Note**
>
> 可配置的前置广告仅适用于直播工作流程。如需详细了解 VOD 的广告插入（包括前置投放）的工作原理，请参阅VOD 的广告拼接行为。

AWS Elemental MediaTailor 可以在播放会话开始时，也就是主内容开始之前插入广告。这些是前置式广告。

要插入前置片广告，请在配置的 “其他设置” 中填写直播前置广告决策服务器和直播前置广告允许的最大持续时间字段，如中所述。可选配置设置

1. MediaTailor 收到播放请求后，它会根据 MediaTailor 播放配置中的以下字段向 ADS 发送预贴片广告请求：
   - 直播前置广告决策服务器是用于 MediaTailor 发送前置广告请求的广告决策服务器 (ADS) 网址。

- 直播前置片允许的最大时长是前置广告的最大总时长。 MediaTailor 根据允许的最大持续时间采取以下操作：
  - 如果 ADS 响应中广告的总时长小于您在直播前置片中允许的最大时长中给出的值，则会 MediaTailor 插入所有广告。最后一个广告完成后， MediaTailor 立即返回到底层内容。
  - 如果 ADS 响应中广告的总时长超过了你在直播前置片中允许的最大时长中给出的值，请 MediaTailor 选择一组符合时长的广告，而不必超过该时长。 MediaTailor 无需剪辑或截断即可插入这些广告。 MediaTailor 当最后一个选定的广告完成后，返回到底层内容。

2. 当 MediaTailor 收到来自 ADS 的预贴片回复时，它会操纵清单以添加指向前贴片广告的链接。 MediaTailor 按如下方式计算前置广告插播的开始时间：
   - 对于 DASH，公式为 `(publishTime - availabilityStartTime) - max(suggestedPresentationDelay, minBufferTime)`。
   - 对于 HLS，公式为 `max(2*EXT-X-TARGETDURATION, EXT-X-START:TIMEOFFSET)`。
3. MediaTailor 决定对任何非预贴广告时段采取什么措施。如果前置片段与另一个广告插播时间重叠，则 MediaTailor 不会对广告插播的重叠部分进行个性化设置。

MediaTailor slate 广告插入

> **Note**
>
> Slate 仅适用于直播和 live-to-VOD工作流程。

使用 AWS Elemental MediaTailor，您可以为广告插播时间指定一个板块广告。Slate 是插入到直播中的默认 MP4 资源，例如静止图像或循环播放的视频，用于代替直播内容播放。

AWS Elemental MediaTailor 在以下特定情况下，在广告中断时段显示 slate：

- 填补尚未被个性化广告取代的休息时间剩余时间
- 如果广告决策服务器 (ADS) 以空的 VAST 或 VMAP 响应进行响应
- 用于错误情况，例如 ADS 超时或 ADS 的 HTTP 500 错误
- 如果广告无法通过以下方式插入 MediaTailor （ 例如，当广告尚未完成转码时 ）

如果您未配置 slate， MediaTailor 则在满足上述条件之一时默认为底层内容流。

配置名单

您可以在 MediaTailor 控制台的其他配置窗格中指定名单。 MediaTailor 从您指定的 URL 中下载 slate，然后将其转码为与内容相同的格式副本。您可以通过控制台中的可选个性化阈值配置来MediaTailor控制显示板块的最长时间。有关更多信息，请参阅 the section called “个性化阈值的工作原理”。

Slate 必须是同时包含音频和视频的高质量 MP4 资产。对于非 vPaid 配置，配置清单是可选的，但对于 VPAID 工作流程，则必须进行配置。

> **Note**
>
> 如果承载画面的服务器使用 HTTPS，则其证书必须来自知名的证书颁发机构。它不能是自签名证书。如果您使用自签名证书，则 AWS Elemental MediaTailor 无法从内容来源检索名单并将其拼接到清单中。

个性化阈值的工作原理

个性化阈值定义了广告插播时间内允许的广告填充不足的最长持续时间（以秒为单位）。此功能专门适用于直播和 VOD 直播中的广告替换，而不是广告插入，因为它依赖于底层内容流。

行为因三种情况而异：

1. 禁用个性化设置时：
   a. Slate 将在整个未填充时间内被插入
   b. 配置后将插入起点/终点保险杠（有关更多信息，请参阅）MediaTailor 保险杠广告插入
   c. 广告将照常插入
2. 启用个性化设置且阈值小于中断持续时间时：
   a. 如果未填写时间超过个性化阈值：
      i. MediaTailor 放弃广告插播的个性化设置
      ii. 显示了底层内容
      iii. 未插入广告、板条或保险杠
   b. 如果未填写时间小于个性化阈值：
      i. 广告和牌照已插入
      ii. 如果配置了保险杠，则会插入保险杠

3. 启用个性化设置且阈值大于中断持续时间时：
   a. 广告将被插入
   b. Slate 将在广告插播时间段的任何剩余时间内插入
   c. 如果配置了保险杠，则会插入保险杠

下表提供了 HLS 和 DASH 协议中特定场景的详细行为：

详细行为矩阵

场景	个性化已禁用	在休息时间内启用个性化设置	启用个性化设置的时间大于休息时长
清空 VAST 或 VMAP	已插入 Slate	未插入石板	已插入 Slate
广告超时	已插入 Slate	未插入石板	已插入 Slate
广告不可用 (Vast 404)	已插入 Slate	未插入石板	已插入 Slate
广告的持续时间比广告插播时间长	已插入广告	未插入广告	已插入广告
填写替换广告未完全使用的时间	已插入 Slate	如果个性化阈值大于未填充时间：未 ads/slates 插入，否则：已插入广告和牌照	已插入 Slate
使用空白 VAST 进行预览	已插入 Slate，未插入预卷	未插入石板，未插入预卷	已插入 Slate，未插入预卷
带有 ADS 超时的预贴片	已插入 Slate，未插入预卷	未插入石板，未插入预卷	已插入 Slate，未插入预卷
广告不可用时预先投放 (Vast 404)	已插入 Slate，未插入预卷	未插入石板，未插入预卷	已插入 Slate，未插入预卷
带空 VAST 的保险杠	板岩已插入，保险杠已插入	没有插入石板，没有插入保险杠	板岩已插入，保险杠已插入

场景	个性化已禁用	在休息时间内启用个性化设置	启用个性化设置的时间大于休息时长
带有 ADS 超时的保险杠	板岩已插入，保险杠已插入	没有插入石板，没有插入保险杠	板岩已插入，保险杠已插入
广告不可用时的碰撞 (Vast 404)	板岩已插入，保险杠已插入	没有插入石板，没有插入保险杠	板岩已插入，保险杠已插入
用于填补广告替换未充分利用的时间的保险杠	已插入 Slate、插入保险杠、插入广告	如果个性化阈值大于未填写时间：未插入 bumpers/ads/slates inserted, else: No bumpers/ads/slates	已插入 Slate、插入保险杠、插入广告

重要注意事项

在使用 slate 和个性化阈值时，请记住以下几点。

- 这种行为在 HLS 和 DASH 协议中都是一致的
- 个性化阈值功能仅在配置 slate 时适用
- 配置 slate 且未配置个性化阈值时，slate 将在完整的 Origin 可用时长或填写广告后的剩余时间内播放
- 对于 VPAID 广告，在 VPAID 广告的持续时间内 MediaTailor 插入 slate，以便为视频播放器将要插入的广告留出空间

Slate 配置和 VPAID

> **Important**
>
> 使用 VPAID 广告时，必须配置 Slate。 MediaTailor 插入 slate 以存放视频播放器将要插入的 VPAID 广告的空间。为了适应用户的互动性，广告时长可能略长于 VPAID 广告时长。

然后，视频播放器会根据 MediaTailor 返回的客户端报告元数据处理 VPAID 广告，如中所述。VPAID 要求有关客户端报告的信息，请参阅the section called “Client-side 追踪”。

中的广告转码 AWS Elemental MediaTailor

AWS Elemental MediaTailor 对广告、bumpers 和 slate 进行转码，使其与原始视频流的格式和规格相匹配，从而确保广告间隔期间的无缝播放。 MediaTailor 支持两种转码方法：动态转码（默认）和自定义转码配置文件。了解这些方法之间的差异将有助于您确定工作流程的正确配置。

主题

- 动态转码
- 自定义转码配置文件

动态转码

动态转码是中的默认广告转码行为。 MediaTailor当需要对广告、bumper 或 slate 进行转码时，无需您进行任何配置，即可在运行时 MediaTailor 自动检查来源清单并获取转码设置。

动态转码处理什么

动态转码会自动处理以下HLS和工作DASH流程。

- 标准编解码器匹配 — MediaTailor 匹配源流的编解码器配置文件。请注意，不能保证编解码器级别匹配。
- 音频标准化 — 使用默认目标级别 MediaTailor 自动将音频标准化应用于转码后的广告。-24 LKFS这样可以确保广告音频级别与您的来源内容保持一致，而无需进行任何其他配置。如果您的工作流程需要除默认值之外的自定义音频标准化目标-24 LKFS，则需要自定义转码配置文件。
- 标准区段长度匹配-默认情况下，使用 2 秒区段对广告进行 MediaTailor 转码。如果您的玩家需要分段长度匹配，则可以启用该`calculateSegmentLength`标志，这会 MediaTailor 通过分析来自原点的前 10 个片段并选择最常见的持续时间作为分段长度来自动计算正确的分段长度。默认情况下，此标志处于禁用状态，必须由 AWS Support 启用。联系 AWS Support `calculateSegmentLength` 申请激活您的账户或特定配置。
- 标准WebVTT字幕-对于HLS工作流程，当字幕出现在原始清单中时，会在个性化的多变体播放列表中 MediaTailor 保留WebVTT字幕组引用。在广告区段中， MediaTailor 插入空白WebVTT区段以保持字幕轨迹的连续性，因为广告素材通常不包含字幕轨道。对于DASH工作流程， MediaTailor 要求对原始清单中的字幕改编集进行分段（如视频和音频）。由构造的广告时段 MediaTailor

仅包含广告素材提供的改编集；由于广告素材通常不包含字幕曲目，因此广告时段将不包含字幕。AdaptationSet内容时段的字幕改编集照常保留。

- CMAF格式支持- MediaTailor 检测HLS源流是使用 CMAF (fMP4) 细分还是标准TS区段，并对广告进行转码以进行相应的匹配。对于DASH，透明地支持CMAF打包，因为DASH本质上使用 ISO-BMFF (fMP4) 作为其容器格式。这两种协议都不需要其他配置。

  > **Note**
  >
  > MediaTailor 不支持使用 HVC1 (HEVC) 作为带有TS分段的编解码器的HLS直播。根据《Apple 设备的 HLS 创作规范》第 1.5 节，HEVC视频的容器格式必须为。fMP4 HLS因此，使用的直播HEVC必须CMAF与分段的 MP4 片段一起使用。

- T@@ rickplay 个性化 — MediaTailor 自动对广告进行转码，其中包含两HLS者兼有 trickplay 变体和相关图像流（包括 I-frame 清单、缩略图和图像流）DASH，匹配原始内容规范。此功能不再需要自定义转码配置文件。
- 压缩DASH清单 — MediaTailor 自动支持压缩DASH清单，可将SegmentTemplate元素从单个Representation元素提升到AdaptationSet级别，从而缩小清单的总体大小并提高交付效率。此功能不再需要自定义转码配置文件。

当动态转码可能还不够时

动态转码可以处理大多数广告转码场景。但是，在某些情况下，可能需要额外的配置或自定义转码配置文件。

- 比特率匹配-默认情况下，动态转码不能保证精确的比特率匹配。如果您的工作流程要求广告比特率与来源比特率保持一致，则可以启用该`enableBitrateRounding`标志，该标志允许对高达 5% 的比特率进行四舍五入，以更好地匹配来源值。此标志必须由 AWS Support 启用。联系 AWS Support `enableBitrateRounding` 申请激活您的账户或特定配置。如果您的工作流程要求比特率匹配超过 5%，则需要自定义转码配置文件。
- 除 MediaTailor 默认值之外的自定义音频标准化目标 -24 LKFS
- Non-integer 分段长度（例如，1.92 秒或 3.84 秒）
- 杜比视界音频
- 下游系统或播放器所需的特定 PMT、视频或音频 PID 值
- 自定义音频采样率（例如，44.1 kHz）
- 自定义像素长宽比 (PAR) — 例如，方形像素为 1:1，NTSC 宽屏为 40:33 D1/DV

- 特定的命名音轨或字幕轨道

自定义转码配置文件

自定义转码配置文件 (CTP) 是您注册时明确编写的 AWS Elemental MediaConvert 任务配置。MediaTailor当 CTP 与 MediaTailor配置关联时， MediaTailor 使用您定义的设置对每个广告、bumper 和 slate 进行转码，从而使您可以精确控制编解码器、比特率、音频标准化级别、片段长度、PID 值、采样率等。由于 CTP 完全覆盖动态转码，因此您必须在配置文件中明确配置所需的所有功能，包括特技游戏功能（I-frame 清单、缩略图和图像流）。

CTP 旨在匹配特定的源配置。如果来源内容与最初创建 CTP 的配置不同，则广告可能因为错误而无法正确转码。NO_VARIANT_MATCH当 CTP 输出与原始清单中存在的变体不一致时，就会发生这种情况。这适用于直播和VOD工作流程。例如，如果 CTP 不考虑VOD库中存在的所有变体组合，则包含来自不同来源的内容或使用不同的编码配置文件（例如不同的分辨率、比特率、编解码器或段长度）编码的库可能会产生NO_VARIANT_MATCH错误。如果您需要单个 CTP 来支持多个来源，或者需要一个包含混合编码配置文件的VOD库，则可以创建一个包含所有目标源变体的超集 CTP。但是，superset CTP 受到 36 个视频变体的硬性限制，这限制了单个配置文件可以容纳多少个来源或变体组合。

> **Important**
>
> 每当你更改编码器设置或使用新的VOD内容来源时，都必须保持CTP同步。否则，将导致NO_VARIANT_MATCH错误和广告转码失败。这包括诸如添加或删除分辨率、调整比特率、更改编解码器、修改分段长度或从具有不同编码配置文件的新提供商添加VOD内容之类的更改。

更新 CTP 时，请注意以下行为。

- 更新现有 CTP 仅会将新设置应用于新转码的广告。之前在旧 CTP 设置下进行转码的广告不会被重新转码。
- 创建新的 CTP 并将其与您的 MediaTailor 配置关联后，系统会提示您使用新的配置文件 MediaTailor 对所有广告进行重新转码，从而确保您的广告资源完全一致。

因此，当您的编码器设置发生变化或您加入新的内容源时，最佳做法是创建新的 CTP，而不是更新现有的 CTP，以确保所有广告（包括之前转码的广告）都能反映更新的设置。

何时使用自定义转码配置文件

当您的工作流程需要以下一项或多项时，请使用自定义转码配置文件。

- 除 MediaTailor 默认值之外的自定义音频标准化目标 -24 LKFS
- Non-integer 分段长度（例如，1.92 或 3.84）
- 杜比视界音频配置
- 下游系统或播放器所需的特定 PMT、视频或音频 PID 值
- 自定义音频采样率（例如，44.1 kHz）
- 带有配置标志的动态转码支持超过 5% 阈值的比特率匹配
- 自定义像素长宽比 (PAR) — 例如，方形像素为 1:1，NTSC 宽屏为 40:33 D1/DV
- 特定的命名音频或字幕曲目

自定义转码配置文件入门

只有列入许可名单的账户才能访问 MediaTailor 转码 API，这是注册 CTP 所必需的。要开始使用，请使用以下选项之一联系 AWS Support。

- 申请许可名单访问权限 — 要创建和管理您自己的 CTP，请联系 AWS Support，请求将您的账户添加到许可名单。将您的账户添加到许可名单后，您可以访问Transcoding API，以便在您已有配置的任何 AWS 地区注册和管理个人资料。 MediaTailor
- 请求 AWS Support 创建 CTP — 要让 AWS Support 代表您生成和注册 CTP，请联系 AWS Support，并提供 MediaTailor 可以访问的非DRM-protected 源终端节点 URL。AWS Support 将使用此终端节点来分析您的源配置并为您的账户生成相应的 CTP。

动态转码与自定义转码配置文件

使用下表来确定适合您的工作流程的方法。

动态转码与自定义转码配置文件对比

考虑因素	动态转码	自定义转码配置文件
设置工作	无	手动编写的 AWS MediaConvert Elemental 任务配置

考虑因素	动态转码	自定义转码配置文件
Trickplay 支持（HLS和DASH）	自动	必须在 CTP 中明确配置
紧凑型DASH清单	自动	必须在 CTP 中明确配置
CMAF格式支持	自动	必须在 CTP 中明确配置
原产地变更处理	自动	需要手动 CTP 更新
Multi-region 复杂性	无	必须按地区注册
音频标准化	-24 LKFS（默认值）	完全可配置
比特率匹配	通过`enableBitrateRounding` 标记进行最多 5% 的四舍五入（请联系 AWS Support 以启用）	在 CTP 中明确配置为匹配源码率
Non-integer 分段长度	默认 2 秒；不支持非整数段长度（请联系 AWS Support `calculateSegmentLength` 以启用自动整数段长度计算）	您必须手动配置分段长度以匹配起点
自定义 PID 值	不支持	支持
杜比视界音频	不支持	支持
自定义音频采样率	不支持	支持
Multi-origin 支持	自动	需要 Superset CTP；仅限 36 个视频变体
MediaTailor 改进	自动	需要手动 CTP 更新
最适合	标准编解码器匹配、常用比特率配置文件、trickplay、紧凑型 DASH	动态转码无法满足特殊的编码要求

预取广告

对直播使用 AWS Elemental MediaTailor 广告预取，以帮助减少广告决策服务器 (ADS) 的峰值负载，并减少每次广告时段开始时的清单投放延迟。在定义预取时间表时，请 MediaTailor 按照时间表从广告中检索广告，并在广告插播需要之前为广告插入做好准备。在直播期间，预取可以帮助缓解广告填充率下降以及由于广告请求和转码超时或其他渠道延迟而错失的获利机会。

> **Note**
>
> 广告预取不适用于服务器引导式广告插入 (SGAI) 方法，包括传统的服务器引导式广告插播和 HLS 插页式广告。SGAI 方法不需要预取，因为玩家只能获取他们要播放的广告，并且清单可以由 CDN 提供，而无需 MediaTailor看到单个会话请求。

要设置广告预取，您需要根据播放配置创建一个或多个预取计划。预取时间表告诉您 MediaTailor 如何以及何时检索和准备即将到来的广告时段的广告。

- 如果某个活动的广告投放时间表是可预测的，请使用单一的预取计划。每个预取计划都定义了一组 MediaTailor 要在单个广告中投放的广告。要在使用单个预提时间表时为多个广告预取广告，您必须创建与每个广告投放相关的多个预取计划（最多 24 小时前）。
- 如果某个活动的广告投放时间表不在可预测的时间表上，请使用定期预取计划。定期预取计划会自动创建时间表并在活动中每个广告中断之前预取广告。定期预取计划会在规定的时间段内（最多在活动结束前 24 小时）检索每个广告可用的广告。您无需为每个广告创建时间表，但是您确实会失去单次预取所提供的一些时间控制。

以下主题详细介绍了广告预取。

主题

- 预取的工作原理
- 创建预回迁计划
- TPS-based 流量整形
- 删除预回迁计划

预取的工作原理

当您的客户端向发出清单请求时 MediaTailor，该服务会评估与播放配置关联的所有预取计划。如果找 MediaTailor 不到匹配的预取时间表，该服务将恢复为正常的广告插入，并且不会预取广告。

如果 MediaTailor 找到匹配的预取计划，该服务将根据两个组成部分来评估该计划：检索和消费。每个组件的配置因单个预取计划和重复的预取计划而异，如以下各节所述。

单一预取计划流程

检索

这定义了检索窗口，即从 ADS 中 MediaTailor 预取广告的时间范围。请务必将此时段安排在广告中断之前的时间。下文概述了如何 MediaTailor 处理单个预回迁计划。

有关在控制台中创建单一预回迁计划的步骤，请参阅创建预回迁计划。有关 API 的说明，请参阅 AWS Elemental MediaTailor API 参考PrefetchSchedules中的。

在指定的检索时段内， MediaTailor 向 ADS 发送请求，要求其检索和准备广告，以便日后在播放会话中插入。

- MediaTailor （可选）使用流量整形来限制一次向 ADS 发出的请求数量。您可以在两种方法之间进行选择：

  Time-window 流量整形-将请求 MediaTailor 分布在指定的秒数内，而不是一次性发送所有会话的请求。这种分散的流量分布有助于防止广告不堪重负，从而导致超时和低广告填充率。

  TPS-based 流量整形-根据每秒交易量 (TPS) 和并发用户 MediaTailor 限制请求。此方法可根据您的 ADS 容量限制提供更直观的配置。有关更多信息，请参阅 TPS-based 流量整形。
- 如果您设置了动态变量，请在 MediaTailor 向 ADS 发出的请求中包含这些变量。 MediaTailor 使用这些变量将广告投放量与消费窗口期间的预提时间表进行匹配。有关更多信息，请参阅以下“消费” 部分。

Example

直播活动从上午 7:45 持续到上午 10 点，广告中断时间为上午 8:15。您配置 MediaTailor为从上午 7:45 到上午 8 点检索广告，流量调整窗口为 60 秒。如果有 500,000 个并发用户，则 MediaTailor 分发 ADS 请求以在 60 秒内实现每秒大约 8,333 笔交易的平均速率（500,000 users/60 秒=每秒 8,333 个请求），而不是同时发送所有请求。

检索配置包括动态变量键`scte.event`和值`1234`。 MediaTailor 在向 ADS 发出的请求中包含此变量，然后可使用该变量将特定广告商定位到事件 ID 1234。

消费

当在消费窗口期间 MediaTailor 遇到 SCTE-35 广告插播标记时，它会将预取的广告放置在广告插播时间中。

- 如果您未设置可用性匹配标准，则会在消费窗口的第一个空白处 MediaTailor 插入广告。
- 如果您确实为可用性匹配条件设置了动态变量键，则会根据您在检索窗口中设置的动态变量来 MediaTailor 评估这些条件。只有在满足广告匹配条件的情况下，广告插播才有资格进行预取广告插入。 MediaTailor 在符合条件的第一个广告时段插入广告。

  有关支持的可用性匹配条件的列表，请参阅上表中的 “可用于广告预取” 列。MediaTailor ADS 请求的会话变量

Example持续

您可以将消费的开始时间设置为上午 8:15，将结束时间设置为上午 8:17。您将密钥包含`scte.event_id`在可用性匹配标准中。

对于从上午 8:15 MediaTailor 到上午 8:17 出现的每个广告时段，它都会评估每个广告时段SCTE的事件 ID。在每个播放会话中， MediaTailor 将预先提取的广告插入事件 ID 为 1234（如检索动态变量中所定义）的第一个广告插播中。对于不包含正确事件 ID 的广告插播时间，将 MediaTailor执行标准广告插入。

重复的预取计划流程

检索

这定义了重复检索窗口，即为直播活动 MediaTailor 预取和插入广告的时间范围（最长 24 小时）。下文概述了如何 MediaTailor 处理重复的预取计划。

有关在控制台中创建重复预取计划的步骤，请参阅创建预回迁计划。有关 API 的说明，请参阅 AWS Elemental MediaTailor API 参考PrefetchSchedules中的。

在指定的周期性预取窗口内， MediaTailor 检索和插入最长 24 小时的直播活动的广告。在窗口中每次广告插播后， MediaTailor 都会自动检索下一个广告时段的广告。

- 如果您在有效期结束后设置延迟，则会 MediaTailor 等待指定的时间，然后再检索下一组广告以备下一个广告时段。
- MediaTailor （可选）使用流量整形来限制一次向 ADS 发出的请求数量。您可以在两种方法之间进行选择：

Time-window 流量整形-将请求 MediaTailor 分布在指定的秒数内，而不是一次性发送所有会话的请求。这种分散的流量分布有助于防止广告不堪重负，从而导致超时和低广告填充率。

TPS-based 流量整形-根据每秒交易量 (TPS) 和并发用户 MediaTailor 限制请求。此方法可根据您的 ADS 容量限制提供更直观的配置。有关更多信息，请参阅 TPS-based 流量整形。

- 如果您设置了动态变量，请在 MediaTailor 向 ADS 发出的请求中包含这些变量。 MediaTailor 使用这些变量将广告投放量与消费窗口期间的预提时间表进行匹配。有关更多信息，请参阅以下“消费” 部分。

Example

直播活动从晚上 7 点持续到晚上 8:45，在此期间有四次广告中断。广告中断的时间表不在可预见的时间表上。您可以将周期性预取配置为从晚上 7 点到晚上 8:45，延迟时间为 10 分钟，流量调整窗口为 60 秒。每次成功后，都会 MediaTailor 检索下一个广告时段的广告。有效期结束十分钟后， MediaTailor 开始向 ADS 发送检索请求。凭借 60 秒的流量整形窗口和 500,000 个并发用户， MediaTailor 分发 ADS 请求以在 60 秒内实现每秒大约 8,333 笔交易的平均速率（500,000 users/60 秒=每秒 8,333 个请求），而不是同时发送所有请求。

检索配置包括动态变量键`scte.event`和值`1234`。 MediaTailor 在向 ADS 发出的请求中包含此变量，然后可使用该变量将特定广告商定位到事件 ID 1234。

消费

当 MediaTailor 遇到 SCTE-35 广告插播标记时，它会将预取的广告放置在广告插播时间点中。

- 如果您设置了检索到的广告过期时间，则在指定的到期日之前，预取的广告可以被插入。
- 如果您未设置可用性匹配标准，则会在消费窗口的第一个空白处 MediaTailor 插入广告。
- 如果您确实为可用性匹配条件设置了动态变量键，则会根据您在检索窗口中设置的动态变量来 MediaTailor 评估这些条件。只有在满足广告匹配条件的情况下，广告插播才有资格进行预取广告插入。 MediaTailor 在符合条件的第一个广告时段插入广告。

  有关支持的可用性匹配条件的列表，请参阅上表中的 “可用于广告预取” 列。MediaTailor ADS 请求的会话变量

Example持续

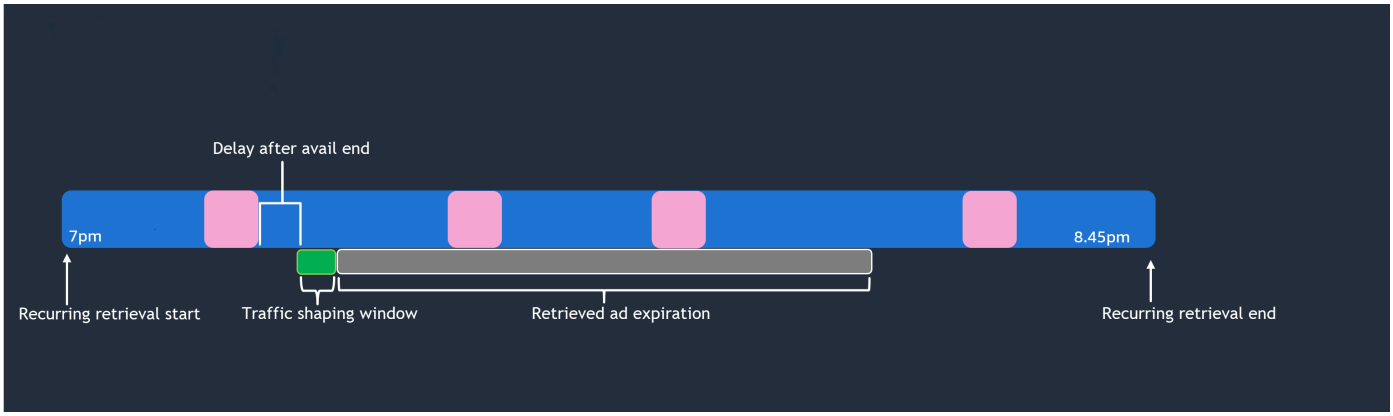
在消费中，您将密钥包含`scte.event_id`在可用性匹配标准中。

对于 MediaTailor 遇到的每个广告插播时间，它都会评估每个广告插播时间SCTE的事件 ID。在每个播放会话中， MediaTailor 将预先提取的广告插入事件 ID 为 1234（如检索动态变量中所定义）

的每个广告插播时间。对于不包含正确事件 ID 的广告插播时间，将 MediaTailor 执行标准广告插入。

您将广告过期时间设置为 2700 秒，因此检索到的广告在 45 分钟内可供插入。

下图说明了该示例，小方块代表广告插播时间。重复的预取计划设置在事件时间轴上进行了说明。



了解预取成本

提出广告检索请求不收取任何费用。但是，对于预取广告，您需要按照标准转码费率对预取的广告进行转码收费。 MediaTailor 对于预取广告的消费，对于在广告插播时间段中放 MediaTailor 置的预取广告，将按标准费率向您收取广告插播费用。有关转码和广告插入费用的信息，请参阅AWS Elemental MediaTailor 定价。

创建预回迁计划

以下过程说明如何使用 MediaTailor控制台创建预回迁时间表。有关使用 API 以编程方式创建和管理预取计划的信息，请参阅 MediaTailor API 参考PrefetchSchedules中的AWS Elemental MediaTailor 。

> **Note**
>
> 在中配置预取计划时 MediaTailor，了解不同类型的变量是如何处理的，这一点很重要。
>
> 可用匹配标准
>
> 如果要在计划中使用可用匹配条件，请务必先使用动态会话变量配置播放配置的 ADS 网址模板，否则可用性匹配条件将不会生效。有关使用动态会话变量的信息，请参阅 MediaTailor广告插入入门主题第 3 步：配置 ADS 请求 URL 和查询参数中的。

预取时间表中的玩家变量

创建预取计划时，不要在预取配置中将玩家变量定义为动态变量。取而代之的是，像往常一样在会话开始时传递玩家变量。 MediaTailor 如果这些变量映射在 ADS 模板网址中，则这些变量会自动包含在预取广告请求中。

使用控制台创建新的预回迁时间表

1. 打开 MediaTailor 控制台，网址为https://console.aws.amazon.com/mediatailor/。
2. 在导航窗格中，选择配置。选择要为其创建预回迁计划的播放配置。
3. 在 “预回迁计划” 选项卡上，选择 “添加预回迁计划”。
4. 在 “预回迁计划详细信息” 窗格下，执行以下操作：
   - 在 “名称” 中，输入预取计划的标识符，例如 m y-prefetch-schedule。
   - 对于直播 ID，可以选择输入唯一的 ID。如果您的来源包含多个播放流，则可以使用此 ID MediaTailor 来指示在特定直播中投放广告。例如，如果您的播放配置包含体育直播和电视节目直播，则可以使用直播 ID 创建预取时间表，以插入针对体育直播的广告。您可以在客户端的会话初始化或清单请求 MediaTailor 中将流 ID 值传递给。有关更多信息，请参阅以下示例。
     - 要进行服务器端跟踪，请在客户端对 MediaTailor 终端节点的GET HTTP请求中包含?aws.streamId查询参数和值。有关服务器端跟踪的一般信息，请参阅MediaTailor 服务器端广告跟踪和报告。向包含直播 ID 的 HLS 终端节点发出的清单请求如下所示，其中*myStreamId*是您的直播 ID 的名称：

```
GET <mediatailorURL>/v1/master/<hashed-account-id>/<origin-id>/<asset-id>?
aws.streamId=myStreamId
```

     - 要进行客户端跟踪，请在客户端的POST HTTP会话初始化请求正文中包含streamId键和值MediaTailor/v1/session。有关客户端跟踪的一般信息，请参阅Client-side 广告追踪。包含直播 ID 的会话初始化请求如下所示，其中*myStreamId*是您的直播 ID 的名称：

```
POST <mediatailorURL>/v1/session/<hashed-account-id>/<origin-id>/<asset-id>
{
    'streamId': 'myStreamId',
    'reportingMode': 'client'
}
```

5. 对于 Prefetch 类型，请进行选择并选择相应的部分，以获得有关其他字段的帮助：

- 如果您要为活动中的一个广告时段创建一个预取时间表，请选择 “单个”。
- 如果您要创建的时间表，以便在活动中的每个广告中断之前自动预取广告，请选择 “重复”。

单次预取时间表

创建一个时间表，在活动中投放一个广告之前预先获取广告。

1. 在 “检索” 窗格上，指定要使用的检索设置。这些设置决定何时从 ADS MediaTailor 中预取广告。它们还会决定在 ADS 请求中包含哪些动态会话变量（如果有）。

   - 在 “开始时间” 中，输入 MediaTailor 可以开始检索此广告时段的预取时间。 MediaTailor 将尝试针对您的客户在此时间或之后提出的清单请求预取广告。默认值为当前时间。如果您未指定值，则该服务会尽快开始预取检索。
   - 在 “结束时间” 中，输入您希望停止 MediaTailor 为此广告时段预取广告的时间。 MediaTailor 将尝试预取在此时间或之前提出的清单请求的广告。检索窗口可以与消费窗口重叠。
   - （可选）配置流量整形以限制一次向 ADS 发出的请求数量。选择以下方法之一：

     Time-window 方法：对于流量调整窗口持续时间，输入 MediaTailor 应将请求分发到 ADS 的秒数。有关更多信息，请参阅单一预回迁时间表检索说明。

     TPS-based 方法：将 Peak TPS 和峰值并发用户配置为根据每秒交易量和并发用户数限制请求。有关更多信息，请参阅 TPS-based 流量整形。
   - 在 “动态变量” 部分，输入多达 100 个动态会话变量。 MediaTailor 在发送给 ADS 的预取请求中使用这些变量进行替换。如果您未输入任何动态会话变量， MediaTailor 请尽力插值 AD S 网址中包含的动态变量的值。
     - 选择 “添加动态变量”。
     - 对于 Key，输入动态会话变量密钥，例如`scte.event_id`。您可以使用任何 MediaTailor 支持的动态变量。有关动态会话变量的信息，请参见MediaTailor ADS 请求的会话变量。
     - 在 “值” 中，输入动态变量值，例如*`my-event`*。
     - 要添加其他动态变量，请选择选择添加动态变量。
2. 在 “消费” 窗格上，指定要用于消费窗口的设置。这些设置决定了何时将广告放 MediaTailor 置在广告插播中。它们还会确定您要使用的任何可用性匹配标准。

   - 在 “开始时间” 中，输入您 MediaTailor 想要开始将预取广告放入广告插播时间的时间。默认值为当前时间。如果您未指定时间，则该服务会尽快开始使用预取数据。

- 在“结束时间”中，输入您希望 MediaTailor 停止将预取的广告放入广告插播时间点的时间。MediaTailor 将尝试针对客户在此时间或之前提出的清单请求预取广告。结束时间必须晚于开始时间，且距离现在不到一天。消费窗口可以与检索窗口重叠。
- 在“可用性匹配条件”部分中，选择“添加可用性标准”，然后将最多五个可用性匹配条件添加到您的日程安排中。然后，在动态变量键下，添加动态变量键，例如`scte.event_id`。MediaTailor 只有当预取的广告符合由您的客户传递给的动态变量值或从会话数据等信息中推断出的动态变量值定义的条件时 MediaTailor，才会将其放置在广告插 MediaTailor 播中。如果广告插播时间不符合指定的匹配条件，则会 MediaTailor 跳过该广告插播时间的预取。有关信息，请参阅单次预取计划消耗说明。

  > **Important**
  >
  > 每个预取计划最多可以指定五个可用匹配标准。如果指定多个条件，则使用逻辑 AND MediaTailor 组合它们。要对逻辑 OR 组合进行建模，请创建多个预取计划。
  > 仅使用每次广告时段发生变化的变量（例如`scte.event_id`、`scte.segmentation_upid`、或`avail.index`）作为可用性匹配标准。请勿使用玩家变量（例如`player_params.*`）作为可用性匹配标准，因为玩家变量在整个会话中保持不变，并且不会区分广告时段。 MediaTailor 如果玩家变量映射到广告模板网址中，则会在预取广告请求中自动包含这些变量。

3. 选择“添加可用性标准”。

预取计划在消费窗口结束时间后自动过期。出于诊断目的，它们至少会在 7 天内保持可见，之后它们 MediaTailor 会自动删除。或者，您可以随时手动删除预回迁时间表。有关如何手动删除预回迁时间表的信息，请参阅下一the section called “删除预回迁计划”节。

确定您的客户应多久调用 CreatePrefetchSchedule API 的频率

如果您确切知道广告中断的发生时间，则您的客户可以每天以编程方式调用一次 CreatePrefetchScheduleAPI 来设置检索和消费。或者，您的客户可以在一天中多次调用 API 来定义检索和使用。选择 API 调用频率时，请考虑有效预取计划的最大数量，以及在您创建预取计划后您的广告中断时间表是否会发生变化的可能性。如果在您创建了预提计划后，广告中断时间表可能会发生变化，那么您可能需要更频繁地调用 API。

定期预取计划

创建时间表，在活动中每个广告投放之前预先获取广告。

1. 在重复检索窗格上，指定要使用的检索设置。这些设置决定何时从 ADS MediaTailor 中预取广告。它们还会决定在 ADS 请求中包含哪些动态会话变量（如果有）。

   - 在“重复预取”窗口中，输入此广告时段 MediaTailor 可以开始预取回的时间。 MediaTailor 将尝试针对您的客户在此时间或之后提出的清单请求预取广告。默认值为当前时间。如果您未指定值，则该服务会尽快开始预取检索。
   - 对于可用性结束后的延迟，请输入在可用性结束后等待的秒数，然后再预取下一个可用性的广告。 MediaTailor如果未指定值，则 MediaTailor 默认为无延迟。
   - （可选）配置流量整形以限制一次向 ADS 发出的请求数量。选择以下方法之一：

     Time-window 方法：对于流量调整窗口持续时间，输入 MediaTailor 应将请求分发到 ADS 的秒数。有关更多信息，请参阅重复预取计划检索说明

     TPS-based 方法：将 Peak TPS 和峰值并发用户配置为根据每秒交易量和并发用户数限制请求。有关更多信息，请参阅 TPS-based 流量整形。
   - 在“动态变量”部分，输入多达 100 个动态会话变量。 MediaTailor 在发送给 ADS 的预取请求中使用这些变量进行替换。如果您未输入任何动态会话变量， MediaTailor 请尽力插值 AD S 网址中包含的动态变量的值。
     - 选择“添加动态变量”。
     - 对于 Key，输入动态会话变量密钥，例如`scte.event_id`。您可以使用任何 MediaTailor 支持的动态变量。有关动态会话变量的信息，请参见MediaTailor ADS 请求的会话变量。
     - 在“值”中，输入动态变量值，例如*`my-event`*。
     - 要添加其他动态变量，请选择选择添加动态变量。

2. 在“消费”窗格上，指定要用于消费窗口的设置。这些设置决定了何时将广告放 MediaTailor 置在广告插播中。它们还会确定您要使用的任何可用性匹配标准。

   - 对于检索到的广告过期，请指明检索广告在多长时间内可供插入。
   - 在“可用性匹配条件”部分中，选择“添加可用性标准”，然后将最多五个可用性匹配条件添加到您的日程安排中。然后，在动态变量键下，添加动态变量键，例如`scte.event_id`。MediaTailor 只有当预取的广告符合由您的客户传递给的动态变量值或从会话数据等信息中推断出的动态变量值定义的条件时 MediaTailor，才会将其放置在广告插 MediaTailor 播中。如果广告插播时间不符合指定的匹配条件，则会 MediaTailor 跳过该广告插播时间的预取。有关信息，请参阅定期预取计划消耗说明。

> **Important**
>
> 每个预取计划最多可以指定五个可用匹配标准。如果指定多个条件，则使用逻辑 AND MediaTailor 组合它们。要对逻辑 OR 组合进行建模，请创建多个预取计划。
> 仅使用每次广告时段发生变化的变量（例如`scte.event_id`、`scte.segmentation_upid`、或`avail.index`）作为可用性匹配标准。请勿使用玩家变量（例如`player_params.*`）作为可用性匹配标准，因为玩家变量在整个会话中保持不变，并且不会区分广告时段。 MediaTailor 如果玩家变量映射到广告模板网址中，则会在预取广告请求中自动包含这些变量。

3. 选择 “添加可用性标准”。

预取计划在消费窗口结束时间后自动过期。出于诊断目的，它们至少会在 7 天内保持可见，之后它们 MediaTailor 会自动删除。或者，您可以随时手动删除预回迁时间表。有关如何手动删除预回迁时间表的信息，请参阅下一the section called “删除预回迁计划”节。

TPS-based 流量整形

AWS Elemental MediaTailor 提供了两种可选的流量整形方法，用于限制一次向 ADS 发出的请求数量。 TPS-based 流量整形为预取计划提供了基于时间窗的流量整形的替代方案。这种方法允许您根据每秒交易量 (TPS) 和预期并发用户而不是时间计算来指定广告决策服务器 (ADS) 容量，从而提供更直观的配置。

TPS-based 流量整形的工作原理

您无需指定检索窗口的持续时间，而是提供以下参数：

峰值 TPS

您的 ADS 每秒可以处理的最大请求数。此参数没有默认值。

并发用户峰值

您的内容的预期并发观看人数峰值。此参数没有默认值。

MediaTailor 无论并发会话数量多少，都会自动将预回迁请求分发到指定的 TPS 限制之内。

Example TPS-based 配置示例

你的广告可以处理 500 TPS，你预计在高峰时段会有 100,000 名并发观众。您可以配置：

- 峰值 TPS：500
- 并发用户峰值：100,000

MediaTailor 无论并发会话数量多少，都会自动将预回迁请求分发到指定的 TPS 限制之内。

删除预回迁计划

以下过程说明如何使用 MediaTailor控制台删除预回迁时间表。有关如何使用 API 以编程方式删除预取计划的信息，请参阅 MediaTailor API 参考DeletePrefetchSchedule中的AWS Elemental MediaTailor 。

> **Note**
>
> 删除不是实时发生的。在 MediaTailor 删除预回迁时间表时，您可能会遇到延迟，在此期间，预回迁的检索和使用将继续在后台运行。

使用控制台删除预回迁时间表

1. 打开 MediaTailor 控制台，网址为https://console.aws.amazon.com/mediatailor/。
2. 在导航窗格中，选择配置。选择包含要删除的预回迁计划的播放配置。
3. 在 “预回迁计划” 选项卡上，选择要删除的预回迁时间表。然后选择删除。

将预处理广告与 AWS Elemental MediaTailor

在典型的广告插入工作流程中，对广告进行 MediaTailor 动态转码以匹配内容流，将其保存，然后将广告拼接到直播中。由于此过程仅在 MediaTailor 收到来自广告决策服务器 (ADS) 的 VAST 响应中的广告后才会发生，因此广告可供拼接的时间会有所延迟。如果在广告拼接工作流程中引入额外的延迟（可能是由于广告超时或其他内容或网络问题），则 MediaTailor 可能会部分填补空白，或者完全错过广告时段。

为了缩短将广告拼接到内容中的时间，您可以使用预处理广告。预处理广告是指您在广告插入中使用之前对其进行转码的 MediaTailor 广告。与其在广告中 URLs 提供无条件广告，不 URLs 如为预处理广告提供。在对 MediaTailor 请求的VAST回应中，ADS包含了指向预处理广告的直接链接。通过删除广告拼接中的转码部分， MediaTailor 只需保存广告并将其拼接到内容流中即可。使用预处理广告的广告拼接过程缩短了从通过 VAST 响应得 MediaTailor 知广告到将广告拼接到内容中的时间。

或者，您也可以使用广告预取，即您配置为在 MediaTailor 需要广告插播之前的预定时间执行广告拼接流程。有关广告预取的更多信息，请参阅。预取广告

预处理广告要求

以下是使用预处理广告设置广告拼接工作流程时需要考虑的要求。

MediaFiles要求

广告服务器发送到的 VAST 响应 MediaTailor 必须包含MediaFiles满足以下要求的响应：

广告 (Creative) 必须具有符合内容流比特率变体的变体。 您有责任确保 VAST 响应使用正确的广告变体来匹配模板清单。

虽然使用预处理广告可以帮助提高广告插入的效率，但 MediaTailor 无法管理转码过程来确保广告的媒体文件与内容清单规范兼容。如果广告与内容流不匹配，则 MediaTailor 可能会错过插入，或者不匹配可能会导致播放设备出错。

此外，要在不进行 MediaTailor 转码的情况下拼接到内容流中，MediaFile必须满足以下要求：

- 它必须可以在公共互联网上访问，这样 MediaTailor 才能下载它。
- 它必须使用流式传输，如VAST响应delivery="streaming"中所示。
- 它必须是.m3u8（对于 HLS）或.mpd（对于 DASH）文件。

Example反应迅速

在以下 VAST 响应示例中， MediaTailor 插入MediaFile带有以下内容的 URLs：

- 对于 HLS 直播， MediaTailor 使用https://example-ad-origin.amazonaws.com/ad1/index_low.m3u8。这是第一款MediaFile采用流媒体传输和支持的文件扩展名的 (. m3u8)。
- 对于 DASH 直播， MediaTailor 使用https://example-ad-origin.amazonaws.com/ad1/index.mpd。这是第一款MediaFile采用流媒体传输和支持的文件扩展名的 (. 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.mp4]]></MediaFile>
                            <MediaFile delivery="streaming" width="1280"
 height="720" type="application/dash+xml" bitrate="533" scalable="true"
 maintainAspectRatio="true"><![CDATA[https://example-ad-origin.amazonaws.com/ad1/
index.mpd]]></MediaFile>
                            <MediaFile delivery="streaming" width="640"
 height="360" type="application/x-mpegURL" bitrate="262" scalable="true"
 maintainAspectRatio="true"><![CDATA[https://example-ad-origin.amazonaws.com/ad1/
index_low.m3u8]]></MediaFile>
                            <MediaFile delivery="streaming" width="2560"
 height="1440" type="application/x-mpegURL" bitrate="1066" scalable="true"
 maintainAspectRatio="true"><![CDATA[https://example-ad-origin.amazonaws.com/ad1/
index_high.m3u8]]></MediaFile>
                        </MediaFiles>
                    </Linear>
                </Creative>
            </Creatives>
        </InLine>
    </Ad>
</VAST>
```

广告清单要求

要使用预处理广告，您的父广告和子广告清单必须满足以下要求：

- 在 VAST 响应`Creative`部分中链接的清单必须是父广告清单。
- f URLs or the child 的广告清单必须是相对路径。
- 子广告清单必须与父多变体播放列表位于同一目录中，且级别相同。子清单不能位于子目录或其他位置。

Example支持的父多变体播放列表

以下父广告多变体播放列表包含子广告媒体 URLs 的相关播放列表。子播放列表也与父多变体播放列表位于同一目录中。

```
#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不支持的父级多变体播放列表：子目录

以下父级广告多变体播放列表包含子播放列表，这些子播放列表位于相对于父多变体播放列表的子目录中。它不支持预处理广告的播放列表。

```
#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不支持的父级多变体播放列表：绝对 URLs

以下父广告多变体播放列表包含绝对值 URLs的子播放列表。它不支持预处理广告的播放列表。

```
#EXTM3U
#EXT-X-STREAM-INF:BANDWIDTH=150000,RESOLUTION=416x234,CODECS="avc1.42e00a,mp4a.40.2"
https://example.mediatailor.us-west-2.amazonaws.com/index_1.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=440000,RESOLUTION=416x234,CODECS="avc1.42e00a,mp4a.40.2"
https://example.mediatailor.us-west-2.amazonaws.com/index_2.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=640000,RESOLUTION=640x360,CODECS="avc1.42e00a,mp4a.40.2"
https://example.mediatailor.us-west-2.amazonaws.com/index_3.m3u8
```

预处理广告工作流程

以下是对预处理广告在广告拼接工作流程中的运作方式的基本描述。 MediaTailor工作流程的第一部分是为使用预处理广告进行设置所必须采取的操作。第二部分描述了如何 MediaTailor 处理广告。

第 1 部分：预处理广告的设置

完成以下步骤，设置在中 MediaTailor使用预处理广告的工作流程。

1. 使用转码器服务（例如）将您的广告素材调整为支持模板清单中不同比特率、分辨率和编解码器的变体。 AWS Elemental MediaConvert
2. 将预先转码 URLs 的媒体文件提供给 ADS，以便在 VAST 响应中使用。
3. 在中@@ 创建您的播放配置 MediaTailor。要使用预处理广告，请在配置中的流媒体文件调节设置中选择 “无”。
4. 像往常一样继续设置内容分发。

第 2 部分： MediaTailor 广告处理

MediaTailor 广告拼接按中所MediaTailor 广告插入的工作原理述完成。当 MediaTailor 收到来自 ADS 的 VAST 回复时，它会使用以下逻辑来确定要对广告采取哪些行动。此逻辑由播放配置上的流媒体文件调节设置决定。

- 当 “流媒体文件调节” 设置为 “转码” 时，在`progressive`交付时对媒体文件进行 MediaTailor 转码，然后将其拼接到清单中。如果没有足够的带有`progressive`投放媒体文件的广告来填补空缺，请对这些广告进行 MediaTailor 转码并使用带有`streaming`投放效果的广告。
- 当 “流媒体文件调节” 设置为 “无” 时，将包含 MediaTailor `streaming`投放媒体文件的广告拼接到清单中，而无需对其进行转码。如果没有足够的带有`streaming`投放媒体文件的广告来填补空缺，请对这些广告进行 MediaTailor 转码并使用带有`progressive`投放效果的广告。

使用 MediaTailor 变现功能

借助 MediaTailor 获利函数（函数），您可以自定义广告插入期间如何 AWS Elemental MediaTailor 管理会话数据和生成广告请求。函数允许您调用外部 API、使用表达式转换数据，以及修改广告决策服务器 (ADS) 请求参数或玩家参数。您无需部署或管理自定义基础架构。

函数使用 jsonata（一种用于 JSON 数据的轻量级查询和转换语言）来评估表达式。您可以编写 JSonata 表达式来读取会话数据、转换值和定义输出。有关支持的 JSonata 函数的完整列表，请参阅。jsonata 表达式参考

在需要时使用函数：

- 使用观看者数据丰富广告请求。在会话开始时致电身份服务，并在每个 ADS 个性化广告定位请求中包含已解析的身份。

- 自定义 ADS 请求参数。根据会话数据、 SCTE-35 信号或外部 API 响应动态设置 ADS 网址、标题或正文。
- 在广告服务器上运行 A/B 测试。根据会话属性或随机分配在不同的 ADS 端点之间分配流量。
- 构建多步扩充管道。将多个函数链接在一起，从一个 API 中获取数据，对其进行转换，然后将结果传递到下一步。

函数如何融入工作 MediaTailor 流程

当观看者开始播放会话时，会在广告插入流程中的特定时刻 MediaTailor 评估您的功能，称为生命周期挂钩。

1. 您可以定义一个函数，并通过函数映射将其附加到播放配置。
2. MediaTailor 在指定的生命周期挂钩上运行该函数。
3. 该函数读取会话数据，可以选择调用外部 API，并写入在播放处理期间 MediaTailor 使用的结果。

> **Tip**
>
> 如果您不熟悉 Functions，请从开始快速入门指南，然后阅读生命周期钩子。有关表达式语法和完整示例，请参阅jsonata 表达式参考和函数示例。

主题

- 函数快速入门指南
- 函数生命周期挂钩
- 函数类型和构成
- 创建和管理函数
- 函数的 jsonata 表达式参考
- 函数示例
- 故障排除和监控功能
- 功能限制

函数快速入门指南

本指南将引导您创建第一个函数、将其附加到播放配置以及验证它是否已运行。最后，您将拥有一个工作函数，该函数可以对每个观看者的设备类型（ctvmobile、或desktop）进行分类，并将其存储为每个广告请求中可用的播放器参数。

先决条件

在开始之前，请确保已有 MediaTailor 播放配置。如果没有，请参阅入门 MediaTailor。

步骤 1：创建 函数

在此步骤中，您将创建一个函数，该函数根据用户代理字符串对查看者的设备类型进行分类，并将结果存储在玩家参数中。该函数使用带有jsonata 表达式参考表达式的自定义输出类型（不调用外部 API）来评估用户代理。

1. 打开 MediaTailor 控制台，网址为https://console.aws.amazon.com/mediatailor/。
2. 在导航窗格中，选择函数。
3. 选择创建函数。
4. 在创建向导模式中，选择从头开始创建，然后选择继续。
5. 在 “函数类型” 下，选择 “自定义输出” 图块。
6. 在函数详细信息下，输入以下内容：
   - 函数 ID：`myFirstFunction`
   - 描述：`Classify device type from user agent`
7. 在 “自定义输出配置” 下的 “输出” 部分中，添加一行：
   - 键：`player_params.deviceType`
   - 值：`{% $contains(session.user_agent, 'CTV') ? 'ctv' : $contains(session.user_agent, 'Mobile') ? 'mobile' : 'desktop' %}`
8. 选择创建函数。

成功通知确认函数已创建，您将被重定向到函数详细信息页面。

由此产生的函数配置为：

```
{
    "FunctionId": "myFirstFunction",
    "FunctionType": "CUSTOM_OUTPUT",
```

```
    "Description": "Classify device type from user agent",
    "CustomOutputConfiguration": {
        "Runtime": "JSONATA",
        "Output": {
            "player_params.deviceType": "{% $contains(session.user_agent, 'CTV') ?
 'ctv' : $contains(session.user_agent, 'Mobile') ? 'mobile' : 'desktop' %}"
        }
    }
}
```

第 2 步：将该功能附加到播放配置

将该函数映射到播放配置上的生命周期挂钩。映射告诉 MediaTailor 何时运行该函数。

1. 在导航窗格中，选择配置。
2. 选择要更新的播放配置。
3. 选择编辑。
4. 展开 “函数配置” 部分。
5. 对于会话初始化挂钩，请`myFirstFunction`从下拉列表中选择。
6. 选择保存。

这会附加`myFirstFunction`到Pre-session 初始化生命周期挂钩上。生成的函数映射为：

```
{
    "FunctionMapping": {
        "PRE_SESSION_INITIALIZATION": "myFirstFunction"
    }
}
```

MediaTailor 在此播放配置下，在每个新会话开始时运行一次该函数。

步骤 3：启动会话并验证函数已运行

启动新的播放会话以触发该功能。向播放配置的会话初始化端点使用会话初始化请求。

MediaTailor 自动发布每次函数执行的 CloudWatch 指标，无需选择加入。启动会话后，检查`AWS/MediaTailor`命名空间中的以下指标以确认您的函数已运行：

- `PreSessionInitHook.Invocations`— 确认钩子已开火。

- PreSessionInitHook.Errors— 如果函数成功，则应为 0。
- Function.Invocations— 确认已执行的单个函数。此指标包括FunctionIdFunctionType、和HookType维度，因此您可以进行myFirstFunction具体筛选。

如果该函数失败，则默认情况下会 MediaTailor 向 Manifest Logs 发出错误日志事件（无需配置）：

- PRE_SESSION_INIT_HOOK_ERROR— 使用errorType和时 Hook-level 失败cause。
- PRE_SESSION_INIT_FUNCTION_ERROR— Function-level 失败，具体functionId和错误细节。

以下示例显示了函数表达式中语法错误PRE_SESSION_INIT_FUNCTION_ERROR的事件：

```
{
    "eventTimestamp": "2024-01-01T12:00:00.076000000Z",
    "eventType": "PRE_SESSION_INIT_FUNCTION_ERROR",
    "eventDescription": "Function execution failed",
    "awsAccountId": "123456789012",
    "originId": "my-config",
    "sessionId": "session-123",
    "requestId": "req-abc",
    "eventId": "5dc6f040-0f72-4e8c-a64e-25eeef62708c",
    "functionId": "myFirstFunction",
    "functionType": "CUSTOM_OUTPUT",
    "executionTimeMs": 2,
    "errorType": "SYNTAX_ERROR",
    "cause": "Expected \")\" before end of expression",
    "input": {}
}
```

使用该eventId字段将钩子和函数错误事件关联到同一执行中。该errorType字段告诉你失败的类别— 故障排除和监控 有关错误类型和修复方法的完整列表，请参阅。

> **Note**
>
> 如需详细的成功日志记录，请在 Manifest Log 配置中选择加入PRE_SESSION_INIT_HOOK_SUMMARY和PRE_SESSION_INIT_FUNCTION_COMPLETED事件。摘要事件显示每次执行的挂钩结果。已完成的事件显示每个函数的输入、输出和 HTTP request/response 详细信息。默认情况下，它们处于禁用状态，以最大限度地降低日志成本。有关更多信息，请参阅 故障排除和监控。

幕后会发生什么

以下是您刚刚创建的函数的完整请求流程：

1. 玩家与发起会话。 MediaTailor
2. MediaTailor 触发PRE_SESSION_INITIALIZATION生命周期挂钩并运行myFirstFunction。
3. 该函数对session.user_agent字段进行求值并写入ctvmobile、或desktop。player_params.deviceType
4. MediaTailor 创建会话并将清单返回给玩家。
5. 玩家在播放过程中遇到广告中断。
6. MediaTailor 触发PRE_ADS_REQUEST生命周期挂钩，然后构造 ADS 请求。由于存储deviceType在玩家参数中，因此可以通过动态变量替换将其包含在 ADS 请求网址中。
7. ADS 使用设备类型来返回定向广告创意。
8. MediaTailor 将广告拼接到清单中，然后将其返回给玩家。

如果函数因任何原因失败，则 MediaTailor 丢弃输出并像未附加任何函数一样继续运行。观看者仍然可以看到广告，只是没有设备类型定位。

建议的主题

现在，播放配置中附带了一个可以正常运行的功能。从这里：

- 要了解每个生命周期挂钩中都有哪些输入字段和输出命名空间可用，请参阅。生命周期钩子
- 要了解不同的函数类型以及如何将它们链接在一起，请参阅函数类型和构成。
- 要查看完整的工作示例，请参阅函数示例。

函数生命周期挂钩

生命周期挂钩定义了在播放期间何时 MediaTailor 运行您的函数。此页面是输入字段、输出命名空间以及管理每个挂钩数据流的规则的完整参考。

概述

MediaTailor 支持两个生命周期挂钩：

- **PRE_SESSION_INITIALIZATION**当查看者启动新会话时触发一次。将其用于一次性设置工作，例如获取受众细分。此时，还没有出现广告插播时间，因此广告插播背景不可用。

- **PRE_ADS_REQUEST**在每个广告决策服务器 (ADS) 请求之前触发——直播中每个广告插播时段触发一次。使用它来使用定位数据自定义 ADS 请求、修改 ADS 网址或添加标题。

关键区别在于时机：PRE_SESSION_INITIALIZATION运行一次即可设置在整个会话中持续保存的数据，同时重复PRE_ADS_REQUEST运行，可以根据特定的广告时段量身定制每个 ADS 请求。

Pre-session 初始化

当它着火的时候

MediaTailor 在新播放会话开始时运行映射到PRE_SESSION_INITIALIZATION的函数。该函数在 MediaTailor 构造初始清单响应之前运行。

Input

session.*、player_params.* 和 event.*。有关所有可用字段，请参阅输入字段参考。

允许输出命名空间

命名空间	可接受的类型
player_params.*	字符串、数字、布尔值

写入的player_params.*值将保留到会话中。它们可用：

- 作为PRE_ADS_REQUEST生命周期挂钩处的输入 player_params.*
- 在 ADS 中，通过以下方式请求网址MediaTailor ADS 请求的动态广告变量（例如，[player_params.deviceType]）
- 在会话的整个生命周期内，涵盖所有广告时段

> **Note**
>
> 所有player_params输出键和值的序列化总大小不得超过 1,000 个字符。如果总数超过此限制，则函数输出将被丢弃。有关更多信息，请参阅 功能限制。

典型用例

- 从外部服务获取身份或受众数据，并将其存储在玩家参数中，以便在以后的 ADS 请求中使用。
- 根据用户代理对设备类型进行分类，并将分类写入玩家参数。
- 设置下游广告插播处理所依赖的默认播放器参数值。
- 通过将值存储在 ADS 网址中包含的玩家参数中MediaTailor ADS 请求的动态广告变量。

失败行为

如果附加到的函数因任何原因PRE_SESSION_INITIALIZATION失败，则 MediaTailor 丢弃该函数的输出，并像没有附加函数一样继续操作。会话在没有函数的播放器参数值的情况下正常启动。

Pre-ads 请求

当它着火的时候

MediaTailor 在向 ADS 发送请求之前，立即运行映射到每个广告插播PRE_ADS_REQUEST一次的函数。在清单处理过程中，每次遇到广告机会时，该函数都会运行。

Input

来自PRE_SESSION_INITIALIZATION、加avail.*、、和adsRequest.*（网址scte.*asset.*、方法、标题、正文）的所有字段。有关所有可用字段，请参阅输入字段参考。

允许输出命名空间

命名空间	可接受的类型	如何使用输出
player_params.*	字符串、数字、布尔值	替换此广告插播时段的会话播放器参数。可通过以下方式访问广告请求网址MediaTailor ADS 请求的动态广告变量。
session.*	字符串、数字、布尔值	覆盖此广告插播时间段的会话变量。通过动态变量替换可用于 ADS 请求网址。

命名空间	可接受的类型	如何使用输出
avail.*	字符串、数字、布尔值	替换此广告插播时间段的可用变量。通过动态变量替换可用于 ADS 请求网址。
scte.*	字符串、数字、布尔值	替换此广告插播时间的 SCTE 变量。通过动态变量替换可用于 ADS 请求网址。
asset.*	字符串、数字、布尔值	替换此广告插播时间段的素材元数据变量（来自EXT-X-ASSET 标签）。通过动态变量替换可用于 ADS 请求网址。有关 HLS 广告标记的更多信息，请参阅 HLS 广告标记。
adsRequest.*	字符串	仅覆盖针对此广告插播的 ADS 请求。支持的字段：url、method、headers.<name> 、body。该url值被视为模板，并在函数运行MediaTailor ADS 请求的动态广告变量后支持。没有坚持下去。

> **Note**
>
> PRE_ADS_REQUEST挂接的所有输出都是临时替代——它们仅适用于当前广告插播的 ADS 请求，不会保留到会话中。

示例 — 重写 ADS 请求：

```
{
    "Output": {
        "adsRequest.url": "{%'https://ads.example.com/v1/vast?sid=' & session.id &
 '&genre=' & player_params.genre%}",
```

```
        "adsRequest.headers.X-Custom-Token": "{%player_params.auth_token%}"
    }
}
```

这取代了广告网址，并为当前的广告插播时间添加了自定义标题。

典型用例

- 重写 ADS 请求网址，在不同的广告服务器之间路由流量以进行 A/B 测试。
- 将丰富数据（受众细分、身份令牌）附加到 ADS 请求网址或标头。
- 根据 SCTE-35 信号数据或可用索引有条件地修改 ADS 请求参数。

失败行为

如果附加到的函数因任何原因PRE_ADS_REQUEST失败，则 MediaTailor 丢弃该函数的输出，并像没有附加函数一样继续操作。ADS 请求使用原始会话和请求参数发送，未经修改。

输入字段参考

字段	Type	会话前初始化	PRE_ADS_REQUEST
session.id	长整型	✓	✓
session.uuid	字符串	✓	✓
session.client_ip	字符串	✓	✓
session.user_agent	字符串	✓	✓
session.referer *	字符串	✓	✓
session.avail_duration_secs	长整型	✗	✓

字段	Type	会话前初始化	PRE_ADS_REQUEST
session.avail_duration_ms	长整型	✗	✓
player_params.*	字符串	✓	✓
event.id	字符串	✓	✓
event.hook	字符串	✓	✓
event.timestamp	字符串	✓	✓
avail.index	Int	✗	✓
avail.random	长整型	✗	✓
avail.source_content_time_epoch_ms	长整型	✗	✓
scte.event_id	Int	✗	✓
scte.avail_num	Int	✗	✓
scte.segmentation_event_id	Int	✗	✓
scte.segmentation_type_id	Int	✗	✓
scte.segmentation_upid	字符串	✗	✓

字段	Type	会话前初始化	PRE_ADS_REQUEST
scte.segmentation_upid.assetId	字符串	✗	✓
scte.segmentation_upid.cueData.key	字符串	✗	✓
scte.segmentation_upid.cueData.value	字符串	✗	✓
scte.unique_program_id	Int	✗	✓
scte.archive_allowed_flag	布尔值	✗	✓
scte.delivery_not_restricted_flag	布尔值	✗	✓
scte.device_restrictions	Int	✗	✓
scte.no_regional_blackout_flag	布尔值	✗	✓
scte.segment_num	Int	✗	✓

字段	Type	会话前初始化	PRE_ADS_REQUEST
scte.segments_expected	Int	✗	✓
scte.sub_segment_num	Int	✗	✓
scte.sub_segments_expected	Int	✗	✓
scte.avails_expected	长整型	✗	✓
asset.*	字符串	✗	✓
adsRequest.url	字符串	✗	✓
adsRequest.method	字符串	✗	✓
adsRequest.headers.<key>	字符串	✗	✓
adsRequest.body	字符串	✗	✓

* session.referer 仅当会话初始化请求中包含 Referer 标头时才会出现。用于$exists(session.referer)在访问之前进行检查。

函数类型和构成

AWS Elemental MediaTailor 提供四种功能类型，每种类型专为不同的工作类别而设计。您在创建函数时选择一种类型，该类型决定了该函数在运行时可以做什么。本页说明每种类型如何执行、何时使用以及如何将函数组合成多步流水线。

函数类型概述

Type	类别	用途
CUSTOM_OUTPUT	数据转换	根据当前会话状态评估表达式并生成输出。没有外部通话。
HTTP_REQUEST	外部整合	对外部服务进行 HTTP 调用，然后评估可以引用响应的输出表达式。
SEQUENTIAL_EXECUTOR	编排	按顺序运行一系列函数，通过临时数据在步骤之间传递数据。
CONCURRENT_EXECUTOR	编排	并行运行一组函数，直至最大并发度，然后合并它们的输出。

每种类型都有不同的作用。创建函数时选择一种类型——该类型决定了该函数在运行时可以做什么。

自定义输出

何时使用

CUSTOM_OUTPUT当你的函数逻辑纯粹是计算时使用。常见用例包括根据用户代理字符串对设备进行分类、使用 A/B 测试逻辑选择广告决策服务器网址，以及在玩家参数到达广告决策服务器之前对其进行重新格式化。

CUSTOM_OUTPUT函数是最快的类型，因为它们无需网络调用即可计算内存中的表达式。只要您无需联系外部服务即可实现目标，请选择此类型。

配置字段

一个CUSTOM_OUTPUT函数有两个字段：

- 运行时 -表达式语言。将此设置为JSONATA。
- 输出 — 键值对的映射。每个键都是一个输出绑定名称（例如player_params.device_type），每个值都是在运行时 MediaTailor 计算的表达式。

> **Note**
>
> 用于{%...%}将 Jsonata 表达式包装在字符串字段中。没有这些分隔符的值被视为静态字符串。例如，"{%session.id%}"计算表达式，而"session.id"是文字文本。

运行时行为

当 MediaTailor 执行一个CUSTOM_OUTPUT函数时，它会根据当前会话状态评估输出块中的每个表达式。所有表达式都看到相同的输入快照。评估结果成为函数的输出。

如果任何表达式无法计算，则 MediaTailor 丢弃整个函数的输出，并像没有附加函数一样继续操作。该函数不会部分提交结果。

示例： A/B 流量分割

以下函数将每个广告插播时间随机分配给两个广告决策服务器网址之一。它是为PRE_ADS_REQUEST生命周期挂钩设计的。

```
{
    "FunctionId": "abTestAdsUrl",
    "FunctionType": "CUSTOM_OUTPUT",
    "CustomOutputConfiguration": {
        "Runtime": "JSONATA",
        "Output": {
            "adsRequest.url": "{%$random() < 0.5 ? 'https://ads.example.com/v1' :
 'https://ads.example.com/v2'%}"
        }
    }
}
```

有关此示例的完整演练，请参阅函数示例。

HTTP 请求

何时使用

在函数需要调用外部服务HTTP_REQUEST时使用。常见用例包括从解析提供商获取身份数据、从数据管理平台检索受众细分以及将会话信息发送到日志端点。

配置字段

HTTP_REQUEST函数具有以下字段：

- 运行时 -表达式语言。将此设置为JSONATA。
- MethodType— HTTP 方法。支持的值为 GET 和 POST。
- 网址 — 发送请求的目标网址。您可以使用静态网址或动态构建 URL 的 JSonata 表达式。
- 标头 -请求中包含的 HTTP 标头，指定为标头名称和值对。对动态标头值使用{%...%}表达式语法。静态值可以直接指定为字符串。
- 正文 -要发送的请求正文。与POST请求一起使用。您可以使用 JSonata 表达式来动态构建身体。
- RequestTimeoutMilliseconds（必填）— 等待回复的时间。
- 输出 -定义 HTTP 调用完成后生成的值。每个条目将输出键（例如player_params.envelope_id）映射到可以引用该response对象的表达式。

有关适用于这些字段的大小限制和限制，请参阅限制。

请求的处理方式

MediaTailor 分两步处理HTTP_REQUEST函数：

1. 生成请求 — 根据当前 MediaTailor 会话状态计算UrlHeaders、和Body表达式。这些评估值构成出站 HTTP 请求。
2. 处理响应 -在 HTTP 调用完成后， MediaTailor 计算输出块中的表达式。这些表达式既可以引用原始会话状态，也可以引用调用返回的response对象。

响应字段

HTTP 调用完成后，您可以在输出表达式中引用以下字段：

字段	Type	说明
response.body	对象或数组	响应正文解析为 JSON。null如果正文超过 20,000 个字符或不是有效的 JSON，则设置为。
response.statusCode	整数	外部服务返回的 HTTP 状态码。null在网络出现故障时设置为。

字段	Type	说明
response.text	字符串	原始响应正文为字符串，截断为 20,000 个字符。"Internal Error"在网络出现故障时设置为。

Important

该response.body字段是响应超过 20,000 个字符null时的字段，即使响应是有效的 JSON 也是如此。

Note

响应对象仅在HTTP_REQUEST函数的 Output 模块中可用。您不能在 Url、Headers 或 Body 字段中引用响应字段。在中SEQUENTIAL_EXECUTOR，每个HTTP_REQUEST函数只能访问自己的响应。

值为null表示数据不可用。当 HTTP 调用失败（网络错误或超时）或者响应正文超过 20,000 个字符或不是有效的 JSON 时，就会发生这种情况。

网络故障行为

如果 HTTP 调用由于网络错误或超时而失败，response.statusCoderesponse.body则设置为null，且response.text设置为"Internal Error"。您的输出表达式仍在运行，因此response.statusCode在使用响应数据之前，请务必进行检查。

Tip

使用条件表达式来优雅地处理故障：{%response.statusCode = 200 ? response.body.value : 'default'%}

示例：获取身份数据

以下函数在会话开始时调用身份解析 API，并将结果存储在玩家参数中。它是为PRE_SESSION_INITIALIZATION生命周期挂钩设计的。

```
{
    "FunctionId": "fetchIdentityEnvelope",
    "FunctionType": "HTTP_REQUEST",
    "HttpRequestConfiguration": {
        "Runtime": "JSONATA",
        "MethodType": "GET",
        "Url": "{%'https://identity.example.com/v1/resolve?ip=' &
 $encodeUrlComponent(session.client_ip)%}",
        "Headers": {
            "Authorization": "{%'Bearer my_api_token'%}",
            "Accept": "application/json"
        },
        "RequestTimeoutMilliseconds": 2000,
        "Output": {
            "player_params.identity_envelope": "{%response.statusCode = 200 ?
 response.body.envelope : ''%}"
        }
    }
}
```

有关类似示例的完整演练，请参阅函数示例。

顺序执行器

何时使用

A 按顺序SEQUENTIAL_EXECUTOR运行一系列函数。每个步骤都可以使用之前的步骤的结果。

SEQUENTIAL_EXECUTOR当您的逻辑需要多个相互依赖的结果的步骤时使用。常见用例包括获取身份数据，然后使用它来检索受众细分，运行分类步骤，然后根据结果有条件地调用不同的外部服务，以及从多个数据源构建复杂的广告请求网址。

配置字段

SEQUENTIAL_EXECUTOR函数具有以下字段：

- 运行时 -表达式语言。将此设置为JSONATA。

- FunctionList— 包含 1 到 10 个步骤的有序列表。每个步骤都指定要运行FunctionId的函数。或者，您可以添加RunCondition表达式来控制该步骤是运行还是跳过。
- 输出 -定义所有步骤完成后要生成的值。每个条目将一个输出键（例如player_params.envelope）映射到一个表达式，该表达式可以引用序列中任何步骤生成的数据。如果省略，则使用序列中各个函数的所有输出。
- TimeoutMilliseconds（必填）— 整个序列完成的最长时间。如果序列超过此超时时间，则 MediaTailor 丢弃该序列的所有输出。

有序执行和数据流

MediaTailor 从头到尾运行序列中的每个步骤。每个步骤完成后，其生成的值将合并为一组正在运行的结果。后续步骤可以访问原始会话数据以及先前步骤生成的所有值。

临时数据是在步骤之间传递数据的主要机制。当一个函数写入一个temp.*键时，下一步可以读取该值。先前步骤编写的玩家参数和广告请求字段也可供后续步骤查看。

> **Note**
>
> 临时数据接受任何数据类型，包括对象和数组。玩家参数和广告请求字段仅接受字符串、数字、布尔值和空值）。

Per-step 运行条件

序列中的每个步骤都有一个可选RunCondition字段。此字段包含返回true或的表达式false。MediaTailor 在运行该步骤之前立即计算RunCondition表达式。

如果RunCondition表达式的计算结果为false，则完全 MediaTailor 跳过该步骤并移至下一个步骤。如果省略该RunCondition字段，则该步骤将始终运行。

```
{ "FunctionId": "retryFetch", "RunCondition": "{%temp.statusCode = 500%}" }
```

此机制允许您构建条件管道。例如，您可以在步骤 1 中运行身份提取，然后仅在步骤 1 返回有效身份时才有条件地在步骤 2 中运行区段查询。

输出模块的工作原理

a 上的输出模块SEQUENTIAL_EXECUTOR控制序列在所有步骤完成后生成的内容：

- 输出模块存在 — MediaTailor 根据最终累积状态评估输出模块中的表达式，并仅保存这些输出。先前步骤生成的未在顺序输出模块中引用的任何输出都将被丢弃。
- 输出模块不存在 — 直接 MediaTailor 保存所有步骤的所有累积输出。

> **Tip**
>
> 当你想让每个函数的输出都通过时，可以省略输出块。当需要在保存累积结果之前过滤、重命名或转换时，请添加输出块。

超时配置

该`TimeoutMilliseconds`字段为整个序列设置了截止日期。此超时涵盖所有步骤，包括函数发出的任何 HTTP 调用。如果序列超过超时时间，则 MediaTailor 丢弃该序列的所有输出，并像没有附加任何函数一样继续操作。

各个`HTTP_REQUEST`功能仍然尊重自己的`RequestTimeoutMilliseconds`设置。序列超时充当外部边界，限制总执行时间。

示例：在 HTTP 失败时重试

此示例调用身份 API，并在第一次调用返回服务器错误时自动重试。它使用由 SEQUENTIAL_EXECUTOR 编排的两个 HTTP_REQUEST 函数。

第 1 步 — 主要 fetch (**`fetchIdentity`**)：

```
{
    "FunctionId": "fetchIdentity",
    "FunctionType": "HTTP_REQUEST",
    "HttpRequestConfiguration": {
        "Runtime": "JSONATA",
        "MethodType": "GET",
        "Url": "{%'https://identity.example.com/v1/resolve?ip=' & session.client_ip%}",
        "Headers": {
            "Accept": "application/json"
        },
        "RequestTimeoutMilliseconds": 1000,
        "Output": {
            "temp.statusCode": "{%response.statusCode%}",
            "temp.envelope": "{%response.statusCode = 200 ? response.body.envelope :
 null%}"
```

```
        }
    }
}
```

第 2 步 — 失败时重试 (**`retryIdentity`**) :

```
{
    "FunctionId": "retryIdentity",
    "FunctionType": "HTTP_REQUEST",
    "HttpRequestConfiguration": {
        "Runtime": "JSONATA",
        "MethodType": "GET",
        "Url": "{%'https://identity-fallback.example.com/v1/resolve?ip=' &
 session.client_ip%}",
        "Headers": {
            "Accept": "application/json"
        },
        "RequestTimeoutMilliseconds": 1000,
        "Output": {
            "temp.statusCode": "{%response.statusCode%}",
            "temp.envelope": "{%response.statusCode = 200 ? response.body.envelope :
 null%}"
        }
    }
}
```

序列 (**`identityWithRetry`**):

```
{
    "FunctionId": "identityWithRetry",
    "FunctionType": "SEQUENTIAL_EXECUTOR",
    "SequentialExecutorConfiguration": {
        "Runtime": "JSONATA",
        "TimeoutMilliseconds": 2000,
        "FunctionList": [
            { "FunctionId": "fetchIdentity" },
            { "FunctionId": "retryIdentity", "RunCondition": "{%temp.statusCode >=
 500%}" }
        ],
        "Output": {
            "player_params.envelope": "{%temp.envelope%}"
        }
    }
```

```
}
```

工作原理：

1. fetchIdentity调用身份 API 并将状态码和信封写入temp.*。
2. 如果状态码为 500 或更高，则RunCondition步骤 2 的计算结果为true并retryIdentity运行。它会temp.envelope用重试响应覆盖temp.statusCode和。
3. 如果第一次调用成功，则跳过步骤 2。
4. 输出块写入temp.envelope的序列player_params.envelope。

并行执行者

何时使用

A 并行CONCURRENT_EXECUTOR运行一组子函数，直至达到最大并发度，然后在所有子函数完成后合并其输出。

CONCURRENT_EXECUTOR当您的逻辑需要多个不依赖彼此结果的独立运算时使用。常见用例包括对不同的丰富服务进行并行 HTTP 调用、同时运行独立的数据查询，以及通过重叠的网络请求减少总延迟。

配置字段

CONCURRENT_EXECUTOR函数具有以下字段：

- 运行时 -表达式语言。将此设置为JSONATA。
- FunctionList— 列出 1 到 10 个孩子并行跑的名单。每个条目都指定要运行FunctionId的函数。或者，您可以添加RunCondition表达式来控制孩子是运行还是被跳过。您还可以在执行人中Alias添加，为孩子指定一个唯一的名字。
- 输出 -定义所有子项完成后要生成的值。每个条目将一个输出键（例如player_params.device_type）映射到一个表达式，该表达式可以引用执行器中任何子级生成的数据。
- MaxConcurrency— 可以同时跑步的最大儿童人数。有效范围：1— 2。默认值：2。
- TimeoutMilliseconds（必填）— 整个执行者完成的最长时间。如果执行器超过此超时时间，则 MediaTailor 丢弃执行器的所有输出。默认值：2000。

并行执行和数据流

MediaTailor 同时启动孩子。当函数列表的条目多于时`MaxConcurrency`， MediaTailor 将在运行的子项完成后启动其他子项。同时运行的子节点数量保持在或低于并发限制。

每个孩子的结果都存储在其命名空间下——显式命名空间，`FunctionId`如果未设置别名`Alias`，则存储在子命名空间下。在所有子代完成后，执行者的`Output`表达式可以使用路径`function.<namespace>.<path>`引用任何孩子的结果。

> **Note**
>
> 执行器中所有子代的命名空间必须是唯一的。 MediaTailor 在创作时验证此要求。

Per-child 运行条件

执行者中的每个孩子都有一个可选`RunCondition`字段。此字段包含返回`true`或的表达式`false`。MediaTailor 在启动该子项之前立即计算`RunCondition`表达式。

如果`RunCondition`表达式的计算结果为`false`，则完全 MediaTailor跳过子节点。如果省略该`RunCondition`字段，则孩子将始终运行。

```
{ "FunctionId": "fetchDevice", "RunCondition": "{%session.player_params.device_id !=
 null%}" }
```

通过这种机制，你可以有条件地包括孩子。例如，当会话中不存在设备 ID 时，您可以跳过设备查询。

输出模块的工作原理

a 上的输出块`CONCURRENT_EXECUTOR`控制执行器在所有子项完成后生成的内容。每个条目都将一个输出键映射到一个表达式，该表达式可以使用引用任何孩子的结果`{%function.<namespace>.<path>%}`。

例如，如果子代的别名为`fetchUser`并写入`temp.user`，则执行器的输出表达式可以将其引用为`{%function.fetchUser.temp.user%}`。

超时配置

该`TimeoutMilliseconds`字段为整个执行者设置了截止日期。此超时涵盖所有子节点，包括函数发出的任何 HTTP 调用。如果执行器超过超时时间，则 MediaTailor 丢弃执行器的所有输出，并像没有附加任何函数一样继续执行。

各个HTTP_REQUEST功能仍然尊重自己的RequestTimeoutMilliseconds设置。执行器超时充当外部边界，限制总执行时间。

示例：并行扩充调用

此示例并行调用两个扩展 API，并将它们的结果合并到玩家参数中。它使用由 CONCURRENT_EXECUTOR 编排的两个 HTTP_REQUEST 函数。

孩子 1 — 用户资料 (**fetchUserProfile**)

```
{
    "FunctionId": "fetchUserProfile",
    "FunctionType": "HTTP_REQUEST",
    "HttpRequestConfiguration": {
        "Runtime": "JSONATA",
        "Url": "{%'https://enrichment.example.com/user/' &
 session.player_params.user_id%}",
        "Headers": {},
        "Output": {
            "temp.user_segment": "{%response.body.segment%}"
        }
    }
}
```

孩子 2 — 设备信息 (**fetchDeviceInfo**)

```
{
    "FunctionId": "fetchDeviceInfo",
    "FunctionType": "HTTP_REQUEST",
    "HttpRequestConfiguration": {
        "Runtime": "JSONATA",
        "Url": "{%'https://enrichment.example.com/device/' &
 session.player_params.device_id%}",
        "Headers": {},
        "Output": {
            "temp.device_category": "{%response.body.category%}"
        }
    }
}
```

执行者 () **parallelEnrichment**

```
{
```

```
    "FunctionId": "parallelEnrichment",
    "FunctionType": "CONCURRENT_EXECUTOR",
    "ConcurrentExecutorConfiguration": {
        "Runtime": "JSONATA",
        "MaxConcurrency": 2,
        "TimeoutMilliseconds": 1500,
        "FunctionList": [
            {
                "FunctionId": "fetchUserProfile",
                "Alias": "userCall"
            },
            {
                "FunctionId": "fetchDeviceInfo",
                "Alias": "deviceCall",
                "RunCondition": "{%session.player_params.device_id != null%}"
            }
        ],
        "Output": {
            "player_params.user_segment": "{%function.userCall.temp.user_segment%}",
            "player_params.device_category":
 "{%function.deviceCall.temp.device_category%}"
        }
    }
}
```

在本示例中：

1. MediaTailor 同时启动两个孩子（MaxConcurrency 是 2）。
2. `deviceCall`孩子只有在存在 (RunCondition) 时才`device_id`会运行。
3. 两个调用都完成后（在 1500 毫秒内），Output 模块将合并的结果写入`player_params`。
4. 如果超过 1500 毫秒的超时时间，则 MediaTailor 丢弃所有输出，在没有丰富数据的情况下继续操作。

构图规则

MediaTailor 对函数组合强制执行以下限制：

规则	限制
最大嵌套深度	2（执行器可以包含函数，但这些子级本身不能成为执行者）
每个序列的步数	1 到 10
每个并行执行者的子女数	1 到 10
最大并发度（并发执行器）	1 到 2
每个生命周期挂钩的函数执行总数	20
循环引用	不允许。函数不能直接或间接引用自身。
函数的存在	在创建或更新父执行器之前，中引用的所有函数都`FunctionList` 必须存在。
唯一的命名空间（并行执行器）	每个孩子在执行器中必须有唯一的别名或函数 ID。

创建函数时，MediaTailor 会验证表达式语法、检查受限函数、验证所有引用的函数是否存在，并检测循环引用。当您将函数附加到播放配置时，MediaTailor 还会验证整个函数树中的所有输出键是否与分配的生命周期挂钩兼容。

创建和管理函数

本页面将引导您使用 MediaTailor 控制台创建、附加、更新和删除函数。函数允许您在播放期间通过在关键点运行逻辑来自定义会话行为和广告请求。

设置函数

使用控制台创建函数

1. 打开 MediaTailor 控制台，网址为https://console.aws.amazon.com/mediatailor/。
2. 在导航窗格中，选择函数。
3. 选择创建函数。
4. 在函数 ID 中，输入函数的唯一名称（例如，`fetchIdentity`）。

5. 对于函数类型，选择CUSTOM_OUTPUTHTTP_REQUEST、或SEQUENTIAL_EXECUTOR。控制台显示所选类型的配置字段。
6. 填写特定类型的配置字段。有关每种函数类型及其字段的说明，请参见函数类型和构成。
7. 选择创建函数。

将函数附加到播放配置

只有通过函数映射（生命周期挂钩和函数之间的链接）将函数附加到播放配置后，该函数才会运行。函数映射指定哪个生命周期挂钩会触发该函数。

1. 打开控制 MediaTailor 台。
2. 在导航窗格中，选择配置。
3. 选择要更新的播放配置。
4. 在函数映射部分，选择编辑。
5. 对于每个生命周期挂钩，选择要附加的函数：
   - 会话初始化挂钩 — 选择要在会话开始时运行一次的函数。
   - 广告请求挂钩 — 选择要在每个 ADS 请求之前运行的函数。
6. 选择保存。

管理函数

查看功能

导航至 “函数” 以查看您账户中的所有功能。选择函数名称以查看其配置。

更新现有函数

要更新函数，请导航到函数，选择函数，然后修改配置。此更新将替换整个函数定义。

> **Tip**
>
> 在进行更改之前，请保存一份函数配置的副本。函数没有内置版本控制或回滚功能。

移除函数

将函数与播放配置分离

在删除某个函数之前，请将其从所有引用该函数的播放配置中删除。

1. 打开控制 MediaTailor 台。
2. 在导航窗格中，选择配置。
3. 选择播放配置。
4. 在函数映射部分，选择编辑。
5. 从生命周期挂钩中移除该函数。
6. 选择保存。

删除函数

导航到函数，选择函数，然后选择删除。

删除屏蔽规则

MediaTailor 防止您删除仍在使用的函数。

条件	结果
通过功能映射将函数附加到播放配置	删除已被阻止。先分离函数。
函数在 a 中被引用 `SEQUENTIAL_EXECUTOR`	删除已被阻止。先从父函数中移除引用。
任何地方都没有引用函数	删除成功。

验证规则

MediaTailor 在您创建或更新函数时对其进行验证。将执行以下检查：

- 表达式语法-所有表达式都必须是有效的 jsonata。
- 受限函数-表达式无法调用受限制的 JSonata 函数。jsonata 表达式参考有关完整列表，请参阅。
- 输出键前缀-所有输出键必须以可识别的命名空间前缀开头。有关可接受的前缀列表，请参见生命周期钩子。

- 函数引用-中的所有函数 ID 都FunctionList必须引用现有函数。
- 循环引用-函数不能直接或间接引用自身。
- 嵌套深度-A SEQUENTIAL_EXECUTOR 可以调用其他函数，但这些函数本身不能是 SEQUENTIAL_EXECUTOR s。

有关具体值和大小限制，请参阅限制。

API 参考

要以编程方式管理函数，请参阅 AWS Elemental MediaTailor API 参考。

函数的 jsonata 表达式参考

本页是 Functions 中可用的表达式语法、运算符和函数的完整参考。在为输出块、URL 字段、标头值、正文模板和运行条件编写表达式时使用它。

表达式分隔符

你在函数中定义的每个值要么是一个常量，要么是一个表达式，而不是两者的混合。 MediaTailor 根据分隔符区分两者。

语法	Type	评估
`https://ads.example.com/vast`	常量	按原样返回，未进行评估。
`{%session.client_ip%}`	Expression	在运行时进行评估。结果将替换整个值。
`GET`	常量	按原样返回。
`{%'https://ads.example.com/vast?ip=' & session.client_ip%}`	Expression	在运行时进行评估。

> **Important**
>
> 值要么完全是一个常数，要么完全是一个表达式。不能将两者混合成一个值。例如，`hello {%'world'%}` 是无效的。要将静态文本与动态值组合在一起，请在表达式中使用字符串连接:. `{%'hello ' & 'world'%}`

语言基础知识

用于路径导航的点符号

使用点符号遍历输入数据。每个点向下移动一级，进入对象层次结构。

```
session.client_ip              # the viewer's IP address
response.body.envelope         # a field inside a parsed JSON response
player_params.campaign_id      # a player parameter
```

缺失的字段返回时`null`不会引发错误：

```
temp.nonExistent               # null
temp.nonExistent.deeply.nested # null
```

字符串连接为 **&**

运&算符连接两个字符串值。 Non-string 值会自动转换为字符串。

```
'https://ads.example.com/vast?ip=' & session.client_ip
# "https://ads.example.com/vast?ip=192.0.2.1"

'duration=' & 30
# "duration=30"
```

条件（三元）表达式

使用三元运算符根据条件返回两个值中的一个。

```
condition ? value_if_true : value_if_false
```

示例：

```
$exists(player_params.env) ? player_params.env : 'prod'
response.statusCode = 200 ? response.body.id : 'unknown'
$random() > 0.5 ? 'groupA' : 'groupB'
```

你可以嵌套三元表达式进行多向分支：

```
$contains(session.user_agent, 'CTV') ? 'ctv'
  : $contains(session.user_agent, 'Mobile') ? 'mobile'
  : 'desktop'
```

变量绑定为 :=

使用括号内的:=运算符在表达式中分配中间值。绑定变量的作用域限制在括号内，并且不会在表达式之外保留。

```
(
  $base := 'https://ads.example.com';
  $base & '/vast?ip=' & session.client_ip
)
```

用分号分隔括号内的语句。最后一条语句是表达式的返回值。

```
(
  $code := response.statusCode;
  $code != null and $code >= 200 and $code < 300
    ? response.body.value
    : 'fallback'
)
```

运算符

算术

运算符	描述	示例	结果
+	加	5 + 3	8
-	减	10 - 4	6

运算符	描述	示例	结果
*	乘	6 * 7	42
/	除	15 / 4	3.75
%	取模	17 % 5	2

> **Important**
>
> 来自玩家参数和会话数据的输入值以字符串形式到达。$number()用于在数值比较或算术之前对其进行转换。将字符串与数字进行比较会产生意想不到的结果。

比较

运算符	描述	示例	结果
=	等于	response.statusCode = 200	true
!=	不等于	player_params.region != 'us-east-1'	true如果不是 us-east-1
<	Less than	avail.index < 3	true如果低于 3
>	Greater than	$number(player_params.age) > 18	true如果年满 18 岁
<=	Less than or equal	$count(items) <= 10	true如果 10 或更少
>=	Greater than or equal	response.statusCode >= 400	true如果是错误状态

布尔值

运算符	描述	示例
and	逻辑 AND	`response.statusCode = 200 and $exists(response.body.id)`
or	逻辑 OR	`player_params.region = 'us-east-1' or player_params.region = 'us-west-2'`

使用圆括号作为优先级：

```
score > 0.5 and (tier = 'premium' or tier = 'gold')
```

> **Note**
>
> $not()用于逻辑否定。没有not关键字运算符。

成员资格 (**in**)

该in运算符测试数组中是否存在一个值。

```
'premium' in segments     # true if segments contains 'premium'
player_params.region in ['us-east-1', 'us-west-2']  # true
```

链接 (**~>**)

链接运算符将左侧表达式的结果作为第一个参数传递给右边的函数。

```
session.user_agent ~> $lowercase ~> $trim
# equivalent to $trim($lowercase(session.user_agent))
```

允许的函数

MediaTailor 支持以下内置函数。此处未列出的任何函数都会被阻止，并且在您创建或更新函数时会导致验证错误。

类型转换 (3)

函数	说明	示例	结果
`$string(value)`	转换为字符串	`$string(200)`	`"200"`
`$number(value)`	转换为数字	`$number('42')`	`42`
`$boolean(value)`	转换为布尔值	`$boolean(1)`	`true`

内省 (4)

函数	说明	示例	结果
`$length(string)`	字符串长度	`$length('hello')`	`5`
`$count(array)`	数组元素数	`$count([1, 2, 3])`	`3`
`$exists(value)`	检查值是否存在（不是未定义的）	`$exists(temp.id)`	`true` 或 `false`
`$keys(object)`	获取对象密钥名称	`$keys(response.body)`	`["id", "name"]`

数字 (7)

函数	说明	示例	结果
`$sum(array)`	数组之和	`$sum([1, 2, 3])`	`6`

函数	说明	示例	结果
`$max(array)`	最大值	`$max([10, 5, 20])`	`20`
`$min(array)`	最小值	`$min([10, 5, 20])`	`5`
`$average(array)`	算术平均值	`$average([10, 20, 30])`	`20`
`$abs(number)`	绝对值	`$abs(-7)`	`7`
`$floor(number)`	向下舍入	`$floor(3.9)`	`3`
`$round(number, precision)`	四舍五入到精度	`$round(3.456, 2)`	`3.46`

字符串 (7)

函数	说明	示例	结果
`$uppercase(string)`	改为大写	`$uppercase('hello')`	`"HELLO"`
`$lowercase(string)`	改为小写	`$lowercase('Hello')`	`"hello"`
`$trim(string)`	移除 leading/trailing 空格	`$trim(' hi ')`	`"hi"`
`$substring(string, start, length)`	提取子字符串（从零开始）	`$substring('abcdef', 2, 3)`	`"cde"`
`$contains(string, pattern)`	检查字符串是否包含模式	`$contains(session.`	`true` 或 `false`

函数	说明	示例	结果
		user_agent, 'CTV')	
$match(string, pattern)	将字符串与正则表达式模式进行匹配	$match('abc-123', /[0-9]+/)	{"match": "123", ...}
$replace(string, pattern, replacement)	替换匹配的图案	$replace('hello', 'l', 'r')	"herro"

数组 (5)

函数	说明	示例	结果
$append(arr1, arr2)	连接数组	$append([1, 2], [3, 4])	[1, 2, 3, 4]
$reverse(array)	反向顺序	$reverse([1, 2, 3])	[3, 2, 1]
$sort(array)	排序数组	$sort([3, 1, 2])	[1, 2, 3]
$distinct(array)	移除重复项	$distinct([1, 2, 2, 3])	[1, 2, 3]
$map(array, func)	对每个元素应用函数	$map([1,2,3], function($v) {$v*2})	[2, 4, 6]

布尔值 (1)

函数	说明	示例	结果
`$not(value)`	不合逻辑	`$not(false)`	`true`

随机 (1)

函数	说明	示例	结果
`$random()`	介于 0（含）和 1（不包括）之间的随机数	`$random() > 0.5 ? 'A' : 'B'`	"A" 或 "B"

> **Note**
>
> `$random()`每次评估都会生成一个新值。如果您需要在多个输出键中使用相同的随机值，请先将其绑定到变量:`($r := $random(); ...)`.

Date/time (4)

函数	说明	示例	结果
`$now()`	以 ISO 8601 字符串表示的当前时间戳	`$now()`	`"2024-01-15T12:00:00.000Z"`
`$millis()`	当前时间戳，以自纪元以来的毫秒为单位	`$millis()`	`1705320000000`
`$toMillis(string)`	将 ISO 8601 字符串转换为毫秒	`$toMillis('2024-01-15T12:00:00.000Z')`	`1705320000000`

函数	说明	示例	结果
`$fromMillis(number)`	将毫秒转换为 ISO 8601 字符串	`$fromMillis(1705320000000)`	`"2024-01-15T12:00:00.000Z"`

编码 (6)

函数	说明	示例
`$encodeUrl(string)`	URL 编码（保留结构字符/，比如、、？）&	`$encodeUrl('https://example.com/path?q=hello world')`
`$encodeUrlComponent(string)`	对单个组件进行 URL 编码（对所有特殊字符进行编码）	`$encodeUrlComponent('a&b=c')` → `"a%26b%3Dc"`
`$decodeUrl(string)`	解码字符串 URL-encoded	`$decodeUrl('hello%20world')` → `"hello world"`
`$decodeUrlComponent(string)`	解码组件 URL-encoded	`$decodeUrlComponent('a%26b')` → `"a&b"`
`$base64encode(string)`	编码为 Base64	`$base64encode('hello')` → `"aGVsbG8="`
`$base64decode(string)`	从 Base64 解码	`$base64decode('aGVsbG8=')` → `"hello"`

> **Tip**
>
> `$encodeUrlComponent()`用于单个查询参数值。`$encodeUrl()`仅在需要对完整 URL 进行编码同时保留其结构时才使用。

常见模式

备用值

当值可能不存在时，请提供默认值。

```
{%$exists(player_params.region) ? player_params.region : 'us-east-1'%}
```

动态网址构造

根据多个输入生成广告决策服务器网址。

```
{%'https://ads.example.com/v1/vast?ip=' & $encodeUrlComponent(session.client_ip) &
 '&ua=' & $encodeUrlComponent(session.user_agent) & '&sid=' & session.id%}
```

状态码检查 HTTP_REQUEST 输出

保护输出值免受 HTTP 故障的影响。

```
{%response.statusCode != null and response.statusCode = 200 ? response.body.envelope :
 'default-envelope'%}
```

根据玩家参数进行数值转换

玩家参数以字符串形式到达。在算术或数字比较之前对其进行转换。

```
{%$number(player_params.max_duration) > 30 ? 'long' : 'short'%}
```

> **Important**
>
> 如果`$number()`收到非数字字符串，则返回`undefined`。将其与`$exists()`参数可能缺失或无效的时间结合起来:`($val := $number(player_params.max_duration); $exists($val) and $val > 30 ? 'long' : 'short')`.

随机流量分割

使用将观众分配到实验组`$random()`。

```
{%$random() > 0.5 ? 'https://ads.example.com/v1/vast-a' : 'https://ads.example.com/v1/
vast-b'%}
```

设备类型分类

根据用户代理字符串对设备进行分类。

```
{%$contains(session.user_agent, 'CTV') ? 'ctv' : $contains(session.user_agent,
 'Mobile') ? 'mobile' : 'desktop'%}
```

函数示例

在研究这些示例之前，您应该熟悉：

- 生命周期钩子— 每个生命周期挂钩中都可用的输入字段和输出命名空间。
- 函数类型和构成— 每种函数类型的工作原理及其配置字段。
- 创建和管理— 如何创建功能并将其附加到播放配置中。

本页为常见用例提供了完整、有效的功能配置。每个示例都包括场景、完整配置、函数映射以及对函数运行时会发生什么的解释。

示例	场景描述
the section called “示例 1：数据扩充”	在会话开始时获取一个 LiveRamp 身份信封并将其存储在玩家参数中。
the section called “示例 2： A/B 流量分割”	在两个广告决策服务器网址之间平均分配广告请求流量以进行 A/B 测试。

示例 1：数据扩充

场景

流媒体服务希望在会话开始时从外部服务（在本例中为 LiveRamp）获取受众身份数据，并将其存储在玩家参数中，以便在会话后期进行广告定位。身份信封是由身份解析服务提供的加密标识符，允许广告系统在不泄露个人信息的情况下跨设备和平台识别观众。

配置

获取 LiveRamp 信封 (HTTP_REQUEST)：

```
{
```

```
    "FunctionId": "fetchLiveRamp",
    "FunctionType": "HTTP_REQUEST",
    "HttpRequestConfiguration": {
        "Runtime": "JSONATA",
        "MethodType": "GET",
        "Url": "{%'https://api.example-identity.com/v2/envelope?pid=12345&iv=' &
 player_params.hashedId%}",
        "Headers": {
            "Authorization": "{%'Bearer ramp-token-789'%}"
        },
        "RequestTimeoutMilliseconds": 2000,
        "Output": {
            "player_params.envelope_id": "{%response.body.envelopes[0].value%}"
        }
    }
}
```

> **Note**
>
> `Bearer ramp-token-789`替换为你自己的 API 凭证。不要将敏感令牌直接存储在生产环境中的函数配置中。考虑使用代币轮换策略。

`RequestTimeoutMilliseconds`根据外部服务的预期响应时间进行设置。较低的值可以减少延迟，但会增加出现超时错误的机会。

函数映射

```
{
    "FunctionMapping": {
        "PRE_SESSION_INITIALIZATION": "fetchLiveRamp"
    }
}
```

函数运行时会发生什么

1. 观看者开始播放会话。
2. MediaTailor 运行`PRE_SESSION_INITIALIZATION`生命周期挂钩并运行`fetchLiveRamp`。
3. 该函数使用构建请求网址`player_params.hashedId`并调用 LiveRamp API。在网址中，`pid`是合作伙伴 ID，是经过哈希`iv`处理的查看者标识符。

4. 输出表达式从响应数组 (`response.body.envelopes[0].value`) 的第一个信封中提取值并将其写入`player_params.envelope_id`。

> **Tip**
>
> 要明确处理错误，请在访问响应数据`response.statusCode`之前进行检查：`{%response.statusCode = 200 ? response.body.envelopes[0].value : ''%}`

有关更多信息，请参阅HTTP 请求、Pre-session 初始化、故障排除和监控和限制。

示例 2： A/B 流量分割

场景

一家流媒体服务希望使用 CUSTOM_OUTPUT 函数在两个广告决策服务器 (ADS) 网址之间随机分配广告请求流量进行 A/B 测试。一半的广告请求发送到 v1 终端节点，一半发送到 v2 终端节点。

配置

A/B 流量分割（自定义输出）：

```
{
    "FunctionId": "abTestAdsUrl",
    "FunctionType": "CUSTOM_OUTPUT",
    "CustomOutputConfiguration": {
        "Runtime": "JSONATA",
        "Output": {
            "adsRequest.url": "{%$random() < 0.5 ? 'https://ads.example.com/v1/
decision?session=' & session.id : 'https://ads.example.com/v2/decision?session=' &
 session.id%}"
        }
    }
}
```

在此配置中：

- `FunctionId`— 此函数的唯一名称。
- `FunctionType`— 在不进行 HTTP 调用的情况下`CUSTOM_OUTPUT`评估表达式。

- `Output`— 映射`adsRequest.url`到随机选择两个 ADS 网址之一的表达式。

函数映射

```
{
    "FunctionMapping": {
        "PRE_ADS_REQUEST": "abTestAdsUrl"
    }
}
```

函数运行时会发生什么

1. 在播放过程中遇到广告中断。
2. MediaTailor 运行`PRE_ADS_REQUEST`生命周期挂钩并运行`abTestAdsUrl`。
3. 该`$random()`函数返回一个介于 0 和 1 之间的值。如果该值小于 0.5，则该函数将 ADS 网址设置为 v1 端点。否则，它会将 URL 设置为 v2 端点。

该`$random()`函数是在每个广告时段进行评估的，因此拆分是根据广告请求进行的，而不是按会话进行的。

> **Tip**
>
> 要调整流量分割率，请更改阈值。例如，`$random() < 0.8`将 80% 的流量发送到第一个 URL，将 20% 的流量发送到第二个 URL。

> **Note**
>
> 写入`adsRequest.url`会覆盖当前广告时段的播放配置中配置的默认 ADS 网址。

有关更多信息，请参阅 自定义输出、Pre-ads 请求 和 jsonata 表达式参考。

故障排除和监控功能

该页面可帮助您诊断常见的功能错误并监控生产中的功能性能。疑难解答部分按症状组织——从您观察到的内容开始，然后按照原因进行修复。

监控

CloudWatch 指标

MediaTailor 向 Amazon 发布函数执行指标 CloudWatch。无需选择加入。

Hook-level metrics — 每次生命周期挂钩执行一个数据点：

指标	说明	Dimensions
`PreSessionInitHook.Invocations`	挂钩执行次数	`ConfigurationName`
`PreSessionInitHook.Errors`	挂钩错误计数	`ConfigurationName`
`PreSessionInitHook.Latency`	挂钩执行时间 (ms)	`ConfigurationName`
`PreAdsRequestHook.Invocations`	挂钩执行次数	`ConfigurationName`
`PreAdsRequestHook.Errors`	挂钩错误计数	`ConfigurationName`
`PreAdsRequestHook.Latency`	挂钩执行时间 (ms)	`ConfigurationName`

Function-level metrics — 每个函数执行一个数据点：

指标	说明	Dimensions
`Function.Invocations`	函数执行次数	`ConfigurationName`，`FunctionId`，`FunctionType`，`HookType`
`Function.Errors`	函数错误计数	`ConfigurationName`，`FunctionId`，`FunctionType`，`HookType`

指标	说明	Dimensions
Function.Latency	函数执行时间 (ms)	ConfigurationName ，FunctionId ，FunctionType ，HookType

有关设置警报和使用这些指标的更多详细信息，请参阅监控 AWS Elemental MediaTailor 使用亚马逊 CloudWatch 指标。

日志事件

MediaTailor 为函数执行发出日志事件。默认情况下会发出错误事件。已完成事件和摘要事件是可选的。

事件类型	默认/ Opt-in	日志组	说明
PRE_SESSION_INIT_HOOK_SUMMARY	Opt-in	清单日志	挂钩执行摘要 (success/error)
PRE_SESSION_INIT_HOOK_ERROR	默认	清单日志	挂机失败，错误类型和原因
PRE_SESSION_INIT_FUNCTION_COMPLETED	Opt-in	清单日志	完成的个人功能有 input/output
PRE_SESSION_INIT_FUNCTION_ERROR	默认	清单日志	个别功能故障
PRE_ADS_REQUEST_HOOK_SUMMARY	Opt-in	ADS 互动日志	挂钩执行摘要 (success/error)

事件类型	默认/ Opt-in	日志组	说明
PRE_ADS_REQUEST_HOOK_ERROR	默认	ADS 互动日志	挂机失败，错误类型和原因
PRE_ADS_REQUEST_FUNCTION_COMPLETED	Opt-in	ADS 互动日志	完成的个人功能有 input/output
PRE_ADS_REQUEST_FUNCTION_ERROR	默认	ADS 互动日志	个别功能故障

要启用选择加入日志事件，请参阅监控 AWS Elemental MediaTailor 使用亚马逊 CloudWatch 指标。

使用该eventId字段将钩子级别和函数级事件关联起来，以实现相同的执行。

以下 Amaz CloudWatch on Logs Insights 查询筛选函数错误事件eventId以跟踪单次执行：

```
fields @timestamp, eventType, functionId, errorType, cause
| filter eventId = "5dc6f040-0f72-4e8c-a64e-25eeef62708c"
| sort @timestamp asc
```

问题排查

当函数失败时，会在错误事件中 MediaTailor 记录一个errorType字段。使用此字段来标识故障类别：

错误类型	说明
SYNTAX_ERROR	表达式编译失败或遇到运行时类型错误
RESOURCE_LIMIT_ERROR	表达式超出了 CPU 时间、内存或堆栈深度限制
RESTRICTION_ERROR	表达式使用了被屏蔽的函数，或者超过了输入有效载荷大小限制

错误类型	说明
TIMEOUT_ERROR	函数执行超过了其时间限制
VALIDATION_ERROR	输出路径的目标是当前挂钩作用域中不可写的字段
INTERNAL_ERROR	与功能无关的基础架构故障

这些条目按症状组织，并在适用的情况下引用这些错误类型。

表达式意外返回空值

症状：您期望填充的输出值在播放器参数中存在null或缺失。

可能的原因：

原因	如何识别	Fix
此生命周期挂钩中不存在输入字段。	你在PRE_SESSION_INITIALIZATION 函数adsRequest.url 中引用了。ADS 请求数据在会话开始时不可用。	将函数移至PRE_ADS_REQUEST 生命周期挂钩，或使用其他输入字段。请参阅生命周期钩子。
会话数据中缺少输入字段。	你引用了player_params.campaign_id ，但是玩家在会话初始化时没有传递该参数。	用于$exists() 在访问之前进行检查:{%$exists(player_params.campaign_id) ? player_params.campaign_id : 'default'%} 。
你向玩家参数或 ADS 请求写了一个对象或数组。	这些命名空间仅接受字符串、数字和布尔值。对象和数组会被过滤掉。	在中存储复杂数据，temp.*并在后续步骤中提取字符串、数字或布尔值。

R@@ **ESOURCE_LIMIT_ERROR#####**

症状：函数在出现errorType: "RESOURCE_LIMIT_ERROR"和时失败cause: "Stack overflow error"。

原因：表达式超过了 100 个等级的最大堆栈深度。这种情况通常发生在深度嵌套的条件 (if/then/else) 表达式或复杂的变量赋值中。

这意味着表达式的嵌套层次太多 MediaTailor ，无法处理。

修复：简化表达式。将复杂的逻辑分解为多个输出条目或顺序执行器中的多个步骤。

R@@ **ESOURCE_LIMIT_ERROR**：CPU 超时

症状：函数在出现errorType: "RESOURCE_LIMIT_ERROR"和时失败cause: "Expression evaluation timeout: Check for infinite loop"。

原因：表达式超过了 100 毫秒的 CPU 时间限制。在大型数据结构上执行昂贵计算的表达式可能会发生这种情况。

修复：降低表达式的复杂性。如果您正在处理大型数组，请考虑将该逻辑移至外部服务并使用HTTP_REQUEST函数调用它。

##########函数

症状：函数在出现errorType: "RESTRICTION_ERROR"和时失败cause: "Function '<name>' is not allowed"。

原因：该表达式调用的 jsonata 函数不在允许的 38 个函数列表中。常见的示例包括$filter$reduce、$eval、$split、和$join。

修复：检查cause字段中是否有被屏蔽的函数名称。将其替换为允许的替代方案。有关允许jsonata 表达式参考的 38 个函数的完整列表，请参阅。

允许使用的常用函数包括$string$number、$substring、$contains、和$encodeUrlComponent。

RESTR@@ **ICTION_ERROR###**式太长

症状：无法使用创建或更新函数cause: "Expression length <actual> exceeds limit <limit>"。

原因：单个表达式超过 1,000 个字符。

修复：将表达式分成较小的部分。使用多个输出条目，或者在顺序执行器中将逻辑分成多个步骤。使用变量绑定 (:=) 来避免重复长子表达式。

HTTP 错误：状态码为空

症状：在HTTP_REQUEST函数的输出中，response.statusCode是null。

原因：无法访问外部 API、连接超时或出现网络错误。发生这种情况时， MediaTailor 将设置response.statusCode为null、设置response.body为null、设置response.text为"Internal Error"。

修复：response.statusCode在访问响应数据之前请务必进行检查：

```
{%response.statusCode = 200 ? response.body.value : 'default'%}
```

此表达式检查 HTTP 调用是否返回了 200 状态码。如果是，则使用响应值。否则，它将回退到默认值。

如果这种情况经常发生，请检查外部 API 是否正常。RequestTimeoutMilliseconds如果 API 很慢，可以考虑增加，或者如果你想快速失败，可以考虑减少它。

HTTP 错误：响应正文为空

症状：response.statusCode是 200 但response.body确实如此null。

原因：响应正文要么是无效的 JSON，要么超过 20,000 个字符。 MediaTailor 只能将最多 20,000 个字符的 JSON 响应解析为response.body。

修复：response.text用作后备。该response.text字段包含截断为 20,000 个字符的原始响应正文：

```
{%response.statusCode = 200 ? ($exists(response.body.id) ? response.body.id :
 $substring(response.text, 0, 100)) : 'error'%}
```

如果您需要的数据超过 20,000 个字符的限制，可以考虑要求外部 API 返回较小的响应（例如，通过请求特定字段）。

HTTP 错误：网址验证失败

症状：函数在运行时失败，并显示一条关于 URL 格式错误、使用无效方案或超过最大长度的消息。

可能的原因：

原因	Fix
该 URL 不使用http或https。	确保 URL 表达式生成一个以http://或开头的 URL https://。
表达式评估后，网址超过 2,048 个字符。	缩短网址。使用 POST 方法将较大的参数值移到请求正文。
网址格式不正确（不是有效的 URI）。	检查表达式中是否缺少字符或多余字符。$encodeUrlComponent() 用于可能包含特殊字符的查询参数值。

VALIDATION_ERROR

症状：函数失败errorType: "VALIDATION_ERROR"。此错误可能发生在创作时（创建或更新函数时），也可能发生在执行时（函数在会话期间运行时）。

可能的原因：

原因	示例	Fix
输出密钥的目标是当前挂钩上不可写入的命名空间。	adsRequest.url 在附加到的函数中写入PRE_SESSION_INITIALIZATION 。	检查生命周期挂钩中允许使用哪些输出命名空间。 PRE_SESSION_INITIALIZATION 只允许player_params.* 。要么将函数移到正确的挂钩上，要么更改输出键。请参阅生命周期钩子。
输出密钥使用无效字符。	一个输出键，比如player_params.device type （带空格）。只允许使用字母、数字、下划线和连字符。	将输出密钥重命名为仅使用有效字符。例如，player_params.device_type 改用。
输出密钥不是以有效的前缀开头。	类似于player_params.myVa	使用有效的输出前缀：player_pa

原因	示例	Fix
	lue 或custom.myValue 的输出键temp.myValue 。	rams.* temp.*、或adsRequest.* 。
JSonata 表达式存在语法错误。	缺少结尾引号或条件不完整:{%session.id & %} .	查看表达式中是否缺少引号、不匹配的圆括号或不支持的运算符（如??或）。?:
HTTP_REQUEST 函数中缺少必填字段。	URL 字段为空或未指定方法。	确保已设置 URL 和方法字段。方法必须为GET或POST。
由表达式构建的 URL 无效。	评估后的网址使用了不支持的方案ftp://，例如超过 2,048 个字符或格式错误。	验证 URL 表达式是否生成有效的http://或https:// URL。$encodeUrlComponent() 用于可能包含特殊字符的查询参数值。
HTTP 标头包含无效字符或使用受限名称。	标题值包含换行符，或者标题名称为host或transfer-encoding 。	从标题值中删除无效字符。避免使用受限的标头名称。HTTP 请求有关标题限制，请参阅。

检查错误日志事件中的cause字段，它会识别哪个字段或表达式未通过验证。

INTERNAL_ERROR

症状：函数失败errorType: "INTERNAL_ERROR"。

原因：发生了与您的功能配置无关的基础设施故障。

修复：重试请求。如果错误仍然存在，请联系 Supp AWS ort。

功能限制

本页列出了适用于函数的限制。这些限制在创作时（创建或更新函数时）和运行时（函数运行时）强制执行。

挂钩执行限制

钩子	限制	值
PRE_SESSION_INITIALIZATION	总超时时间	2,000 毫秒
PRE_SESSION_INITIALIZATION	总player_params 大小	1000 个字符
PRE_ADS_REQUEST	总超时时间	2,000 毫秒

这些超时涵盖了整个挂接执行，包括所有函数步骤和 HTTP 调用。如果挂接超过其超时时间，则 MediaTailor 丢弃所有输出并继续操作，就好像没有附加任何函数一样。

单个函数的超时（RequestTimeoutMilliseconds例如 HTTP_REQUEST 函数）必须符合挂接的总超时时间。例如，如果挂接超时时间为 2,000 毫秒，并且 HTTP_REQUEST 函数设置RequestTimeoutMilliseconds为 2,000 毫秒，则如果在 HTTP 调用之前或之后发生任何处理，则该函数可能会在完成之前超时。

player_params总大小包括所有输出密钥名称和值的组合字符数。例如，输出player_params.deviceType = "mobile"并将、和player_params.region = "us-east-1"计deviceType入mobile总region计 1,000 个字符。us-east-1

表达限制

限制	值	强制执行于
每个表达式的 CPU 时间	100 毫秒	运行时
堆栈深度	100	运行时
表达式长度	1000 个字符	编写

当表达式在运行时超过限制时，将 MediaTailor 停止该表达式并记录错误。有关特定错误类型的详细信息，请参阅故障排除和监控。

函数组合限制

限制	值
每个顺序执行器的步骤数	1–10
每个并行执行者的子女数	1–10
最大并发度（并发执行器）	1—2
默认超时（并发执行器）	2,000 毫秒
每个并发执行器的唯一命名空间	必需（在创作时验证）
最大嵌套深度	2
每个挂接的函数执行总数	20
每个函数的输出条目	20
循环引用	不允许

这些限制是在创作时强制执行的。

HTTP 请求限制

限制	值
网址长度	2,048 个字符
请求身体尺寸	64 KB
标题数	50
标题名称长度	256 个字符
标题值长度	8,192 个字符
请求超时	100—2,000 毫秒（客户可配置通过）`RequestTimeoutMilliseconds`

限制	值
允许的网址架构	`https`, `http`
受限标题	`Host`, `Transfer-Encoding` , `Content-Length` , `Connection`

URL 长度、正文大小、标头数量和标头大小限制在创作时强制执行。请求超时和受限标头在运行时强制执行。

如果函数设置了受限标头，则在创作时 MediaTailor 接受函数配置，但在执行期间发送 HTTP 请求时会删除该标头。

响应限制

限制	值
`response.body` 最大尺寸	20,000 字符
`response.text` 最大长度	20,000 字符

这些限制是在运行时强制执行的。

输出限制

限制	值
输出值最大长度（每键）	1000 个字符

这些限制是在运行时强制执行的。

允许的 JSonata 函数

有关允许的函数的完整列表，请参阅jsonata 表达式参考。

有关诊断与这些限制相关的错误的帮助，请参阅故障排除和监控。

MediaTailor ADS 请求的动态广告变量

AWS Elemental MediaTailor 使用动态广告变量将信息从您的查看会话传递到广告决策服务器 (ADS)。此信息可帮助广告为您的观众选择最相关的广告。

本节概述了动态广告变量，并提供了指向特定实施指南的链接。有关分步配置说明，请参阅以下各个主题。

动态变量类型

MediaTailor 支持四种类型的动态变量：

- 会话变量 -自动生成的值，例如会话 ID 和 SCTE-35 数据。请参阅MediaTailor ADS 请求的会话变量。
- 播放器变量 -视频播放器发送的自定义参数。请参阅MediaTailor ADS 请求的玩家变量。
- 带有配置别名的域变量 — 用于多源配置的动态 URL 域。
- 服务变量 -控制中会话级别行为的保留`aws.*`参数。 MediaTailor请参阅服务变量。
- 配置别名 -用于动态变量替换的预定义映射。请参阅配置别名。

常见使用案例

使用动态广告变量来：

- 将观众的受众特征和偏好传递给您的 ADS
- 根据地理位置将请求路由到不同的来源
- 通过集成实现时移查看 MediaPackage
- 实施 A/B 测试和故障转移方案

以下部分提供了有关将动态广告变量与一起使用的更多详细信息 MediaTailor。

主题

- MediaTailor ADS 请求的会话变量
- MediaTailor ADS 请求的玩家变量
- MediaTailor 多个内容源的域变量
- MediaTailor 用于会话控制的服务变量
- MediaTailor 配置别名概述

- MediaTailor 向 ADS 传递参数
- MediaTailor ADS 和源的参数路由
- MediaTailor 和 MediaPackage 时移视图集成
- MediaTailor 参数会话行为和持久性
- MediaTailor 参数参考和限制
- MediaTailor 参数疑难解答指南
- MediaTailor 配置别名疑难解答指南

有关参数格式化要求和故障排除，请参阅MediaTailor 参数参考和限制和MediaTailor 参数疑难解答指南。

MediaTailor ADS 请求的会话变量

AWS Elemental MediaTailor 当您配置为在模板 ADS URL 中指定本节中列出的一个或多个变量时，会将会话数据发送 AWS Elemental MediaTailor 到广告决策服务器 (ADS)。您可以使用单个变量，也可以连接多个变量以创建单个值。 MediaTailor 生成一些值并从清单和玩家的会话初始化请求等来源获取其余值。

下表描述了您可以在模板 ADS 请求 URL 配置中使用的会话数据变量。表中列出的章节编号对应于有线电信工程师协会 (SCTE) -35 规范《数字节目插入提示消息》的 2019a 版本。有关广告预取的详细信息，请参阅。预取广告

Name	可用于广告预取	SCTE-35 规格部分	Description
[avail.index]	是		一个数字，表示广告在索引中的位置。在播放会话开始时， MediaTailor 创建清单中所有可用广告的索引，并存储会话剩余部分的索引。当 MediaTailor 向 ADS 提出填补空缺的请求时，它会包含广告可利用率索引号。此参数使 ADS 能够通过使用竞争排除和频次上限等功能来改进广告选择。
[avail.random]	是		一个介于 0 到 10,000,000,000 之间的随机数字，作为一个长数， MediaTailor 它会为向广告发出的每个请求生成。某些广告服务器使用此参数来实现将广告与竞争对手公司的广告进行隔离等功能。

Name	可用于广告预取	SCTE-35 规格部分	Description
[scte.archive_allowed_flag]	是	10.3.3.1	一个可选的布尔值。当该值为 0 时，将对该分段施加录制限制。当此值为 1 时，不对该分段施加录制限制。
[scte.avail_num]	是	9.7.2.1	MediaTailor 从 SCTE-35 字段中解析出的值`avail_num` ，为长数。 MediaTailor 可以使用此值来指定线性广告可用数字。 值必须是整数。
[scte.avails_expected]	是	9,7.2.1	一个可选的长整数值，用于给出当前事件中预期的可用次数。
[scte.delivery_not_restricted_flag]	是	10.3.3.1	一个可选的布尔值。当此值为 0 时，将保留接下来的五位。当该值为 1 时，接下来的五位采用 SCTE-35 规格中描述的含义。
[scte.device_restrictions]	是	10.3.3.1	一个可选的整数值，用于表示三组预定义、独立和非分层的设备。有关此变量的更多信息，请参阅规范中的 segments_expect ed 描述。 SCTE-35
[scte.event_id]	是	9.1 和 9.7.2.1	MediaTailor 从 SCTE-35 字段中解析出的值`splice_event_id` ，为长数。 MediaTailor 使用此值来指定线性广告可获得性数字或填充广告服务器查询字符串，例如广告窗格位置。 值必须是整数。
[scte.no_regional_blackout_flag]	是	10.3.3.1	一个可选的布尔值。当该值为 0 时，区域封锁限制适用于该分段。当此值为 1 时，区域封锁限制不适用于该分段。

Name	可用于广告预取	SCTE-35 规格部分	Description
[scte.segment_num]	是	10.3.3.1	一个可选的整数值，用于对分段集合中的分段进行编号。有关此变量的更多信息，请参阅规范中的 segment_num 描述。 SCTE-35
[scte.segmentation_event_id]	是	10.3.3.1	MediaTailor 将此变量公开为。scte.event_id
[scte.segmentation_type_id]	是	10.3.3.1	一个可选的 8 位整数值，用于指定分段类型。有关此变量的更多信息，请参阅规范中的 segmentation_type_id 描述。 SCTE-35

Name	可用于广告预取	SCTE-35 规格部分	Description
[scte.segmentation_upid]	segmentation_upid_type ：是 private_data ：是	segmentation _upid: 10.3.3.1 托管私有 UPID：10.3.3.3	对应于 SCTE-35 segmentation_upid 元素。该segmentation_upid 元素包含segmentation_upid_type 和segmentation_upid_length 。 MediaTailor 支持以下segmentation_upid 类型： • 广告信息 (0x0E) -广告信息。有关更多信息，请参阅规范中的 s egmentation_upid 描述。 SCTE-35 • 托管私有 UPID (0x0C) -规范中定义的托管私有 UPID (MPU) 结构。 SCTE-35 MediaTailor 支持二进制或 DASH XML SCTE 表示法。 你可以在播客工作流程中使用这个结构。为此，请指定 32 位（4 字节）format_identifier ，并在private_data 属性中包含以下参数： `ABCD{"assetId":" my_program ","cueData": {"cueType":" theAdType ","key":" pb","value" :" 123456"}}` MediaTailor 解析前面的 JSON 中的值，并将它们传递到scte.segmentation_upid.assetId scte.segmentation_upid.cueData.key 、和scte.segmentation_upid.cueData.value 动态变量中。 • 用户定义 (0x01) -用户定义的结构。有关更多信息，请参阅规范中的 s egmentation_upid 描述。 SCTE-35

Name	可用于广告预取	SCTE-35 规格部分	Description
[scte.segmentation_upid.assetId]	是		与 Managed Private UPID (0xC) 配合使用，segmentation_ upid_type 用于播客工作流程。 MediaTailor该值从 MPU 的 private_data JSON 结构中的assetId参数中派生。有关更多信息，请参阅 Managed Private UPID JSON structure for a podbuster workflow。
[scte.segmentation_upid.cueData.key]	是		与 Managed Private UPID (0xC) 配合使用，segmentation_ upid_type 用于播客工作流程。 MediaTailor该值从 MPU 的 private_data JSON 结构中的cueData.key 参数中派生。有关更多信息，请参阅 Managed Private UPID JSON structure for a podbuster workflow。
[scte.segmentation_upid.cueData.value]	是		与 Managed Private UPID (0xC) 配合使用，segmentation_ upid_type 用于播客工作流程。 MediaTailor该值从 MPU 的 private_data JSON 结构中的cueData.key 参数中派生。有关更多信息，请参阅 Managed Private UPID JSON structure for a podbuster workflow。 值可以是字符串。

Name	可用于广告预取	SCTE-35 规格部分	Description
[scte.segmentation_upid.private_data.{index}]	是		与 Managed Private UPID (0xC) 配合使用，segmentation_upid_type 用于定向广告工作流程。 MediaTailor 拆分以冒号分隔的 UPID 标记并创建索引会话变量。索引对应于以冒号分隔的列表中的位置，忽略初始冒号中的前导空格。 例如，如果segmentation_upid = ":3213214:2313321/5:3943" ，那么： • [scte.segmentation_upid.private_data.0] = 3213214 • [scte.segmentation_upid.private_data.1] = 2313321/5 • [scte.segmentation_upid.private_data.2] = 3943 值可以是字符串。
[scte.segments_expected]	是	10.3.3.1	一个可选的整数值，它给出分段集合中单个分段的预期数量。有关此变量的更多信息，请参阅规范中的 segments_expect ed 描述。 SCTE-35
[scte.sub_segment_num]	是	10.3.3.1	一个可选的整数值，用于标识子分段集合中的特定子分段。有关此变量的更多信息，请参阅规范中的 sub_segment_num 描述。 SCTE-35
[scte.sub_segments_expected]	是	10.3.3.1	一个可选的整数值，它给出子分段集合中各个子分段的预期数量。有关此变量的更多信息，请参阅规范中的 sub_segments_expected 描述。 SCTE-35

Name	可用于广告预取	SCTE-35 规格部分	Description
[scte.unique_program_id]	是	9.7.2.1	MediaTailor 从 SCTE-35`splice_insert` 字段`unique_program_id` 解析出的整数值。ADS 使用唯一程序 ID (UPID) 为直播线性流提供程序级广告定向。如果 SCTE-35 命令不是 splice insert，则将其 MediaTailor 设置为空值。 值必须是整数。
[session.avail_duration_ms]	是		广告可用时段的持续时间（以毫秒为单位）。默认值为 300,000 毫秒。 AWS Elemental MediaTailor 从输入清单中获取持续时间值，如下所示： • 对于 HLS：从`#EXT-X-CUE-OUT: DURATION` 或标签中的值中 MediaTailor 获取持续时间。`#EXT-X-DATERANGE` 如果输入清单中这些标签中的广告有效期为空、无效或 0，则 MediaTailor 使用默认值。 • 对于 DASH：从事件持续时间中 MediaTailor 获取持续时间值（如果已指定）。否则，它将使用默认值。 • 对于 VOD：当 VOD 直播触发片前广告调用时，如果清单中不包含具有持续时间值的 SCTE 消息，则 MediaTailor 不会输入 [session.avail_duration_ms] 的持续时间，包括默认时长值。
[session.avail_duration_secs]	是		广告可用时段的持续时间（以秒为单位），四舍五入到最接近的秒数。 MediaTailor 确定该值的方式与确定该值的方法相同`[session.avail_duration_ms]` 。
[session.client_ip]	否		MediaTailor 请求来自的远程 IP 地址。如果 `X-forwarded-for` 标头已设置，则该值就是 MediaTailor 用于 `client_ip` 的值。

Name	可用于广告预取	SCTE-35 规格部分	Description
[session.id]	否		当前播放会话的唯一数字标识符。播放器针对某个会话发出的所有请求都具有相同的 ID，因此它可用于旨在关联针对单一查看的请求的 ADS 字段。
[session.referer]	否		通常是托管视频播放器的页面的网址。 MediaTailor 将此变量设置为玩家在请求中使用的`Referer`标头的值 MediaTailor。如果播放器未提供此标头， MediaTailor 会将 **[session.referer]** 留空。如果您在清单端点前使用内容分发网络 (CDN) 或代理，并且希望此变量出现，请在此处代理来自播放器的正确标头。
[session.user_agent]	否		从玩家的会话初始化请求中 MediaTailor 收到的`User-Agent` 标头。如果您要在清单终端节点前面使用 CDN 或代理，则必须在此处代理播放器中的正确标头。
[session.uuid]	否		替代方案**[session.id]** 。这是当前播放会话的唯一标识符，如下所示： `e039fd39-09f0-46b2-aca9-9871cc116cde`

Name	可用于广告预取	SCTE-35 规格部分	Description
[avail.source_content_time_epoch_ms]	否		对于 HLS，该值是开始使用的原始分段的 PDT。对于 DASH，该<SupplementalProperty> urn:scte:dash:utc-time 值是<Period>包含的值<EventStream> 。 • 对于预贴式 HLS 或 DASH 广告，[avail.source_content_time_epoch_ms] 是第一个 HLS 细分市场的 PDT，也是第一个区段的 <SupplementalProperty> urn:scte:dash:utc-time PDT。<Period>对于清单窗口较短的直播，每个观众的此值将根据他们开始观看直播的时间而有所不同。对于带有完整事件 DVR 窗口的直播，每个观众的值将相同。 • 对于由于无用性抑制而导致的部分可用性，[avail.source_content_time_epoch_ms] 是开始使用该漏洞的源内容区段的 PDT。例如，如果 MediaTailor 移除可用项的前 20 秒，则仍[avail.source_content_time_epoch_ms] 将设置为原始 PDT，不会移动 20 秒。 • 对于预取 ADS 请求，[avail.source_content_time_epoch_ms] 为空，但是客户可以使用预回调度检索动态变量进行设置。

Example

如果 ADS 要求使用唯一会话标识符传递名为 deviceSession 的查询参数，则 AWS Elemental MediaTailor 中的模板 ADS URL 可能类似于下面这样：

```
https://my.ads.server.com/path?deviceSession=[session.id]
```

AWS Elemental MediaTailor 自动为每个流生成一个唯一标识符，并输入该标识符来代替session.id。如果标识符是1234567，则向 ADS 发出的最终请求将如下所示： MediaTailor

```
https://my.ads.server.com/path?deviceSession=1234567
```

如果 ADS 需要传递多个查询参数，则中的模板 ADS 网址 AWS Elemental MediaTailor 可能如下所示：

```
https://my.ads.server.com/sample?
e=[scte.avails_expected]&f=[scte.segment_num]&g=[scte.segments_expected]&h=[scte.sub_segment_nu
```

以下 DASH 标记示例 XML 片段显示了如何使用`scte35:SpliceInsert`：

```
<Period start="PT444806.040S" id="123456" duration="PT15.000S">
  <EventStream timescale="90000" schemeIdUri="urn:scte:scte35:2013:xml">
    <Event duration="1350000">
      <scte35:SpliceInfoSection protocolVersion="0" ptsAdjustment="180832" tier="4095">
        <scte35:SpliceInsert spliceEventId="1234567890"
 spliceEventCancelIndicator="false" outOfNetworkIndicator="true"
 spliceImmediateFlag="false" uniqueProgramId="1" availNum="1" availsExpected="1">
          <scte35:Program><scte35:SpliceTime ptsTime="5672624400"/></scte35:Program>
          <scte35:BreakDuration autoReturn="true" duration="1350000"/>
        </scte35:SpliceInsert>
      </scte35:SpliceInfoSection>
```

以下 DASH 标记示例 XML 片段显示了如何使用`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>
```

以下 DASH 标记示例 XML 片段显示了如何使用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>/DAhAAAAAAAAAP/wEAUAAAHAf+9/fgAg9YDAAAAAAAA25aoh</Binary>
      </scte35:Signal>
    </Event>
    <Event presentationTime="1541436360" duration="24" id="30">
      <scte35:Signal xmlns="http://www.scte.org/schemas/35/2016">

 <scte35:Binary>QW5vdGhlciB0ZXN0IHN0cmluZyBmb3IgZW5jb2RpbmcgdG8gQmFzZTY0IGVuY29kZWQgYmluYXJ5Lg=
Binary>
      </scte35:Signal>
    </Event>
```

以下 HLS 标签示例显示了如何使用EXT-X-DATERANGE：

```
#EXT-X-DATERANGE:ID="splice-6FFFFFF0",START-DATE="2014-03-05T11:
15:00Z",PLANNED-DURATION=59.993,SCTE35-OUT=0xFC002F0000000000FF0
00014056FFFFFF000E011622DCAFF000052636200000000000A0008029896F50
000008700000000
```

以下 HLS 标签示例显示了如何使用EXT-X-CUE-OUT：

```
#EXT-OATCLS-SCTE35:/DA0AAAAAAAAAAAABQb+ADAQ6QAeAhxDVUVJQAAAO3/PAAEUrEoICAAAAAAg
+2UBNAAANvrtoQ==
#EXT-X-ASSET:CAID=0x0000000020FB6501
#EXT-X-CUE-OUT:201.467
```

以下 HLS 标签示例显示了如何使用EXT-X-SPLICEPOINT-SCTE35：

```
#EXT-X-SPLICEPOINT-SCTE35:/DA9AAAAAAAAAP/wBQb+uYbZqwAnAiVDVUVJAAAKqX//
AAEjW4AMEU1EU05CMDAxMTMyMjE5M19ONAAAmXz5JA==
```

以下示例显示了如何使用scte35:Binary解码：

```
{
  "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
    "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 ADS 请求的玩家变量

AWS Elemental MediaTailor 当您配置为在模板 ADS URL 中指定`player_params.<query_parameter_name>`变量时，会将从玩家那里收到的数据发送 AWS Elemental MediaTailor 到 ADS。例如，如果玩家向发送请求`user_id`中命名的查询参数 MediaTailor，则要在 ADS 请求中传递该数据，请将其包含`[player_params.user_id]`在 ADS 网址配置中。

这使您能够控制 ADS 请求中包含的查询参数。通常，您将 ADS 可识别的特殊查询参数添加到 ADS 请求 URL，并提供键-值对作为参数值。

以下过程中使用的示例将使用以下键-值对：

- 值为 1 的 param1：
- 值为 2 的 param2：

添加查询参数作为键-值对

1. 在中 AWS Elemental MediaTailor，配置 ADS 请求模板网址以引用参数。以下 URL 显示包含了示例参数：

   ```
   https://my.ads.com/path?param1=[player_params.param1]&param2=[player_params.param2]
   ```

2. （可选）对于服务器端广告跟踪报告，播放器 URL-encode 上的键值配对。当 MediaTailor 收到会话初始化请求时，它 URL-decodes 会先将这些值替换为 ADS 请求网址。

   > **Note**
   >
   > 如果您的 ADS 需要一个 URL-encoded 值， URL-encode 则玩家需要两次该值。这样，由完成的解码会为 AD MediaTailor S 生成一个曾经编码过的值。

   例如，如果发送到 ADS 的值的解码表示形式为`param1=value1:&param2=value2:`，则 URL-encoded表示形式为`param1=value1%3A&param2=value2%3A`。

3. 在来自玩家的会话初始化调用中，将键值对 MediaTailor 作为单个查询参数的值传递给。以下示例调用将为服务器端和客户端广告跟踪报告提供示例键-值对。

   - 服务器端广告跟踪报告请求示例-使用配对 URL-encoded

     HLS：

     ```
     <master>.m3u8?ads.param1=value1%3A&ads.param2=value2%3A
     ```

     DASH：

     ```
     <manifest>.mpd?ads.param1=value1%3A&ads.param2=value2%3A
     ```

   - 客户端广告跟踪报告的请求示例-不是 URL-encoding

     HLS：

     ```
     POST <master>.m3u8
         {
             "adsParams": {
                "param1": "value1:",
                "param2": "value2:"
            }
         }
     ```

     DASH：

     ```
     POST <manifest>.mpd
         {
             "adsParams": {
                "param1": "value1:",
                "param2": "value2:"
            }
         }
     ```

对于服务器端报告，在收到玩家请求时对参数进行 MediaTailor 解码。对于客户端报告，它不会改变 JSON 负载中接收到的参数。 MediaTailor 向 ADS 发送以下请求：

```
https://my.ads.com/<path>?param1=value1:&param2=value2:
```

这样，param1 和 param2 键-值对将作为 ADS 请求中的第一类查询参数包含在内。

MediaTailor 多个内容源的域变量

AWS Elemental MediaTailor 动态域变量允许您使用多个域名，例如网址中的 my-ads-server.com 部分http://my-ads-server.com，并在配置中使用玩家参数。这使您可以在单个配置中使用多个内容源或广告决策服务器 (ADS)。

您可以将域变量与任何包含 URI 的参数一起使用：

- AdDecisionServerUrl
- AdSegmentUrlPrefix
- ContentSegmentUrlPrefix
- LivePreroll.AdDecisionServerUrl
- VideoContentSourceUrl

域变量与配置别名一起使用来执行动态变量替换。配置别名将一组别名和值映射到用于动态域配置的玩家参数。有关设置过程，请参见创建和使用配置别名 MediaTailor。有关详细的参考信息，请参阅MediaTailor 配置别名概述。

MediaTailor 用于会话控制的服务变量

AWS Elemental MediaTailor 为控制会话级行为的服务变量保留aws.查询参数命名空间。与ads.参数（转发到 ADS）和manifest.参数（附加到个性化清单 URL）不同，aws.参数由源服务器或 ADS 直接使用 MediaTailor ，不会转发到源服务器或 ADS。

支持的参数

下表列出了可用于控制会话级别行为的aws.*参数。

参数	Type	值	默认值	说明
aws.startTime	ISO 8601 时间戳	例如，2026-06-17T10:00:00Z	未设置（实时加入）	在 DVR 窗口的特定时刻启动会话。MediaTailor 将时间戳解析为最近的分段边界并在 HLS 清单EXT-X-START:TIME-OFFSET 中发出。

参数	Type	值	默认值	说明
aws.preroll	String enum	enabled，disab 区分大小写）	enabled	控制是否在会话中插入前置广告。当时disabled，即使回放配置有，预先播放也会被抑制。LivePreRollConfiguration
aws.overlayAvails	String enum	on，off（不区分大小写）	on	控制是否为会话处理叠加（非线性）和可用空间。当时off，源清单中的叠加广告标记将被忽略，不会插入叠加广告。
aws.logMode	String enum	DEBUG, DISABLED	DISABLED	为会话启用详细的调试日志。设置为时DEBUG，向日志 MediaTailor 发送详细的会话 CloudWatch 日志以进行故障排除。
aws.availSuppressionMode	String enum	OFF,BEHIND_LIVE_EDGE ,AFTER_EDGE （不区分大小写）	OFF	控制是否根据广告相对于实时边缘的位置对其进行抑制。
aws.availSuppressionValue	持续时间	HH:MM:SS 格式（例如，00:00:10）	未设置	无用抑制的时间窗口。当availSuppressionMode 为BEHIND_LIVE_EDGE 或时为必填项AFTER_LIVE_EDGE 。
aws.availSuppressionFillPolicy	String enum	FULL_AVAIL_ONLY ，PARTIAVAIL （不区分大小写）	FULL_AVAIL_ONLY	当模式为时AFTER_LIVE_EDGE ，控制是否填充部分抑制的空白区域。

AWS.startTime

设置后，在aws.startTime距离指定程序日期时间最近的分段边界处 MediaTailor 启动会话。

用量

在清单请求中aws.startTime作为查询参数传递：

```
GET /v1/master/{hashed-account-id}/{origin-id}/{asset}.m3u8?
aws.startTime=2026-06-17T10:00:00Z
```

或者在显式会话初始化中，将其作为不带aws.前缀的顶级字段传递：

```
POST /v1/session/{hashed-account-id}/{origin-id}/{asset}.m3u8

{
    "startTime": "2026-06-17T10:00:00Z"
}
```

要求

以下要求适用于aws.startTime：

- 源清单必须包含区段上的 EXT-X-PROGRAM-DATE-TIME (PDT)。如果没有 PDT，aws.startTime则无法解析并被忽略。
- 仅适用于 HLS 直播会话（SSAI 和 SGAI）。

行为

下表描述了在不同场景中的aws.startTime行为：

场景	结果
DVR 窗口内的时间戳	从最近的分段边界开始，发射 EXT-X-START
时间戳早于 DVR 窗口	从窗口头夹紧到 3×目标持续时间
时间戳在实时边缘的 3x 目标持续时间内	视作实时边缘联接（否）EXT-X-START
实时边缘或之后的时间戳	普通实时加入
格式错误或不是 ISO 8601	忽略 — 回退到实时加入，已记录错误
清单没有 PDT	忽略 — 回退到实时加入，已记录错误

场景	结果
省略参数	默认：普通实时边缘联接

夹紧行为

如果在会话中期的起始点在 DVR 窗口之外老化，EXT-X-START则将从窗口头限定到 3×TargetDuration。3×TargetDuration 缓冲区符合 RFC 8216 §6.3.3 玩家缓冲建议。

Example使用 AWS.StartTime 清单输出

当玩家使用初始化会话aws.startTime=2026-06-17T10:00:00Z且解析后的偏移量为距离实时边缘 120 秒时，会 MediaTailor 发出：

```
#EXTM3U
#EXT-X-TARGETDURATION:6
#EXT-X-START:TIME-OFFSET=-120.120,PRECISE=YES
#EXT-X-MEDIA-SEQUENCE:500
#EXT-X-PROGRAM-DATE-TIME:2026-06-17T09:58:00.000Z
#EXTINF:6.006,
segment500.ts
#EXTINF:6.006,
segment501.ts
...
```

TIME-OFFSET=-120.120它告诉玩家在直播边缘后 120 秒，在距离请求的开始时间最近的片段边界处开始播放。

限制

适用以下限制：

- 会话初始化后，您无法更改此参数。
- EXT-X-PROGRAM-DATE-TIME在源清单中需要。
- EXT-X-START是玩家的提示—— MediaTailor 不能保证所有玩家都会兑现。

aws.preroll

当aws.preroll=disabled，即使播放配置有，也会 MediaTailor 禁止在会话中插入前置广告。LivePreRollConfiguration

用量

在清单请求中aws.preroll作为查询参数传递：

```
GET /v1/master/{hashed-account-id}/{origin-id}/{asset}.m3u8?aws.preroll=disabled
```

或者在显式会话初始化中，将其作为不带aws.前缀的顶级字段传递：

```
POST /v1/session/{hashed-account-id}/{origin-id}/{asset}.m3u8

{
    "preroll": "disabled"
}
```

要求

以下要求适用于aws.preroll：

- 播放配置必须有，LivePreRollConfiguration此参数才会生效。如果未配置预先投放，则设置此参数无效。
- 适用于 HLS 直播会话（包括 SSAI 和 SGAI）。

行为

下表描述了aws.preroll行为方式：

值	结果
enabled（或省略）	Pre-roll 像往常一样插入
disabled	Pre-roll 在此会话中已禁用
值无效	记录为错误，视为 enabled

限制

会话初始化后，您无法更改此参数。

AWS.OverlayVails

当aws.overlayAvails=off， MediaTailor 忽略源清单中的叠加（非线性）广告标记，并且不会为会话插入叠加广告。

用量

在清单请求中aws.overlayAvails作为查询参数传递：

```
GET /v1/master/{hashed-account-id}/{origin-id}/{asset}.m3u8?aws.overlayAvails=off
```

要求

以下要求适用于aws.overlayAvails：

- 要使此参数生效，来源清单必须包含叠加广告标记（例如，具有叠加分段类型 SCTE-35 的事件）。
- 适用于 HLS 和 DASH、直播和 VOD 会话（包括 SSAI 和 SGAI）。

行为

下表描述了aws.overlayAvails行为方式：

值	结果
on（或省略）	我们照常处理叠加广告并插入广告
off	叠加广告标记被忽略，没有插入叠加广告
值无效	记录为错误，视为未指定（默认值：on）

限制

会话初始化后，您无法更改此参数。

AWS.LogMode

当时aws.logMode=DEBUG，为会话 MediaTailor 启用详细的调试日志记录。调试日志发送到日 CloudWatch 志，提供有关清单个性化、广告决策服务器请求和会话状态的详细信息，这对于解决广告插入问题很有用。

用量

在清单请求中aws.logMode作为查询参数传递：

```
GET /v1/master/{hashed-account-id}/{origin-id}/{asset}.m3u8?aws.logMode=DEBUG
```

要求

以下要求适用于aws.logMode：

- 播放配置必须启用（PercentEnabled > 0或EnabledLoggingStrategies配置）日志记录才能发出调试日志。
- 调试日志记录对每个客户都有速率限制，以防止日志量过大。

行为

下表描述了aws.logMode行为方式：

值	结果
DEBUG	为会话发出的详细调试日志
DISABLED（或省略）	正常的记录行为（基于播放配置设置）
值无效	引发错误，会话初始化失败

限制

适用以下限制：

- 会话初始化后，您无法更改此参数。
- 值区分大小写（DEBUG，不区分debug）。

aws.avai SuppressionMode

控制是否根据广告相对于实时边缘的位置对其进行抑制。使用它可以跳过落在直播边缘之后或之后的时间窗口内的广告插播时间，例如，避免填充观众在活动中途加入直播时已经经过的广告插播时间。

用量

在清单请求中将空闲抑制参数作为查询参数传递：

```
GET /v1/master/{hashed-account-id}/{origin-id}/{asset}.m3u8?
aws.availSuppressionMode=BEHIND_LIVE_EDGE&aws.availSuppressionValue=00:00:10
```

此参数与两个配套参数一起使用：

- aws.availSuppressionValue— 时间窗口（非模式时为必填项OFF）
- aws.availSuppressionFillPolicy— 控制部分填充行为（仅适用于AFTER_LIVE_EDGE模式）

要求

以下要求适用于空闲抑制参数：

- 适用于 HLS 和 DASH 直播会话。
- aws.availSuppressionValue当模式为BEHIND_LIVE_EDGE或时，必须HH:MM:SS以格式提供AFTER_LIVE_EDGE。
- aws.availSuppressionFillPolicy仅在模式为时有效AFTER_LIVE_EDGE。

模式行为

下表描述了每种抑制模式的效果：

Mode	效果
OFF（或省略）	抑制无济于事 — 所有广告插播时间均正常填充
BEHIND_LIVE_EDGE	抑制在指定时间窗口内在实时边缘后面开始的广告插播时间
AFTER_LIVE_EDGE	从实时边缘隐藏在指定时间窗口之后开始的广告插播时间

填写政策（仅限 AFTER_LIVE_EDGE）

下表描述了模式为时的填充策略行为AFTER_LIVE_EDGE：

填写政策	效果
FULL_AVAIL_ONLY （默认值）	仅填充完全处于抑制窗口之外的可用空间
PARTIAL_AVAIL	填补空余地中超出抑制窗口的部分

Example无效抑制示例

以下请求会在 Live Edge 后的 10 秒内禁止广告插播时间：

```
GET /v1/master/{hashed-account-id}/{origin-id}/{asset}.m3u8?
aws.availSuppressionMode=BEHIND_LIVE_EDGE&aws.availSuppressionValue=00:00:10
```

限制

适用以下限制：

- 会话初始化后，您无法更改此参数。
- aws.availSuppressionValue模式为时不得提供OFF。
- 中的时间格式无效，aws.availSuppressionValue导致模式回退到OFF。

MediaTailor 配置别名概述

AWS Elemental MediaTailor 配置别名允许在 URL 域和其他支持的字段中进行动态变量替换。使用此功能可在会话初始化 URLs 期间使用多个域并进行动态配置。

使用案例

配置别名可为以下场景启用复杂的多配置架构：

- 地理路由：使用特定地区的别名，根据观看者的位置将请求路由到不同的来源或广告服务器。有关实施指南，请参阅CloudFront 源故障转移。
- 基于内容的路由：将不同的内容类型定向到专门的来源或处理管道。有关路由行为的配置，请参阅the section called “设置 CDN 路由行为”。
- 故障转移场景：使用别名切换实现自动故障转移的备份源和广告服务器。有关实施的详细信息，请参见MediaTailor 使用 MQAR 实现多区域弹性和the section called “规划 CDN 集成”。

- A/B 测试：根据玩家参数路由流量，测试不同的广告服务器、来源或配置。有关负载测试指南，请参阅 CloudFront 使用真实用户监控功能为 Amazon 准备和运行性能测试。
- 设备特定优化：针对不同的设备类型或功能优化内容投放和广告投放。有关全面的指导，请参阅使用 MediaTailor、 MediaPackage和 CDN 设置清单筛选。
- 负载平衡：使用动态路由在多个来源或广告服务器之间分配负载。有关实施指导，请参见MediaTailor 使用 MQAR 实现多区域弹性和the section called “规划 CDN 集成”。

支持的字段

您可以在以下配置字段中使用动态变量：

- `VideoContentSourceUrl`
- `AdDecisionServerUrl`
- `LivePreroll.AdDecisionServerUrl`
- `AdSegmentUrlPrefix`
- `ContentSegmentUrlPrefix`
- `TranscodeProfileName`
- `SlateAdUrl`
- `StartUrl`
- `EndUrl`

以下各节介绍如何使用配置别名。

主题

- 创建和使用配置别名 MediaTailor
- 带有 MediaTailor 使用示例的配置别名

创建和使用配置别名 MediaTailor

在开始使用域变量之前，需要为配置创建配置别名。在会话初始化时，您可以使用配置别名作为域替换变量。

限制

使用配置别名时，请注意以下限制：

- 域中使用的所有动态变量都必须定义为ConfigurationAliases动态变量。
- 玩家参数变量前缀必须为。player_params.例如 player_params.origin_domain。
- 对于处于临界状态 URLs (,,ContentSegmentUrlPrefix) 的域变量VideoContentSourceUrl，AdSegmentUrlPrefix别名值列表必须详尽无遗。
- 如果请求的关键 URLs 域变量未指定动态变量或使用无效别名，则请求将失败，并显示 HTTP 400 状态码。非关键字段（SlateAdUrl、TranscodeProfileName、bump URLs er）将记录警告，但不会使请求失败。

缺少别名的后备行为

当找不到配置别名或配置别名无效时，将 MediaTailor 实现以下回退行为：

- 域变量：如果域变量别名缺失或无效，则请求将失败，并显示 HTTP 400 状态码。所有域变量都必须定义有效的别名。
- 非域变量：对于的非域部分中使用的变量 URLs（例如路径元素或查询参数），缺少别名会导致替换空字符串。
- 配置验证： MediaTailor 验证在配置创建和更新操作期间是否存在所有必需的别名。

步骤 1：创建配置别名

要使用 MediaTailor控制台创建用于域替换的配置别名，请执行以下步骤。

Console

使用控制台创建配置别名

1. 打开 MediaTailor 控制台，网址为https://console.aws.amazon.com/mediatailor/。
2. 在配置页面的配置别名部分，选择添加玩家参数。
3. 在 P layer 参数中，输入要用作动态变量的玩家参数的名称。例如 player_params.origin_domain。
4. 对于别名，请输入要用于玩家参数的别名及其值。
5. 选择确定。

   AWS Elemental MediaTailor 在 “配置别名” 部分的表中显示新参数。
6. 重复上述步骤以添加更多播放器参数。
7. 选择保存。

API

使用 API 创建配置别名

创建或更新 MediaTailor 配置时，请使用具有以下 JSON 结构的ConfigurationAliases参数：

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

步骤 2：在会话初始化中使用配置别名

设置配置别名后，可以在会话初始化请求中将其用作域的替换变量。这使您能够动态配置会话的域。

Example基本配置别名示例

以下是包含配置别名和动态域变量的配置的基本示例：

```
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使用别名进行会话初始化

使用上述配置，使用玩家变量和别名的会话初始化请求将类似于以下内容：

```
POST index.m3u8
{
    "playerParams": {
        "origin_domain": "pdx",
        "region": "pdx",
        "endpoint_id": "pdx",
        "ad_type": "customized"
    }
}
```

MediaTailor 用配置别名配置中的映射值替换别名字符串。

向 ADS 发出的请求将如下所示：

```
https://abc.execute-api.us-west-2.amazonaws.com/ads?sid=[session.id]&ad_type=abc12345
```

向来源索取清单的请求将如下所示：

```
https://abc.mediapackage.us-west-2.amazonaws.com/out/v1/abcd
```

带有 MediaTailor 使用示例的配置别名

以下示例显示了如何使用配置别名的完整 MediaTailor 配置、带别名的会话初始化请求以及别名的处理流程。

Example使用别名完成配置

以下示例显示了包括配置别名和动态域变量的完整配置：

```
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使用别名进行会话初始化

以下示例显示了指定玩家变量和别名的会话初始化请求：

```
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参数处理流程

在以下示例中， MediaTailor 将别名字符串替换为配置别名中的映射值。处理后会产生以下请求：

- 广告请求：

```
https://abc.execute-api.us-west-2.amazonaws.com/ads?sid=[session.id]&ad_type=abc12345
```

- VideoContentSource 请求：

```
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 向 ADS 传递参数

AWS Elemental MediaTailor 支持使用以下步骤在向 ADS 发出的 MediaTailor 请求中设置动态变量。

- 有关查询参数支持的格式的信息，请参阅MediaTailor 参数参考和限制。

- 有关配置别名和域变量，请参见MediaTailor 配置别名概述。
- 如需对 ADS 请求进行其他自定义，请参阅高级用法。

会话初始化方法

MediaTailor 支持多种会话初始化和参数传递方法：

1. 带有请求正文的 POST：

```
POST <master>.m3u8
{
    "adsParams": {"param1": "value1", "param2": "value2"},
    "playerParams": {"param3": "value3"}
}
```

2. URL 中的查询参数：

```
GET <master>.m3u8?ads.param1=value1&ads.param2=value2&playerParams.param3=value3
```

> **Important**
>
> 在初始化时，您只能指定一次参数。在转发之前，配置别名解析为实际值。

将会话和播放器信息传递到 ADS

1. 与 ADS 合作，确定其响应广告查询所需的信息 AWS Elemental MediaTailor。
2. 在中创建使用满足 ADS MediaTailor 要求的模板 ADS 请求网址的配置。在 URL 中，包含静态参数和用于动态参数的占位符。在配置的 Ad decision server (广告决策服务器) 字段中输入您的模板 URL。

   在以下示例模板 URL 中，`correlation` 提供会话数据，`deviceType` 提供播放器数据：

   ```
   https://my.ads.server.com/path?
   correlation=[session.id]&deviceType=[player_params.deviceType]
   ```

3. 在播放器上，为 AWS Elemental MediaTailor 配置会话发起请求以便为播放器数据提供参数。在会话发起请求中包含您的参数，并在后续会话请求中省略它们。

玩家为初始化会话而进行的调用类型决定了玩家（客户端）或 MediaTailor （服务器）为会话提供广告跟踪报告。有关这两个选项的信息，请参阅报告广告跟踪数据。

根据您是需要服务器端还是客户端广告跟踪报告来进行以下类型的呼叫之一。在两个示例调用中，userID 适用于 ADS，auth_token 适用于源：

- （可选）要求提供服务器端广告跟踪报告 — 在要发送 MediaTailor 到 ADS 的参数前加上前缀。ads为您希望 MediaTailor 将其发送到源服务器的参数消除前缀：

  以下示例显示了向 HLS 和 DASH 发出的传入请求。 AWS Elemental MediaTailor MediaTailor deviceType在向 ADS 发出的请求中使用，auth_token在向源服务器发出的请求中使用。

  HLS 示例：

  ```
  GET master.m3u8?ads.deviceType=ipad&auth_token=kjhdsaf7gh
  ```

  DASH 示例：

  ```
  GET manifest.mpd?ads.deviceType=ipad&auth_token=kjhdsaf7gh
  ```

- （可选）要求客户端广告跟踪报告-为对象内的广告提供参数。adsParams

  HLS 示例：

  ```
  POST master.m3u8
      {
          "adsParams": {
             "deviceType": "ipad"
         }
      }
  ```

  DASH 示例：

  ```
  POST manifest.mpd
      {
          "adsParams": {
             "deviceType": "ipad"
         }
      }
  ```

当玩家发起会话时，将模板 ADS 请求 URL 中的变量 AWS Elemental MediaTailor 替换为会话数据和玩家的ads参数。它将剩余参数从播放器传递到源服务器。

Example MediaTailor 带有广告变量的请求

以下示例显示了来自 AWS Elemental MediaTailor 的针对 ADS 和源服务器的调用（对应于上述播放器的会话初始化调用示例）。

- MediaTailor 使用会话数据和玩家的设备类型调用 ADS：

```
https://my.ads.server.com/path?correlation=896976764&deviceType=ipad
```

- MediaTailor 使用玩家的授权令牌调用源服务器。
  - HLS 示例：

    ```
    https://my.origin.server.com/master.m3u8?auth_token=kjhdsaf7gh
    ```

  - DASH 示例：

    ```
    https://my.origin.server.com/manifest.mpd?auth_token=kjhdsaf7gh
    ```

高级用法

您可以通过播放器和会话数据以多种方式自定义 ADS 请求。您只需要包含 ADS 主机名。

以下示例显示了一些可用来自定义请求的方法：

- 连接播放器参数和会话参数以创建新参数。示例：

```
https://my.ads.com?key1=[player_params.value1][session.id]
```

- 使用播放器参数作为路径元素的一部分。示例：

```
https://my.ads.com/[player_params.path]?key=value
```

- 使用播放器参数传递路径元素和键本身，而不仅仅是值。示例：

```
https://my.ads.com/[player_params.path]?[player_params.key1]=[player_params.value1]
```

MediaTailor ADS 和源的参数路由

AWS Elemental MediaTailor routes 根据参数的前缀和用途将参数查询到不同的目的地。了解参数路由对于实现特定于起源的功能（例如时移查看）至关重要。 MediaPackage

参数路由规则

MediaTailor 对查询参数使用以下路由规则：

- 原始参数（无前缀）：没有特定前缀的参数会传递到源服务器以实现特定于起源的功能
- ADS 参数（**ads.**前缀）：前缀为的ads.参数发送到广告决策服务器
- 清单参数（**manifest.**前缀）：前缀的参数manifest.用于 CDN 路由和授权

Example参数路由示例

以下会话初始化演示了参数路由：

```
POST /v1/session/123456789/originId/index.m3u8
{
    "adsParams": {
        "param1": "value1",
        "param2": "value2"
    },
    "manifestParams": {
        "auth_token": "abc123"
    }
}
```

在本示例中：

- param1并param2被发送到 ADS
- auth_token用于 CDN 路由和授权
- 没有前缀的参数将传递给源服务器

源服务器参数行为

传递给源服务器的参数支持特定于来源的功能，例如时移查看、内容筛选和身份验证。

常见的原始参数用例

原点参数通常用于：

- Time-shifted 查看：`start`以及 MediaPackage 时移内容的`end`参数
- 内容身份验证：源服务器所需的身份验证令牌
- 内容过滤：控制返回哪些内容变体的参数
- Origin-specific 功能：源服务器用于内容处理的任何参数

> **Important**
>
> 参数在会话初始化时进行处理，并在整个会话中维护。要修改时移窗口等参数，必须使用更新的值创建一个新会话。

MediaTailor 和 MediaPackage 时移视图集成

AWS Elemental MediaTailor 可以将时移查看参数传递给 MediaPackage 起点，以启用开始和追赶查看功能。这种集成使观众可以从较早的时间点开始观看直播内容。

MediaPackage 时移观看参数

MediaPackage 支持以下可以传递的时移查看参数： MediaTailor

- `start`: 用于定义时移清单开始的 Epoch 或 ISO 8601 时间戳
- `end`: 定义时移清单结尾的 Epoch 或 ISO 8601 时间戳
- `time_delay`: 将内容可用性延迟指定秒数
- `manifest_window_seconds`: 请求比配置的窗口更短的清单

Example MediaTailor 使用 MediaPackage 时移参数初始化会话

以下示例显示如何使用时移查看参数初始化会话：

```
GET /v1/master/123456789/originId/index.m3u8?
start=2024-08-26T10:00:00Z&end=2024-08-26T11:00:00Z
```

或者使用显式会话初始化：

```
POST /v1/session/123456789/originId/index.m3u8
```

```
{
    "adsParams": {
        "param1": "value1"
    }
}
```

使用其他查询参数：

```
?start=2024-08-26T10:00:00Z&end=2024-08-26T11:00:00Z
```

会话期间的参数行为

Time-shifted 查看参数具有特定的行为特征：

- 会话初始化：创建会话时处理参数
- 参数持久化：参数在整个播放过程中保持与会话的关联
- 初始化后不可变：在活动会话期间无法更改参数
- 需要新会话：要修改时移窗口，请使用更新的参数值创建一个新会话

MediaPackage startover 窗口要求

要使用时移视图 MediaPackage，请确保满足以下条件：

1. 在 MediaPackage 终端节点上配置启动窗口（最多 24 小时）
2. 确保您的 CDN 将必要的查询参数转发到 MediaPackage
3. 在玩家会话中使用一致的播放窗口，以获得更好的 CDN 缓存
4. 验证开始和结束时间是否在配置的开始/结束时间范围内

> **Important**
>
> 使用时移视图时，在玩家会话中使用一致的播放窗口，而不是为每个观看者生成独特的开始或结束时间。这样可以更好地在 CDN 上进行缓存，并避免潜在的限制。

有关 MediaPackage 时移视图配置和参数的完整信息，请参阅AWS Elemental MediaPackage 用户指南 AWS Elemental MediaPackage中的使用Time-shifted 查看。

MediaTailor 参数会话行为和持久性

AWS Elemental MediaTailor 在会话初始化时处理参数，并在会话生命周期中对其进行维护。了解会话行为对于实现动态参数场景至关重要。

会话初始化方法

MediaTailor 支持使用参数进行会话初始化的多种方法：

1. 隐式会话初始化：初始清单请求中包含的参数

```
GET /v1/master/123456789/originId/index.m3u8?
manifest.auth_token=abc123&start=2024-08-26T10:00:00Z
```

2. 显式会话初始化 (POST)：请求正文中提供的参数

```
POST /v1/session/123456789/originId/index.m3u8
{
    "adsParams": {"param1": "value1"},
    "manifestParams": {"auth_token": "abc123"}
}
```

3. 显式会话初始化 (GET)：作为查询参数提供的参数

```
GET /v1/session/123456789/originId/index.m3u8?
ads.param1=value1&manifestParams.auth_token=abc123
```

参数持久性和不可变性

MediaTailor 参数行为遵循以下规则：

- One-time 规范：在会话初始化时只能指定一次参数
- Session-wide 持久性：参数在整个会话中保持不变
- 初始化后不可变：创建会话后无法修改参数
- 配置别名解析：别名在转发到目标之前先解析为实际值

参数修改场景

要在播放期间修改参数，请执行以下操作：

- 创建新会话：使用更新的参数值初始化新会话
- 玩家过渡：将玩家无缝过渡到新会话
- 参数继承：保留未更改的参数以保持一致性

Example修改时移参数

要从 1 小时窗口更改为 2 小时窗口，请执行以下操作：

1. 当前会话：`start=2024-08-26T10:00:00Z&end=2024-08-26T11:00:00Z`
2. 创建新会话：`start=2024-08-26T10:00:00Z&end=2024-08-26T12:00:00Z`
3. 将玩家切换到新的会话 URL

> **Important**
>
> 单个会话的多个多变量播放列表请求在第一次请求后不会更新参数。参数在会话期间保持不可变。

MediaTailor 参数参考和限制

在配置动态广告变量之前，请了解适用于所有 MediaTailor 配置的参数格式要求和限制。

AWS Elemental MediaTailor 提供有关参数字符限制、长度限制以及清单查询参数和 ADS 参数支持的格式的全面信息。

清单查询参数字符限制

清单查询参数支持特定字符，并定义了长度限制。

支持的字符（没有 URL-encoding）

您可以直接在清单查询参数中使用以下字符：

- 字母数字字符（A-Z，a-z，0-9）
- 周期 (.)
- 连字符 (-)
- 下划线 (_)
- 反斜杠 (\)

支持的字符有 URL-encoding

在以下情况下支持以下特殊字符 URL-encoded：

- 时期 (.) = %2E
- 短划线 (-) = %2D
- 下划线 (_) = %5F
- 百分比 (%) = %25
- 波浪号 (~) = %7E
- 正斜杠 (/) = %2F
- 星号 (*) = %2A
- 等于 (=) = %3D
- 问题 (?) = %3F

URL-encoding 支持

MediaTailor 在中使用百分号 (%) 时支持百分号 URL-encoding （例如，你好%20world = 你好世界）。您可以使用任何 URL-encoded 字符，只要它们 URL-encodings 根据 HTTP 规范有效。

不支持的字符

如果没有 URL-encoding:`:`、、、、、、/（正斜杠）`?&=%`，则不能在清单查询参数中使用以下字符。

> **Important**
>
> MediaTailor 不支持%%% 或 == 等双字符。由于字符限制，您不能使用完整的 URL 作为清单查询参数值。

长度限制

所有清单查询参数（键和值的组合）的总长度不得超过 2000 个字符。

ADS 参数长度限制

以下长度限制适用于向 ADS 发出的请求中使用的参数：

- ADS 参数名称：最多 10,000 个字符

- ADS 参数值：最多 25,000 个字符
- 广告网址：最多 25,000 个字符

MediaTailor 参数疑难解答指南

AWS Elemental MediaTailor 为解决中与参数相关的常见问题提供指导 MediaTailor，包括字符限制、URL 编码问题和配置别名错误。

字符限制错误

包含不支持的字符的参数值可能会导致错误或意外行为。

常见症状

以下症状可能表明存在字符限制问题：

- 参数未出现在清单 URL 中
- 会话初始化期间 HTTP 400 错误
- 参数值被截断或损坏
- 由于 URL 格式不正确，ADS 请求失败

解决步骤

要解决字符限制错误，请执行以下操作：

1. 查看不支持的字符的参数值：:、?、&、=、%/
2. 正确使用 URL-encoding 特殊字符（参见MediaTailor 参数参考和限制）
3. 避免使用双字符，例如%%%或 ==
4. 如果无法使用完整的 URL，请考虑使用其他参数格式

Example网址编码示例

而不是使用：

```
manifest.redirect_url=https://example.com/path?param=value
```

使用 URL-encoded 格式：

```
manifest.redirect_url=https%3A%2F%2Fexample.com%2Fpath%3Fparam%3Dvalue
```

长度限制错误

超过长度限制的参数可能会被截断或导致错误。

长度限制

以下长度限制适用（有关完整MediaTailor 参数参考和限制详细信息，请参见）：

- 清单查询参数（总计）：2000 个字符
- ADS 参数名称：10,000 个字符
- ADS 参数值：25,000 个字符
- 广告网址：25,000 个字符

解决策略

要处理长度限制，请执行以下操作：

1. 尽可能使用较短的参数名称和值
2. 将较大的参数值拆分成多个较小的参数
3. 使用配置别名将短别名映射到较长的值（参见）MediaTailor 配置别名概述
4. 考虑使用外部存储来存储带有参数引用的大型数据

配置别名错误

配置别名问题可能导致 HTTP 400 错误或意外的参数值。

常见的配置别名错误

配置别名通常会出现以下错误：

- HTTP 400 错误：别名值缺失或无效
- 域变量无法正确解析
- 玩家参数未替换为别名值

分辨率清单

要解决配置别名错误，请执行以下操作：

1. 验证所有域变量均定义为 ConfigurationAliases
2. 确保玩家参数变量使用player_params.前缀
3. 确认关键 URL 中的域变量的别名值列表是详尽无遗的 (VideoContentSourceUrl,,AdSegmentUrlPrefix) ContentSegmentUrlPrefix
4. 检查会话初始化请求是否指定了有效的别名值
5. 验证 ConfigurationAliases 参数的 JSON 结构

有关详细的故障排除指南，请参阅MediaTailor 配置别名疑难解答指南。

Example配置别名验证

确保您的配置包含所有必需的别名：

```
"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
    }
}
```

参数处理流程问题

了解参数处理流程有助于解决参数转发和转换的问题。

参数处理顺序

MediaTailor 按以下顺序处理参数：

1. 会话初始化参数验证
2. 配置别名解析（如果适用）
3. 参数过滤（ADS vs Origin vs 清单）
4. URL 编码和格式
5. 将参数应用到 URL

调试参数流

要调试参数处理问题，请执行以下操作：

1. 验证会话初始化中是否正确指定了参数
2. 检查配置别名是否解析为预期值
3. 确认参数出现在正确的网址中（清单、广告、来源）
4. 验证是否正确应用了 URL 编码

Example参数流示例

会话初始化：

```
POST master.m3u8
{
    "playerParams": {"origin_domain": "pdx"},
    "manifestParams": {"test": "123"}
}
```

在别名解析和处理之后：

- 来源请求：`https://abc.mediapackage.us-west-2.amazonaws.com/out/v1/abcd`
- 清单网址：`/v1/master/.../index.m3u8?aws.sessionId=session&test=123`

安全性注意事项和最佳实践

MediaTailor 为参数处理实施安全措施，以防止常见的安全问题。

安全措施

MediaTailor 实施以下安全措施：

1. 输入大小限制以防止数据库膨胀
2. 对用户输入进行正确编码和消毒
3. URL-encoding 输入以防止响应损坏

最佳实践

请遵循以下安全参数处理的最佳实践：

- 发送前在客户端验证参数值

- 使用配置别名限制可能的参数值
- 避免在参数中包含敏感信息
- 监控异常模式的参数使用情况
- 将参数值保持在建议的长度限制内

MediaTailor 配置别名疑难解答指南

AWS Elemental MediaTailor 为常见的配置别名问题和错误场景提供系统的故障排除指导。

配置别名验证错误

当配置别名缺失或无效时，MediaTailor 会返回特定的错误响应以帮助识别问题。

常见错误场景

下表描述了常见的配置别名错误及其解决步骤：

错误	原因	解决方案
HTTP 400：玩家参数别名无效	中找不到玩家参数值 ConfigurationAliases	验证玩家参数值是否作为键存在于相应的 ConfigurationAliases 映射中
HTTP 400：缺少所需的配置别名	未使用相应 ConfigurationAliases条目的域变量	将缺失的玩家参数添加到所有必需 ConfigurationAliases 的别名映射中
HTTP 400：配置验证失败	ConfigurationAliases 结构格式错误或不完整	验证 JSON 结构并确保所有域变量都有相应的别名
在 URL 中替换空字符串	Non-domain 未找到变量别名	添加缺失的别名映射或在中提供默认值 ConfigurationAliases

验证核对清单

使用以下清单来验证您的配置别名设置：

1. 域变量覆盖范围：确保在 URL 的域名部分中使用的所有变量都有相应的 ConfigurationAliases 条目
2. 别名完整性：验证所有可能的玩家参数值均作为密钥包含在别名映射中
3. JSON 结构：验证 ConfigurationAliases JSON 的格式和嵌套是否正确

4. 参数命名：确认所有玩家参数都使用前player_params.缀
5. 值一致性：确保别名值对其预期用途（URL、配置文件名称等）有效

调试配置别名解析

遵循这种系统的方法来调试配置别名解析问题。

Step-by-step 调试方法

使用以下步骤识别和解决配置别名问题：

配置别名调试程序

1. 验证配置结构：确认您的播放配置包含正确的格式 ConfigurationAliases

```
{
    "ConfigurationAliases": {
        "player_params.example_param": {
            "alias1": "value1",
            "alias2": "value2"
        }
    }
}
```

2. 检查播放器参数格式：确保会话初始化包含格式正确的播放器参数

```
{
    "playerParams": {
        "example_param": "alias1"
    }
}
```

3. 验证别名映射：确认玩家参数值（“alias1”）作为键存在于映射中 ConfigurationAliases
4. 使用简单配置进行测试：从最低配置开始以隔离问题
5. 监控错误响应：检查特定验证消息的 MediaTailor错误响应
6. 验证已解析的 URL：确认最终解析的 URL 有效且可访问

配置别名最佳实践

遵循这些最佳实践，确保实现可靠的配置别名。

安全注意事项

使用配置别名时实施以下安全措施：

- 输入验证：在别名解析中使用所有玩家参数值之前对其进行验证
- 别名值清理：确保别名值仅包含预期的字符和格式
- 域名限制：将域名限制为可信、受控的域
- 访问控制：仅限授权人员修改配置

性能优化

使用以下建议优化配置别名性能：

- 最大限度地减少别名数：仅使用必要的别名以减少处理开销
- 高效命名：对别名和参数使用清晰、一致的命名规范
- 默认值：为常见用例提供合理的默认别名
- 配置缓存：利用 MediaTailor配置缓存来提高性能

维护和监控

使用以下做法保持可靠的配置别名操作：

- 定期验证：定期验证所有别名映射是否最新且功能正常
- 错误监控：监控与别名缺失或无效相关的 HTTP 400 错误
- 文档：保留所有别名映射及其用途的清晰文档
- 测试程序：对所有别名组合实施全面测试

MediaTailor 清单查询参数

AWS Elemental MediaTailor 处理用于不同目的的查询参数：用于 CDN 路由和授权的清单查询参数，以及可用于源特定功能的其他查询参数。

AWS Elemental MediaTailor 保留会话初始化的查询参数并将其附加到个性化清单 URLs 和其他资产。当您在客户端播放器之间有内容分发网络 (CDN) 时 MediaTailor ，请使用此功能。

当您的 CDN 需要以下内容的查询参数时，请使用清单查询参数：

- 动态路由到不同的 MediaTailor 端点
- 令牌授权

客户端行为与 CDN 行为

MediaTailor 为客户端报告端点添加查询参数，但不会为 CDN 分段附加查询参数。更新后的功能为各种 MediaTailor 资产的查询参数提供了更全面的支持，增强了 CDN 路由和授权用例的灵活性。

MediaTailor 为客户端报告端点添加查询参数，但它不会为 CloudFront （或其他 CDN）细分追加查询参数。

要使用参数保存，请联系 AWS S upport，请求启用清单查询参数通过。

在 HLS 和 DASH 之间，以及显式和隐式会话初始化之间，行为会有所不同。以下主题介绍如何配置会话初始化请求，以便将参数 MediaTailor 传递到清单：

MediaTailor 原点的查询参数处理

AWS Elemental MediaTailor 根据查询参数的用途，对查询参数的处理方式有所不同。清单查询参数（前缀为`manifest.`）用于 CDN 路由和授权，而其他查询参数可用于源特定功能。

时移观看 MediaPackage

要使用时移查看功能 MediaPackage，您可以在请求中包含`start`和`end`参数。这些参数定义了特定的内容窗口，用于从头到尾查看和追赶查看。

Example时移观看请求

在清单请求中加入开始和结束参数，以便进行时移查看：

```
GET /v1/master/111122223333/originId/index.m3u8?
start=2024-08-26T10:00:00Z&end=2024-08-26T11:00:00Z
```

会话期间的参数行为

查询参数在会话初始化时处理。对于时移观看或其他特定于原点的功能：

- 会话初始化：在初始清单请求中包含必需的参数
- 参数持久性：参数与会话相关联，并在整个播放过程中保持不变
- 更改参数：要修改时移窗口或其他参数，请使用更新的值创建一个新会话

> **Important**
>
> 查询参数的具体处理取决于您的来源配置和内容来源支持的参数。要进行 MediaPackage 集成，请确保您的 CDN 已配置为转发必要的查询参数，如中所述。配置基本查询参数

MediaTailor 参数字符限制和 URL 编码

AWS Elemental MediaTailor 支持清单查询参数中的特定字符。您可以对特殊字符使用 URL 编码。

支持的带有 URL 编码的字符

URL 编码支持以下特殊字符：

- 时期 (.) = %2E
- 短划线 (-) = %2D
- 下划线 (_) = %5F
- 百分比 (%) = %25
- 波浪号 (~) = %7E
- 正斜杠 (/) = %2F
- 星号 (*) = %2A
- 等于 (=) = %3D
- 问题 (?) = %3F

网址编码支持

MediaTailor 当你在 URL 编码中使用百分号 (%) 时（例如，hello%20world = hello world = hello world），则支持该符号。您可以使用任何 URL 编码字符，只要它们是符合 HTTP 规范的有效 URL 编码即可。

> **Important**
>
> MediaTailor 不支持双字符，例如%%% 或 ==。

安全注意事项

MediaTailor 为参数处理实现了以下安全措施：

1. 防止数据库膨胀的输入大小限制
2. 对用户输入进行适当的编码和审查
3. 对输入进行网址编码，以防止响应损坏

主题

- MediaTailor HLS 隐式会话初始化
- MediaTailor DASH 隐式会话初始
- MediaTailor HLS 和 DASH 显式会话初始化
- MediaTailor 特定于协议的参数行为
- MediaTailor CDN 集成和参数路由

MediaTailor HLS 隐式会话初始化

AWS Elemental MediaTailor 如果您的请求包含带有密钥的查询参数，则在 MediaTailor 资源链接中包含查询参数`manifest.*`。以下示例显示了这种请求格式：

```
GET /v1/master/111122223333/originId/index.m3u8?manifest.test=123&other=456
```

链接不包含前`manifest.`缀。

HLS 中的参数应用

对于 HLS 隐式会话，将参数 MediaTailor 应用于清单层次结构中的以下位置：

- 多变体播放列表 URLs
- 媒体播放列表 URLs
- 内容细分 URLs
- 广告细分 URLs
- HLS 初始化 URLs

Example多变播放列表

以下示例显示了如何在多变体播放列表的网址中 MediaTailor 包含查询参数：

```
#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媒体播放列表

以下示例显示了如何在中 MediaTailor 包含内容分段 URLs的查询参数：

```
#EXTM3U
#EXT-X-VERSION:6
#EXT-X-TARGETDURATION:7
#EXT-X-MEDIA-SEQUENCE:28716269
#EXT-X-DISCONTINUITY-SEQUENCE:0
#EXTINF:6.006,
https://origin.com/contentSegment_1.ts?originQueryParam=foo&test=123
#EXT-X-DISCONTINUITY
#EXTINF:6.006,
../../../../segment/111122223333/originId/session/0/2?test=123
```

MediaTailor DASH 隐式会话初始

AWS Elemental MediaTailor 为客户端创建会话，并在客户端在没有会话的情况下发出清单请求时使用查询参数重定向该会话。以下示例显示了这种请求格式：

```
GET /v1/dash/111122223333/originId/index.mpd?manifest.test=123&other=456
```

MediaTailor 为客户端创建会话并使用查询参数重定向该会话：

```
/v1/dash/111122223333/originId/index.mpd?sessionId=session&test=123
```

DASH 中的参数应用

DASH 清单响应包括不同位置的查询参数，包括内容细分、广告细分和初始化 URLs。 MediaTailor 将参数应用于以下内容：

- DASH 清单位置元素
- SegmentTemplate 初始化属性
- SegmentTemplate 媒体属性
- 内容细分 URLs
- 广告细分 URLs

当客户端发出请求时，会使用类似于以下示例的 DASH 清单进行 MediaTailor 响应。第一个句点是内容周期，因此 MediaTailor 不要在那里插入清单查询参数。在第二个时段（即广告时段）中，将清单查询参数 MediaTailor插入`SegmentTemplate`元素的`initialization`属性和`media`属性中。该`Location`元素还具有清单查询参数。

```
<?xml version="1.0" encoding="UTF-8"?>
<MPD availabilityStartTime="2018-07-27T09:48:23.634000+00:00"
 id="201" minBufferTime="PT30S" minimumUpdatePeriod="PT15S"
 profiles="urn:mpeg:dash:profile:isoff-live:2011" publishTime="2023-02-14T23:37:43"
 suggestedPresentationDelay="PT25.000S" timeShiftBufferDepth="PT56.997S" type="dynamic"
 xmlns="urn:mpeg:dash:schema:mpd:2011" xmlns:scte35="urn:scte:scte35:2013:xml"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 http://standards.iso.org/ittf/
PubliclyAvailableStandards/MPEG-DASH_schema_files/DASH-MPD.xsd">
    <BaseURL>https://origin.com/contentSegments/</BaseURL>
    <Location>https://mediatailor.com/v1/dash/111122223333/originId/index.mpd?
test=123&aws.sessionId=session</Location>
    <Period duration="PT29.963S" id="28737823" start="PT143732873.178S">
        <AdaptationSet bitstreamSwitching="true" mimeType="video/mp4"
 segmentAlignment="true" startWithSAP="1" subsegmentAlignment="true"
 subsegmentStartsWithSAP="1">
            <Representation bandwidth="2200000" codecs="avc1.640029"
 frameRate="30000/1001" height="540" id="1" width="960">
                <SegmentTemplate initialization="index_video_7_0_init.mp4?
m=1611174111" media="index_video_7_0_$Number$.mp4?m=1611174111"
 presentationTimeOffset="4311986195351" startNumber="28737828" timescale="30000">
                    <SegmentTimeline>
                        <S d="180180" t="4311986911066"/>
                        <S d="3003" t="4311987091246"/>
                    </SegmentTimeline>
                </SegmentTemplate>
            </Representation>
        </AdaptationSet>
    </Period>
```

```
    <Period id="28737829_1" start="PT39925H48M23.141S">
        <BaseURL>https://mediatailor.com/v1/
dashsegment/111122223333/originId/session/28737829/28737829_1/</BaseURL>
        <AdaptationSet bitstreamSwitching="false" frameRate="30000/1001"
 mimeType="video/mp4" segmentAlignment="true" startWithSAP="1"
 subsegmentAlignment="true" subsegmentStartsWithSAP="1">
            <SegmentTemplate startNumber="1" timescale="90000"/>
            <Representation bandwidth="2200000" codecs="avc1.64001f" height="540"
 id="1" width="960">
                <SegmentTemplate initialization="asset_540_2_0init.mp4?test=123"
 media="asset_540_2_0_$Number%09d$.mp4?test=123" startNumber="1" timescale="90000">
                    <SegmentTimeline>
                        <S d="180180" r="6" t="0"/>
                        <S d="87087" t="1261260"/>
                    </SegmentTimeline>
                </SegmentTemplate>
            </Representation>
        </AdaptationSet>
    </Period>
</MPD>
```

MediaTailor HLS 和 DASH 显式会话初始化

AWS Elemental MediaTailor 在多变播放列表中包含 a `manifestParams` s query 参数，并在客户端发出显式会话初始化请求时的响应 URLs 中进行跟踪。

会话初始化方法

对于显式会话初始化，你可以使用带有请求正文的 POST 或带有查询参数的 GET：

1. 带有请求正文的 POST：

```
POST /v1/session/111122223333/originId/index.m3u8
{
    "adsParams": {"param1": "value1", "param2": "value2", "param3": "value3"},
    "manifestParams": {"test": "123"}
}
```

2. 使用查询参数获取：

```
GET /v1/session/111122223333/originId/index.m3u8?
ads.param1=value1&ads.param2=value2&manifestParams.test=123
```

Example会话初始化请求

```
POST /v1/session/111122223333/originId/index.m3u8
{
    "adsParams": {
        "param1": "value1",
        "param2": "value2",
        "param3": "value3"
    },
    "manifestParams": {
        "test": "123"
    },
    "reportingMode": "client"
}
```

Example清单和追踪响应

```
{
    "manifestUrl": "/v1/master/111122223333/originId/index.m3u8?
aws.sessionId=session&test=123",
    "trackingUrl": "/v1/tracking/111122223333/originId/session?test=123"
}
```

会话的清单响应的具体内容与前面描述manifestParams的隐式会话初始化工作流程 MediaTailor URLs 类似。关键区别在于，显式会话初始化的清单参数不是以开头的manifest.。

清单查询参数是不可变的，您只能在会话初始化时进行设置。如果客户端为单个会话发出多个多变体播放列表请求，则 MediaTailor 不会在第一个请求之后更新清单查询参数。

参数处理流程

在初始化时，只能指定一次参数。在转发之前，配置别名会解析为实际值。例如：ad_type=abc12345根据 ConfigurationAliases配置player_params.ad_type=customized解析为。

MediaTailor 特定于协议的参数行为

AWS Elemental MediaTailor 对于 HLS 和 DASH 协议，对清单查询参数的处理方式有所不同。每种协议类型都有特定的应用位置和处理方法。

HLS 与 DASH 参数处理对比

下表比较了如何 MediaTailor 处理 HLS 和 DASH 协议中的清单查询参数：

方面	HLS 行为	达世币行为
参数应用	直接应用于清单 URLs 和区段 URLs	应用于位置元素、 SegmentTemplate 属性和区段 URLs
清单层次结构	多变体播放列表 → 媒体播放列表 → 片段	MPD → 时期 AdaptationSets → 陈述
初始化 URLs	存在时应用于 HLS 初始化 URLs	应用于 SegmentTemplate 初始化属性
会话处理	播放列表更新期间保留的参数	MPD 位置元素中包含的用于会话连续性的参数
广告细分处理	应用于媒体播放列表 URLs 中的广告区段	应用于广告周期 SegmentTemplate 媒体属性

参数应用程序位置

MediaTailor 将清单查询参数应用于以下位置：

HLS 参数应用程序

对于 HLS 流， MediaTailor 将清单查询参数应用于：

- 多变播放列表 URLs：参数附加到媒体播放列表引用中
- 媒体播放列表 URLs：参数包含在媒体播放列表的片段 URLs 中
- 内容分段 URLs：所有内容区段都包含清单查询参数
- 广告细分 URLs：广告区段接收用于 CDN 路由和授权的参数
- HLS 初始化 URLs：Init 分段包含流中存在的参数
- Slate 片段 URLs：Slate 内容包括用于一致 CDN 行为的参数

Example HLS 参数应用示例

鉴于会话初始化：

```
GET /v1/master/123456789/originId/index.m3u8?
manifest.auth_token=abc123&manifest.region=us-west
```

多变播放列表包含媒体播放列表参考中的参数：

```
#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
```

媒体播放列表包含片段中的参数 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
```

DASH 参数应用程序

对于 DASH 直播， MediaTailor 将清单查询参数应用于：

- MPD 位置元素：位置元素包括清单刷新请求的参数
- SegmentTemplate 初始化属性：Init 段 URLs 包含参数
- SegmentTemplate 媒体属性：媒体片段 URL 模板包含参数
- 内容区段 URLs：根据模板生成的所有内容区段都包含参数
- 广告细分 URLs：广告周期细分包含 CDN 集成的参数
- 服务器端举报重定向：302 重定向到广告细分会保留参数

Example DASH 参数应用示例

鉴于会话初始化：

```
GET /v1/dash/123456789/originId/index.mpd?
manifest.auth_token=abc123&manifest.region=us-west
```

DASH 清单包含多个位置的参数：

```
<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 集成和参数路由

AWS Elemental MediaTailor 清单查询参数支持复杂的 CDN 集成场景。您可以将它们用于动态路由、授权和负载平衡。

CDN 路由用例

受益于清单查询参数的常见 CDN 集成场景包括：

- 地理路由：根据查看者的位置将请求路由到特定区域的 MediaTailor 终端节点
- 基于令牌的授权：通过 CDN 将授权令牌传递给， MediaTailor 以实现安全的内容访问
- 负载平衡：使用 CDN 路由逻辑在多个 MediaTailor 端点之间分配流量
- A/B 测试：将不同的用户分段路由到不同的 MediaTailor 配置进行测试
- 设备特定优化：根据设备类型或功能路由请求

在 CDN 层之间保留参数

MediaTailor 确保在多个 CDN 层和请求类型中保留清单查询参数：

1. 初始请求：参数是从会话初始化请求中提取的
2. 清单生成：参数应用于清单 URLs 中的所有相关内容
3. 分段请求：所有分段中都包含参数， URLs 以实现一致的 CDN 行为
4. 广告插入：在广告插入和区段替换期间会保留参数

Example CDN 授权流程

以下示例演示了使用清单查询参数的完整的 CDN 授权流程：

1. 客户端请求以授权令牌出现：

```
GET https://cdn.example.com/mediatailor/v1/master/123456789/originId/index.m3u8?
manifest.auth_token=jwt_token_here&manifest.user_id=12345
```

2. CDN MediaTailor 使用以下参数将请求转发给：

```
GET https://mediatailor.amazonaws.com/v1/master/123456789/originId/index.m3u8?
manifest.auth_token=jwt_token_here&manifest.user_id=12345
```

3. MediaTailor 生成将参数应用于所有内容的清单 URLs：

```
#EXTM3U
#EXT-X-STREAM-INF:BANDWIDTH=2665212
../../../manifest/123456789/originId/session/0.m3u8?
auth_token=jwt_token_here&user_id=12345
```

4. 后续的分段请求包括用于 CDN 授权的参数：

```
GET https://cdn.example.com/mediatailor/segment/123456789/originId/session/0/1?
auth_token=jwt_token_here&user_id=12345
```

报告广告跟踪数据

MediaTailor 提供了两个选项，用于跟踪和报告观众观看了多少广告。在服务器端广告报告方法中，MediaTailor 跟踪广告并将信标（跟踪信号）直接发送到广告服务器。或者，在客户端跟踪方法中，客户玩家（用户的设备）跟踪广告并将信标发送到广告服务器。播放会话中使用的广告报告类型取决于播放器为启动会话而发出的具体请求 MediaTailor。

有关使用动态变量将会话和玩家数据传递到广告服务器的信息，请参阅MediaTailor ADS 请求的动态广告变量。有关会话初始化参数的详细信息，请参见MediaTailor 清单查询参数。

主题

- MediaTailor 服务器端广告跟踪和报告
- Client-side 广告追踪

MediaTailor 服务器端广告跟踪和报告

AWS Elemental MediaTailor 默认为服务器端报告，用于全面的广告跟踪和评估。借助服务器端报告，当播放器请求清单中的广告 URL 时，此服务会直接将广告使用情况报告至广告跟踪 URL。播放器使用初始化播放会话后 MediaTailor，无需您或播放器进一步输入即可执行服务器端报告。播放每则广告时， MediaTailor 会向广告服务器发送信标，以报告广告的浏览量。 MediaTailor 以四分位数为单位为广告开始和广告进度发送信标：第一个四分位数、中点、第三四分位数和广告完成时间。

执行服务器端广告报告

- 在播放器中，根据您的协议，使用以下格式之一的请求初始化新的 MediaTailor 播放会话：
  - 示例：HLS 格式

    ```
    GET <mediatailorURL>/v1/master/<hashed-account-id>/<origin-id>/<asset-id>?
    ads.<key-value-pairs-for-ads>&<key-value-pairs-for-origin-server>
    ```

  - 示例：DASH 格式

    ```
    GET <mediatailorURL>/v1/dash/<hashed-account-id>/<origin-id>/<asset-id>?ads.<key-
    value-pairs-for-ads>&<key-value-pairs-for-origin-server>
    ```

  这些键值对是广告跟踪的动态目标参数。有关向请求添加参数的信息，请参阅 the section called “动态广告变量”。

AWS Elemental MediaTailor 使用清单 URL 响应请求。清单包含媒体清单的 URL。媒体清单包含广告分段请求的嵌入式链接。

> **Note**
>
> 当跟踪网址中 MediaTailor 遇到双斜杠 (//) 时，它会将斜线折叠为一 (/)。

当播放器通过广告分段 URL (`/v1/segment` 路径) 请求播放时， AWS Elemental MediaTailor 会通过广告跟踪 URL 将相应的信标发送到广告服务器。同时，此服务会将重定向发送到实际的 `*.ts` 广告分段。广告区段要么位于 MediaTailor 存储转码广告的亚马逊 CloudFront 发行版中，要么位于您缓存广告的内容分发网络 (CDN) 中。

以下部分提供了有关使用服务器端广告跟踪的 MediaTailor更多信息。

主题

- Server-side 使用服务器引导的广告插入 (SGAI) 进行跟踪
- Server-side 追踪信标词汇表
- Server-side 跟踪时间和缓存行为
- MediaTailor 服务器端跟踪特性和功能

Server-side 使用服务器引导的广告插入 (SGAI) 进行跟踪

当您使用服务器引导式广告插入 (SGAI) 时，服务器端跟踪使用的无会话信标机制与上述拼接模式方法不同。SGAI 没有将广告片段 MediaTailor 拼接到内容清单中（用于跟踪/v1/segment请求），而是在资产列表响应中将广告参考作为单独的播放列表返回，并将信标元数据嵌入到广告 URI 中。

无会话服务器端信标的工作原理

以下步骤描述了服务器端信标如何用于 SGAI 会话：

1. 会话初始化：玩家使用请求HLS多变量播放列表。aws.insertionMode=GUIDED Server-side 报告是默认值（不需要aws.reportingMode参数）。与拼接模式不同，会话初始化响应不包含。trackingUrl
2. 可缓存清单： MediaTailor返回一个可缓存的清单，其中包含带有EXT-X-DATERANGE标签CLASS="com.apple.hls.interstitial"和指向 MediaTailor插页式资产列表端点的X-ASSET-LIST属性。
3. 包含信标元数据的资产列表：当玩家遇到广告插播时，它会获取资产列表。 MediaTailor返回 JSON 响应，其中每个广告 URI 都包含加密的信标元数据：

```
{
  "ASSETS": [
    {
      "DURATION": 30.0,
      "URI": "https://cdn.example.com/ad/master.m3u8?
awsBeaconData=<encrypted>&awsBeaconDomain=<MediaTailor-
endpoint>&awsConfigurationName=<config-name>"
    }
  ]
}
```

当服务器端报告处于活动状态时，响应中不包含TRACKING部分。广告 URI 携带所有信标数据。

4. HLS 变量替换：玩家获取广告的多变量播放列表。广告清单使用#EXT-X-DEFINE:QUERYPARAM指令通过 HLS 变量替换将信标参数从 URI 查询字符串传递到区段网址：

```
#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
```

   玩家使用广告清单 URI 查询字符串中的值解析{$awsBeaconData}{$awsBeaconDomain}、和{$awsConfigurationName}变量，然后通过 MediaTailor请求每个广告区段。

5. 信标在区段请求时触发：当玩家请求每个广告区段时，请求就会通过 MediaTailor。该服务解密信标数据，确定该区段在广告中的位置（曝光量、第一四分位数、中点、第三四分位数或完整），然后向广告服务器发射相应的 VAST 跟踪信标。 MediaTailor 然后将玩家重定向到实际的广告内容分段。

玩家对 SGAI 服务器端信标的要求

要在 SGAI 中使用服务器端信标，您的玩家必须满足以下要求：

- HLS 版本 11 或更高版本
- 支持 HLS 插页式广告的EXT-X-DATERANGE带CLASS属性
- 支持#EXT-X-DEFINE:QUERYPARAM变量替换 (RFC 8216bis)。玩家必须对查询参数值进行百分比解码，然后才能将其替换为区段网址。

> **Note**
>
> SGAI 服务器端信标目前仅支持 HLS。SGAI 服务器端信标尚不支持 DASH。

与拼接模式服务器端跟踪的比较

下表汇总了拼接广告和服务器引导广告插入之间的服务器端跟踪有何区别：

方面	缝合 (SSAI)	Server-guided (SGAI)
清单可缓存性	Per-session，不可缓存	可缓存，可与观众共享
广告区段路由	通过/v1/segment/ 使用会话 ID	通过/v1/segment/ 使用加密的信标数据 blob
信标的会话状态	每个会话存储在 MediaTailor	无会话 — 所有状态都包含在加密awsBeaconData 参数中
会话初始化时的跟踪 URL	在会话初始化响应中返回	未提供 — 信标数据嵌入在每个资产清单响应的广告 URI 中
DASH 支持	支持	暂不支持

> **Note**
>
> 对于实时 SGAI 会话，您可以使用启用基于清单的广告预取。aws.guidedPrefetchMode=MANIFEST这与缝合 (SSAI) 会话中使用的基于时间表的预取 API 是分开的。有关更多信息，请参阅 引导式预取，带有明显的心跳声。

Server-side 追踪信标词汇表

MediaTailor 服务器端跟踪使用一组标准化的信标向广告服务器和验证服务报告广告查看进度。这些信标符合互动广告局 (IAB) 的视频广告衡量标准，可提供广告曝光量和完成率的准确报告。

Server-side 跟踪信标类型

信标类型	被解雇时	用途	计时详情
印象	当玩家请求第一个广告区段时	表示广告内容已开始加载并即将向查看者显示	在第一次/v1/segment 请求广告时触发。符合 IAB 指导方针，要求广告内容在计算曝光量之前先开始加载。Server-side 跟踪工作流程有关完整序列，请参见。

信标类型	被解雇时	用途	计时详情
启动	当玩家开始呈现广告内容时	确认广告播放已实际开始	通常在第一个区段请求中与展示信标同时触发，但代表广告呈现的实际开始。这种区别对于分别跟踪曝光和启动事件的验证服务非常重要。
第一四分位数	当玩家达到广告时长的 25% 时	衡量广告第一季度的持续广告观看情况	当玩家请求包含广告时长 25% 点的区段时触发。例如，在包含 2 秒区段的 20 秒广告中，这通常会根据第 3 段的请求触发（广告开始大约 4-6 秒时）。
中点	当玩家达到广告时长的 50% 时	衡量一半广告持续观看广告的情况	当玩家请求包含广告时长 50% 的区段时触发。例如，在包含 2 秒区段的 20 秒广告中，这通常会根据第 5 段的请求触发（广告开始大约 8-10 秒时）。
第三四分位数	当玩家达到广告时长的 75% 时	衡量四分之三广告持续观看广告的情况	当玩家请求包含广告时长 75% 点的区段时触发。例如，在包含 2 秒区段的 20 秒广告中，这通常会根据第 8 段的请求触发（广告开始大约 14-16 秒时）。
完成	当玩家到达广告的结尾时	确认整则广告已交付给观众	当玩家请求广告的最后一部分时触发。这表明观众可能已经看到了整个广告内容。例如，在包含 2 秒区段的 20 秒广告中，这通常会在第 10 个区段的请求时触发（广告开始大约 18-20 秒时）。

> **Note**
>
> 信标发射的确切时间取决于区段持续时间和广告长度。 MediaTailor 根据特定的广告时长和区段结构计算与每个四分位数位置相对应的相应区段请求。

Server-side 跟踪时间和缓存行为

在服务器端报告中，根据玩家的实际区段请求 MediaTailor 触发跟踪事件，而不是根据清单解析或预加载活动触发跟踪事件。这种方法可确保精确的曝光量统计，符合视频广告衡量的行业标准。

关键时机原则

MediaTailor 服务器端跟踪遵循以下基本定时原则：

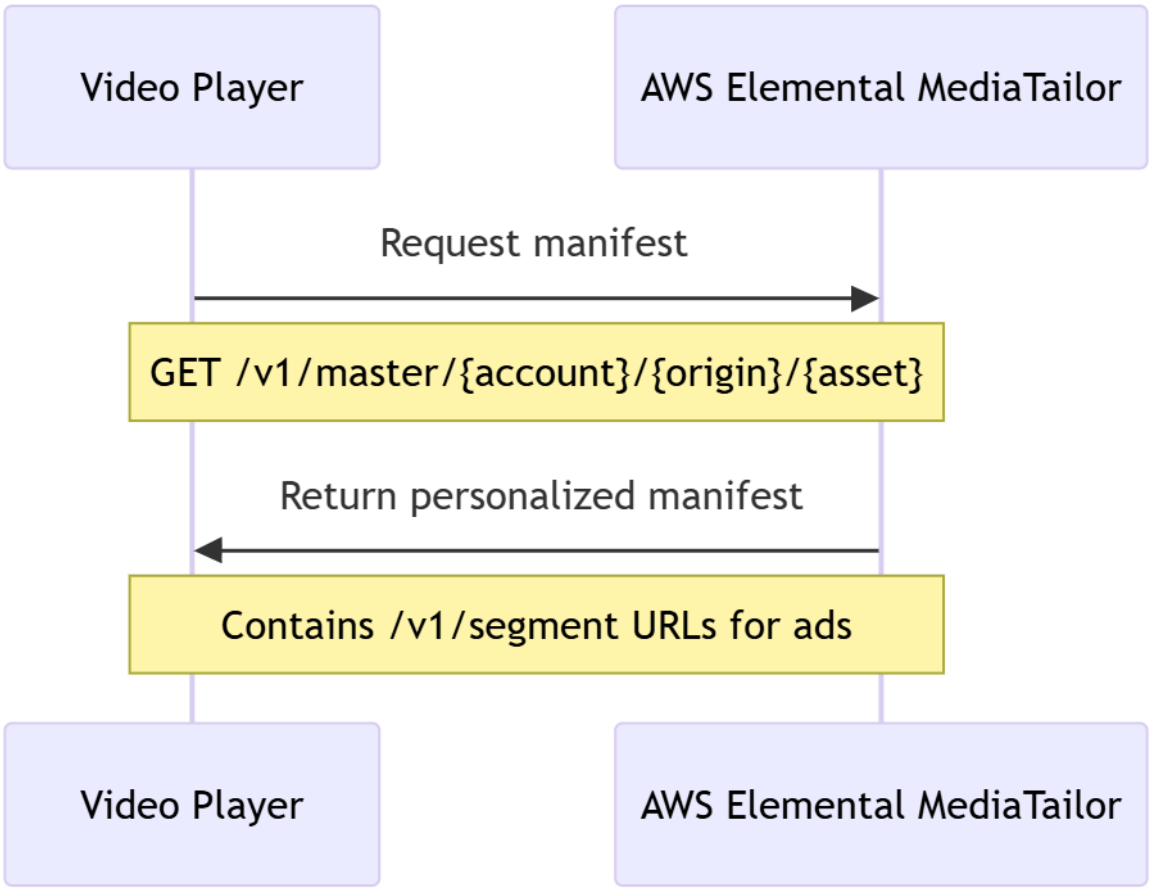
- 跟踪事件会根据实际的区段请求触发 ——信标仅在玩家向/v1/segment网址发出 HTTP 请求时发送，而不是在清单解析或缓存期间发送。
- 玩家缓存和预加载清单不会触发事件 -玩家可以在不生成任何跟踪事件的情况下解析、缓存或预加载清单信息。
- 区段预提取将触发事件 ——如果玩家在播放前预提取实际广告片段，则遵循行业标准行为，即区段请求构成有效曝光量。
- 每个/ v1/segment 请求都会触发相应的信标 -特定的跟踪事件（展示次数、四分位数、完成次数）由所请求的广告位置和区段决定。
- 时机符合IAB标准 ——该方法遵循互动广告局关于视频广告衡量和展示次数计算的指导方针。

Server-side 跟踪工作流程

下图说明了完整的服务器端跟踪工作流程，显示了何时触发了与玩家请求相关的跟踪事件：

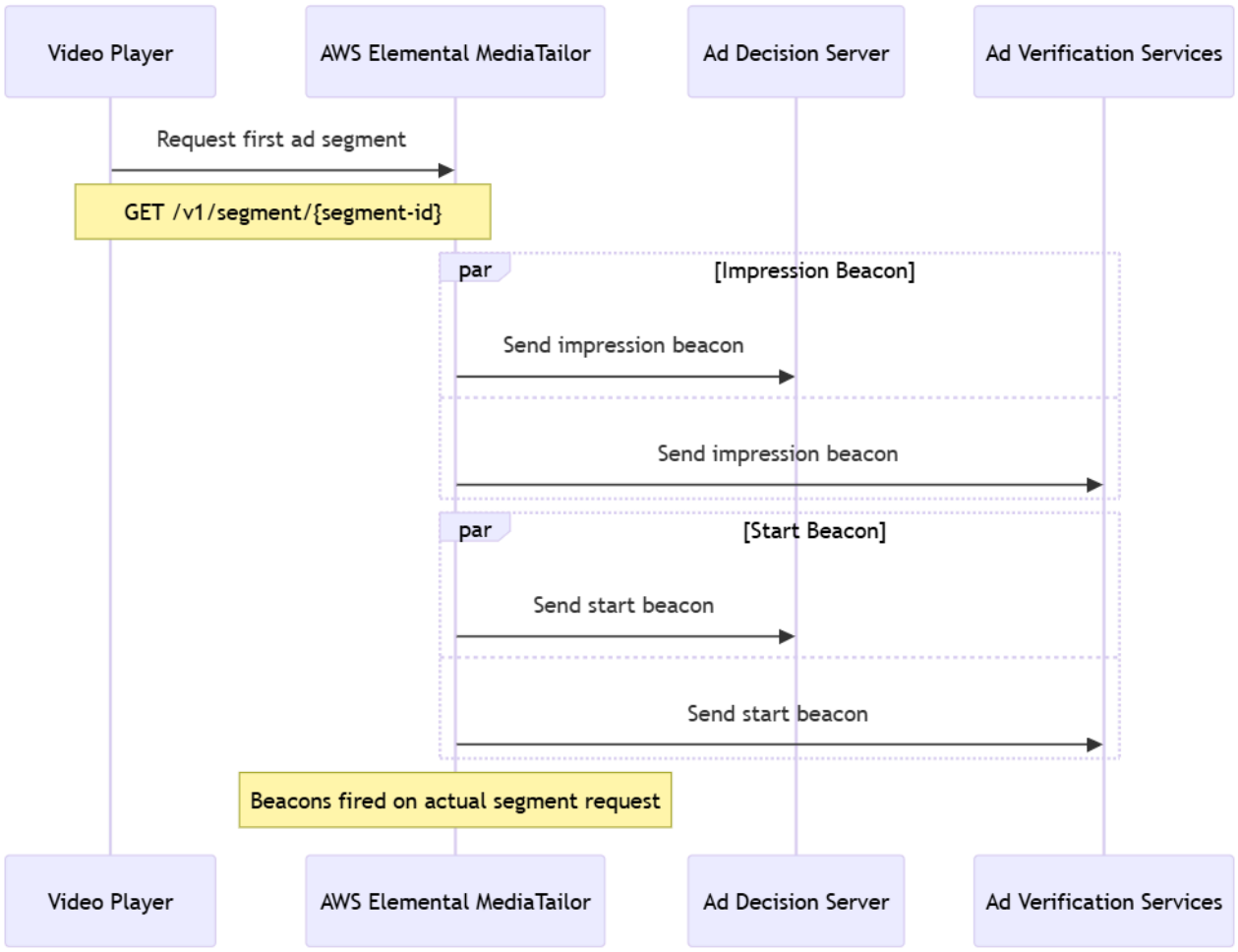
阶段 1：会话初始化

玩家向其请求清单 MediaTailor，该清单将返回包含广告区段网址的个性化清单：



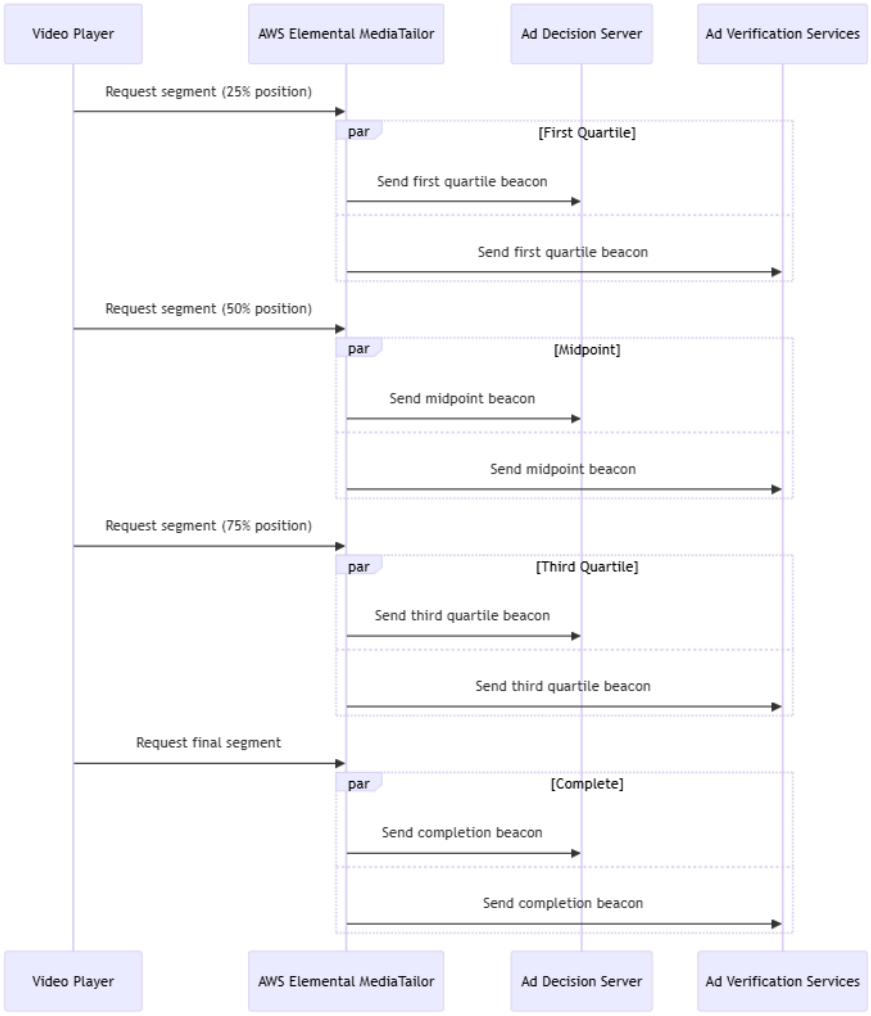
第 2 阶段：广告请求和曝光量跟踪

当玩家请求第一个广告区段时， MediaTailor 触发曝光量并启动向广告决策服务器和广告验证服务的信标：



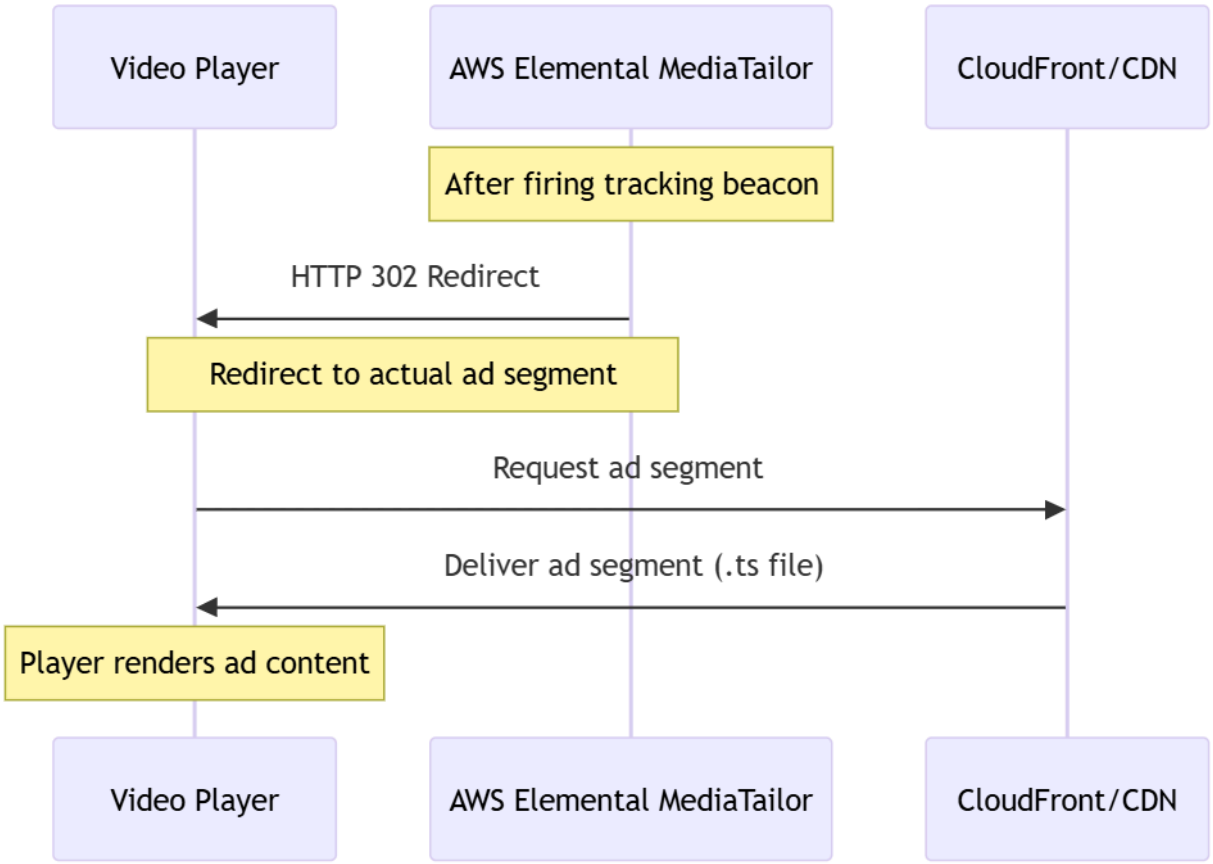
第 3 阶段：四分位数跟踪

MediaTailor 根据后续分段请求发射四分位信标（第一四分位数、中点、第三四分位数、完成）：



第 4 阶段：区段交付

触发跟踪信标后， MediaTailor 重定向到来自亚马逊 CloudFront 或您的 CDN 的实际广告片段：



服务器端跟踪工作流程包括以下关键计时行为：

1. 会话初始化 -玩家向其请求清单 MediaTailor。 MediaTailor 返回包含广告区段网址和/v1/segment路径的个性化清单。
2. 清单解析和缓存 -玩家解析清单，并可能预加载或缓存区段信息。无论玩家的缓存行为如何，在此阶段都不会触发任何跟踪事件。
3. 广告区段请求和展示量跟踪 -当玩家实际请求第一个广告区段（通常用于播放）时， MediaTailor 触发曝光信标并开始向广告决策服务器和广告验证服务跟踪事件。这发生在对 /v1/segment URL 的实际 HTTP 请求上，而不是在解析清单时。
4. 基于区段请求的四分位数跟踪 -根据与广告持续时间内计算出的四分位数位置相对应的后续区段请求，向广告决策服务器和广告验证服务 MediaTailor 触发四分位信标（第一四分位数、中点、第三四分位数、完成）。

5. 区段投放-触发相应的跟踪信标后， MediaTailor 发出 HTTP 重定向到实际广告段（来自亚马逊 CloudFront 或您的 CDN）。

玩家缓存和预加载注意事项

MediaTailor 服务器端跟踪旨在与各种玩家缓存和预加载策略兼容，同时保持精确的印象测量：

- 清单预加载 -预加载或缓存清单信息的玩家不会触发追踪事件。跟踪事件仅在发出实际区段请求时触发。
- 区段预取 -如果玩家在播放前预提取广告片段，则跟踪事件将在请求这些区段时触发，可能早于实际播放时间。这种行为符合将区段请求视为有效曝光量的行业标准。
- 玩家缓冲 -标准玩家缓冲行为（稍早于回放时请求片段）将根据区段请求模式在适当的时间触发跟踪事件。

对跟踪差异进行故障排除

如果您发现 MediaTailor 服务器端跟踪和第三方指标之间存在差异，请考虑以下因素：

- 玩家行为差异 -不同的玩家可能有不同的预提取和缓冲策略，这些策略会影响分段请求的提出时间。
- 网络状况 -网络状况不佳可能会导致玩家多次请求分段，或以不同于预期的时间间隔请求分段。
- CDN 配置 -错误的 CDN `/v1/segment` 请求缓存可能导致错过或重复的跟踪事件。
- 会话管理 -确保每个播放会话使用唯一的会话标识符，以防止跟踪事件冲突。

有关详细的故障排除指南，请参阅排查常见 问题。

MediaTailor 服务器端跟踪特性和功能

AWS Elemental MediaTailor 自动应用这些集成的服务器端跟踪功能，以优化广告测量的准确性和可靠性。该系统可防止重复信标，在高流量期间管理流量，保持正确的事件顺序，并提供全面的性能监控，无需您进行任何配置。您只需要确保您的广告决策服务器 (ADS) 在 VAST 响应中提供跟踪信标即可。

> **Note**
>
> 从 2025 年 9 月 30 日起，这些功能可供新客户使用。作为持续服务改进的一部分，现有客户将在2025年全年获得访问权限。如果您想立即访问这些功能，请联系 AWS 支持。

> **Note**
>
> 这些功能适用于拼接 (SSAI) 和服务器引导 (SGAI) 广告插入方法。两种模式下的信标类型和时间相同。它们在 MediaTailor 触发信标的方式上有所不同—— Server-side 使用服务器引导的广告插入 (SGAI) 进行跟踪 有关SGAI服务器端信标的详细信息，请参见。

信标重复数据删除

MediaTailor 防止在相同的广告事件中重复触发信标。服务器端跟踪系统仅在每个广告查看会话中发送一次曝光量、四分位数和完成信标一次。当视频播放器由于网络状况、比特率变化或缓冲策略而多次请求相同的广告段时，会 MediaTailor 跟踪发射的信标并阻止冗余传输。

重复数据删除可自动解决导致信标数量膨胀的常见情况：

- 自适应比特率直播 -当玩家下载同一广告段的不同质量变体时
- 网络重试场景 -当玩家因网络问题或超时而重新请求分段时
- 玩家缓冲策略 -当玩家出于缓冲目的预取或重新提取片段时

该系统旨在仅发射一次印象信标，即使玩家在不同的质量等级之间切换也是如此。

自适应节流和信标重试

MediaTailor 根据服务器响应指示器自动管理信标流量速率。系统监控 HTTP 响应模式、连接超时和错误代码以检测拥塞，然后相应地调整流量速率。当系统识别出服务器压力指标时，它会降低受影响域的流量速率，并在服务器显示出容量提高时自动提高流量。

系统使用以下指标监控服务器运行状况：

- HTTP 连接超时 -当测量平台未在预期的时间范围内做出响应时
- 错误响应代码 -表示服务器过载的 503、504 和 507 响应。您的广告服务器还必须支持这些错误代码才能完全兼容。
- 响应模式 -显示容量问题的测量平台性能变化

重试行为会自动尝试配送，最长可达 1 小时，每次尝试之间至少延迟 30 秒。无法配置此重试行为。

每秒信标流量管理

您可以设置 TPS 限制来控制信标传送率。这是服务器端跟踪功能的唯一可配置设置。 Account-level 限制向所有评估合作伙伴发送的广告跟踪请求总数的上限。 MediaTailor 强制执行 10,000 的最低 TPS 限制，以便为企业级运营提供足够的容量。

提交 AWS 支持请求单，使用以下信息设定 TPS 限制：

- AWS 账户 ID -您的特定账户标识符
- 目标区域 -您想要应用 TPS 限制的 AWS 区域
- 所需的 TPS 阈值 -您所需的每秒交易限额（最低 10,000）

默认情况下，没有 TPS 限制。如果您的广告决策服务器 (ADS) 要求，您可以申请 TPS 限制，但该限制必须大于 10,000 TPS。 MediaTailor 不会超过您指定的限制，但不能保证持续的吞吐量达到该限制。您的广告决策服务器会告诉您它可以支持哪些 TPS 限制。

In-order 信标

MediaTailor 自动保持广告跟踪事件的顺序交付。即使发生网络问题、重试或流量管理，系统也会保留信标顺序。这样可以确保测量合作伙伴以正确的顺序接收事件，以便进行准确的分析。

该系统遵循标准的行业信标顺序：

1. 开始事件 -广告播放开始时触发
2. 前四分位赛事 -广告完成率为 25% 时触发
3. 中点事件 -广告完成率达到 50% 时触发
4. 第三四分位数事件 -广告完成率达到 75% 时触发
5. 完成事件 -广告结束时触发

这些功能会自动协同工作：

- 在节流期间保持信标以维持正常秩序
- 每个评估合作伙伴域都有单独的事件队列，以防止在费率调整期间出现中断
- 重复数据删除可跟踪事件类型和时间表位置，同时保持时间顺序

Client-side 广告追踪

使用 AWS Elemental MediaTailor 客户端跟踪 API，您可以在直播工作流程中加入广告插播期间的玩家控件。在客户端跟踪中，玩家或客户向广告决策服务器 (ADS) 和其他广告验证实体发出跟踪事件，例如曝光量和四分位数广告信标。这些事件既跟踪整体广告时段状态，也跟踪每个广告插播时间内的单个广告的可用性。了解有关曝光量和四分位数 (ADS) 和其他广告验证实体的更多信息。有关曝光量和四分位数广告信标的更多信息，请参阅。Client-side 信标有关 ADS 和其他广告验证实体的更多信息，请参阅Client-side 广告跟踪集成。

有关向 ADS 传递玩家参数和会话数据以进行客户端跟踪的信息，请参阅MediaTailor ADS 请求的玩家变量和。MediaTailor ADS 请求的会话变量

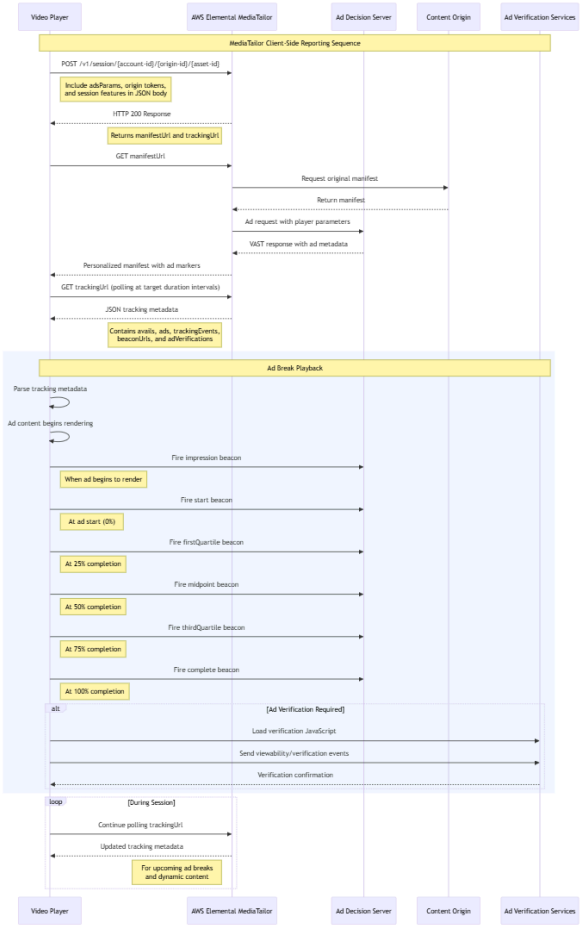
Client-side 跟踪启用了如下功能：

- Ad-break 倒计时器-有关更多信息，请参阅广告倒计时器。
- 广告点击率-有关更多信息，请参阅。广告点击率
- 展示配套广告-如需更多信息，请参阅配套广告。
- 可跳过的广告-有关更多信息，请参阅。可跳过的广告
- 显示符合隐私要求的 VAST 图标-有关更多信息，请参阅Google Why this Ad (WTA) 的图标。
- 控制广告期间的玩家清理情况-有关更多信息，请参阅擦洗。

使用 MediaTailor 客户端跟踪 API，您可以向播放设备发送元数据，该设备除了支持客户端跟踪之外还能启用其他功能：

Client-side 报告工作流程

下图显示了从会话初始化到广告播放和信标的完整客户端报告工作流程：



客户端报告工作流程包括以下步骤：

1. 会话初始化 -视频播放器向会 MediaTailor 话端点发送 POST 请求，其中包含 JSON 元数据`adsParams`，包括原始令牌和会话功能。 MediaTailor 使用`manifestUrl`和对会话`trackingUrl`进行响应。
2. 清单请求和广告决定 -玩家向其请求个性化清单 MediaTailor。 MediaTailor 从来源请求原始内容清单，使用玩家参数向广告决策服务器 (ADS) 发出广告请求，接收包含广告元数据的 VAST 响应，并向玩家提供带有广告标记的个性化清单。
3. 追踪数据检索 -玩家定期轮询跟踪网址（与 HLS 的目标持续时间或 DASH 的最小更新周期相匹配）。 MediaTailor 返回包含可用性、广告、跟踪事件、信标网址和广告验证数据的 JSON 跟踪元数据。
4. 广告播放和信标 -在广告休息期间，玩家解析跟踪元数据，在广告开始呈现时触发曝光信标，在适当的时间触发四分位信标（开始、firstQuartile、中点、第三四分位数、完成），必要时加载和执行广告验证，并将事件发送到第三方验证服务。 JavaScript viewability/verification
5. 持续投票 -玩家在整个会话中继续轮询跟踪网址，以接收即将到来的广告插播时间和动态内容的更新元数据。

该工作流程支持广告倒计时器、点击功能、伴随广告、可跳过广告和 VAST 图标显示等高级功能，以实现隐私合规。

主题

- 启用客户端跟踪
- 广告服务器参数
- Origin 交互查询参数
- Session-configured features
- 客户端跟踪的最佳实践
- Client-side 广告跟踪架构和属性
- Ad-tracking 活动时机
- 用于跟踪客户端广告的玩家控件和功能
- Client-side 信标
- 带有服务器端广告信标的混合模式
- Client-side 广告跟踪集成
- 使用以下方式浏览广告信标 GetTracking

启用客户端跟踪

您可以为每个会话启用客户端跟踪。玩家POST向 MediaTailor 配置的会话初始化前缀端点创建 HTTP。或者，玩家可以发送额外的元数据， MediaTailor 以便在进行广告调用、调用源以获取清单以及在会话级别调用或禁用 MediaTailor功能时使用。

以下示例显示了 JSON 元数据的结构：

```
{
  "adsParams": {                  # 'adsParams' is case sensitive
    "param1": "value1",           # key is not case sensitive
    "param2": "value2",           # Values can contain spaces. For example, 'value 2'
 is an allowed value.
    },
  "origin_access_token":"abc123", # this is an example of a query parameter designated
 for the origin
  "overlayAvails":"on"            # 'overlayAvails' is case sensitive. This is an
 example of a feature that is enabled at the session level.
```

```
}
```

使用 MediaTailor 控制台或 API 配置 ADS 请求模板网址以引用这些参数。在以下示例中，`player_params.param1`是的玩家参数`param1`，`player_params.param2`是的玩家参数`param2`。

```
https://my.ads.com/path?param1=[player_params.param1]&param2=[player_params.param2]
```

广告服务器参数

在 JSON 结构的最顶层是一个 J `adsParams` SON 对象。该对象内部有 key/value 几对， MediaTailor 可以在所有会话请求中读取并发送到广告服务器。 MediaTailor 支持以下广告服务器：

- 谷歌广告管理器
- SpringServe
- FreeWheel
- Publica

Origin 交互查询参数

JSON 结构最顶层的任何预留 key/value 对，例如`adsParams`、`availSuppression`、`overlayAvails`和，都不会以查询参数的形式添加到原始请求网址中。向源发出的 MediaTailor 每个会话清单请求都包含这些查询参数。源会忽略无关的查询参数。例如， MediaTailor 可以使用这些 key/value 对向源发送访问令牌。

Session-configured features

使用会话初始化 JSON 结构来启用、禁用或覆盖`overlayAvails`、`availSuppression`和等 MediaTailor功能。`adSignaling`会话初始化期间传递的任何功能配置都会覆盖 MediaTailor 配置级别的设置。

> **Note**
>
> 会话初始化 MediaTailor 时提交给的元数据是不可变的，并且在会话期间无法添加其他元数据。使用 SCTE-35标记来携带会话期间发生变化的数据。有关更多信息，请参阅 MediaTailor ADS 请求的会话变量。

Example: 为 HLS 执行客户端广告跟踪

```
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: 为 DASH 执行客户端广告跟踪

```
POST mediatailorURL/v1/session/hashed-account-id/origin-id/asset-id.mpd

        {
            "adsParams": {
               "deviceType": "androidmobile",
               "uid": "xjhhddli-9189901-uic"
           }
        }
```

报告模式参数

通过在请求正文中加入reportingMode参数，可以在初始化会话时指定报告模式。此参数控制是 MediaTailor 对会话执行客户端还是服务器端广告跟踪。

- client-玩家执行广告跟踪并将信标发送到广告服务器。如果未指定，则reportingMode这是默认模式。
- server- MediaTailor 执行服务器端广告跟踪并将信标直接发送到广告服务器。

Example使用服务器端报告模式初始化会话

```
POST mediatailorURL/v1/session/hashed-account-id/origin-id/asset-id.m3u8

        {
            "adsParams": {
               "deviceType": "ipad",
               "uid": "abdgfdyei-2283004-ueu"
           },
           "reportingMode": "server"
```

```
        }
```

Example使用客户端报告模式初始化会话（显式）

```
POST mediatailorURL/v1/session/hashed-account-id/origin-id/asset-id.mpd

        {
            "adsParams": {
               "deviceType": "androidmobile",
               "uid": "xjhhddli-9189901-uic"
            },
            "reportingMode": "client"
        }
```

> **Note**
>
> 该reportingMode参数是在会话初始化时设置的，在会话期间无法更改。如果未指定，reportingMode则 MediaTailor 默认为客户端报告以保持向后兼容性。

成功的响应是200带有响应正文的 HTTP。正文包含一个带有manifestUrl和trackingUrl密钥的 JSON 对象。这些值是相对网址，玩家可以将其用于播放和广告事件跟踪目的。

```
{
  "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"
}
```

有关客户端跟踪架构的更多信息，请参阅。Client-side 广告跟踪架构和属性

客户端跟踪的最佳实践

本节概述了直播和视频点播（VOD） MediaTailor 工作流程中客户端跟踪的最佳实践。

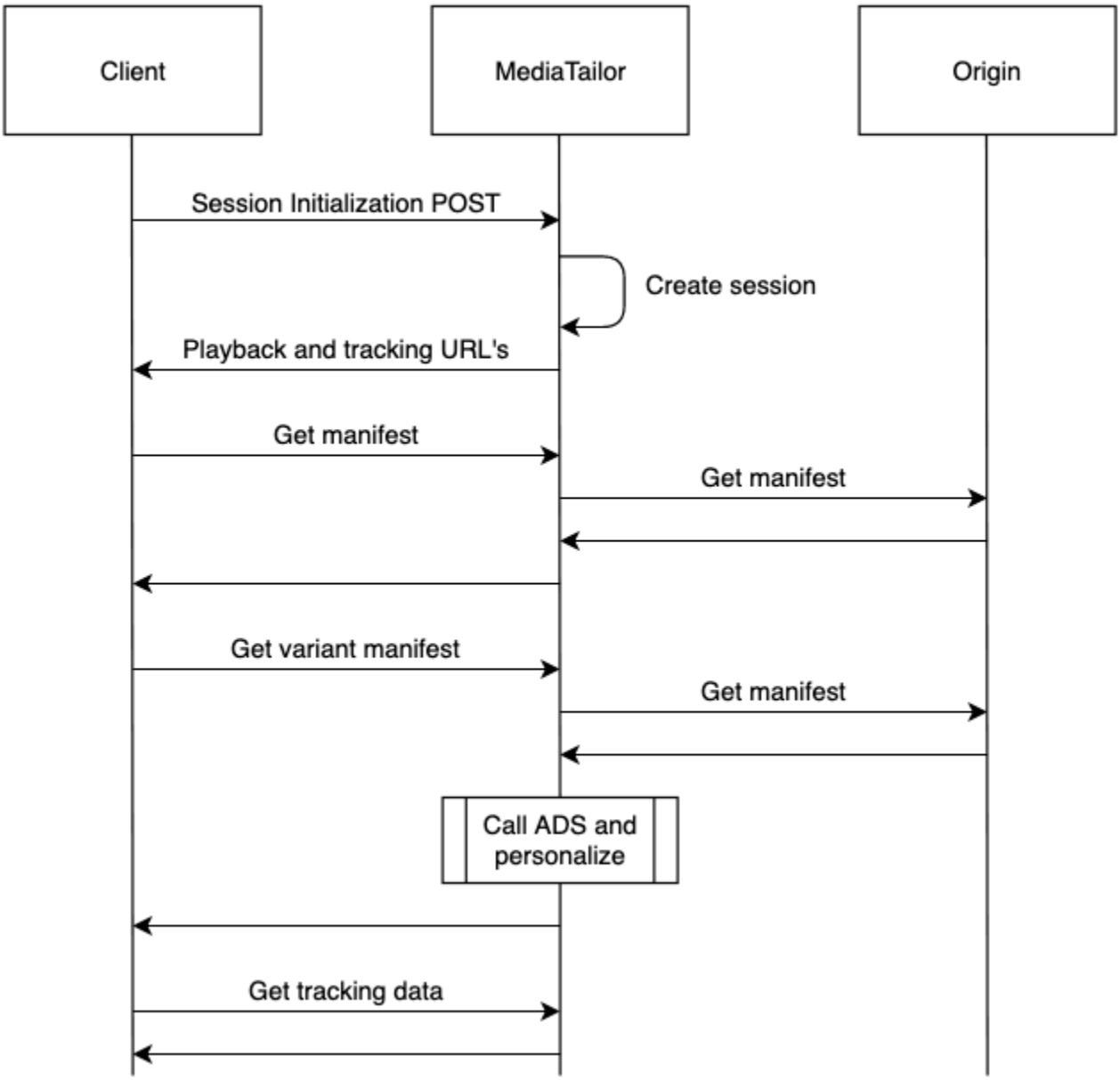
直播工作流程

按照与 HLS 的每个目标持续时间或 DASH 的最小更新周期相匹配的间隔对跟踪端点进行轮询，以便始终获得最新的广告跟踪元数据。在创意人员可能具有交互式或叠加组件的工作流程中，匹配此间隔尤为重要。

> **Note**
>
> 一些玩家支持事件监听器，可以将其用作投票的替代方案。例如，需要为每个会话启用 MediaTailor 广告 ID 装饰功能。有关更多信息，请参阅 广告 ID 装饰。使用此功能可在每个广告上添加日期范围 (HLS) 或事件元素 (DASH) 标识符。玩家可以使用这些清单标签作为提示来调用会话的 MediaTailor跟踪端点。

VOD 工作流程

成功初始化会话后，在 MediaTailor 收到第一个包含媒体的清单后，您只需要调用一次跟踪端点。



Server-guided 广告插入

Server-guided 广告插入 (SGAI) 会话不使用 `GetTracking` API。取而代之的是`aws.reportingMode=CLIENT`，当你使用时，当玩家请求广告内容时，会在每个资产列表响应的`TRACKING`部分中 MediaTailor提供跟踪信息。会话初始化响应不包括`trackingUrl`。

客户端跟踪的 SGAI 会话的资产清单响应具有以下结构：

```
{
  "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...
  }
}
```

在为 SGAI 方法实现客户端跟踪时：

- 从资产列表响应中解析该TRACKING部分，而不是调用 GetTracking
- 使用素材资源列表中提供的跟踪 URL 进行广告事件报告
- 根据玩家中的实际广告播放事件发射追踪信标
- 在提取资产列表时独立处理每个广告插播时间段的跟踪

> **Important**
>
> 该TRACKING部分仅在设置时包含在资产列表aws.reportingMode=CLIENT中。使用服务器端报告（SGAI 的默认设置）时，会省 MediaTailor 略该TRACKING部分，改为在广告 URI 中嵌入信标数据。有关更多信息，请参阅 Server-side 使用服务器引导的广告插入 (SGAI) 进行跟踪。

Client-side 广告跟踪架构和属性

借助 MediaTailor 客户端广告跟踪功能，您可以将详细的客户端广告跟踪数据集成到玩家环境中。以下部分涵盖了整体广告跟踪架构以及构成该架构的特定属性和值。

架构

以下 JSON 结构显示了 MediaTailor 客户端广告跟踪架构。此表示法说明了架构的嵌套结构，以帮助您了解不同部分之间的关系。

有关每个属性的详细信息，请参阅the section called “Properties”。

```
{
  "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": {}
}
```

Properties

下表列出了客户端跟踪 API 中的属性及其定义、值类型和示例。

属性	定义	值类型	示例
adID	• HLS-与广告开头相关的序列号 • DASH-广告的时段 ID 路径：/avails/ads/adId VAST 映射：无	字符串	10
adBreakTrackingEvents	承载来自 VAST 响应的 VMAP 跟踪事件的阵列。有关更多信息，请参阅 VMAP 1.0 规范的第 2.3.3 节。 路径：/avails/ads/adBreakTrackingEvents	数组	[]
adMarkerDuration	从清单中的广告标记中观察到的有效时长。	字符串	30
adParameters	来自 VAST VPAID 的一系列广告参数， MediaTailor 传递给玩家。 路径：/avails/ads/adParameters VAST 映射：VAST/Ad/InLine/Creatives/Creative/Linear/AdParameters	字符串	
adProgramDateTime	• HLS-以 ISO/IEC 8601:2004 格式表示广告的第一个媒体序列的日期。 • DASH-[未指定]	字符串	

属性	定义	值类型	示例
ads	一个包含构成漏洞的广告对象的数组。广告按其在清单中出现的顺序列出。 路径：/avails/ads	数组	[]
adSystem	投放广告的系统的名称。 **Important** 确保提供一个值。如果您不提供值，则可能会出现问题。	字符串	myADS
adTitle	广告的标题。	字符串	ad1
adVerifications	包含执行第三方测量代码以验证创意播放所需的资源和元数据。有关此属性的更多信息，请参阅 VAST 4.2 规范的第 3.16 节。 MediaTailor 支持adVerifications 作为 VAST 3 扩展节点。 路径：/avails/ads/adVerifications VAST 映射：VAST/Ad/InLine/AdVerifications	数组	[]
altText	配套广告图片的替代文字。此文本使支持描述性音频的玩家能够回读图像的描述。 路径：/avails/ads/companionAds/altText	字符串	video sequence advertising sneakers

属性	定义	值类型	示例
apiFramework	设置VPAID为告诉玩家此广告是 VPAID 广告。 可以出现在架构中的多个位置。	字符串	VPAID
availID	• HLS-与广告开头相关的序列号。 • DASH-广告利用率的时段 ID，通常是要替换为广告的内容的时段 ID。 路径：/avails/availID	字符串	• 34 • PT34S_1
avails	一个包含广告插播对象或空白对象的数组，这些对象显示在活动清单窗口中。可用项按其在清单中出现的顺序列出。 路径：/avails	数组	[]
adType	广告的类型。 路径：/avails/adType 和 /avails/ads/adType	字符串	
dateTime	以 ISO 8601 秒格式设置广告开始日期和时间。 路径：/avails/dateTime 和 /avails/ads/dateTime	字符串	
duration	长度，采用 ISO 8601 秒格式。响应包括整个广告利用率以及每个广告和信标的持续时间，尽管信标持续时间始终为零。 路径：/avails/duration 和 /avails/ads/duration	字符串	15.015

属性	定义	值类型	示例
durationInSeconds	长度，以秒为单位。 路径：/avails/durationInSeconds 和 /avails/ads/durationInSeconds	数字	
extensions	广告服务器使用的 VAST 的自定义扩展插件。有关扩展的更多信息，请参阅 V AST 4.2 规范的第 3.18 节。 路径：/avails/ads/extensions VAST 映射：VAST/Ad/InLine/Extensions	数组	[]
icons	广告的图标元素。 路径：/avails/ads/icons VAST 映射：VAST/Ad/InLine/Creatives/Creative/Linear/Icons	数组	
mediaFiles	玩家广告所需的视频和其他资源都可用。 路径：/avails/ads/mediaFiles	对象	
nonLinearAvails	非线性广告可用对象数组。 路径：/nonLinearAvails	数组	

属性	定义	值类型	示例
executableResource	用于验证的可执行资源。 路径：/avails/ads/adVerifications/executableResource VAST 映射：VAST/Ad/InLine/AdVerifications/Verification/ExecutableResource	数组	
javaScriptResource	JavaScript 用于验证的资源。 路径：/avails/ads/adVerifications/javaScriptResource VAST 映射：VAST/Ad/InLine/AdVerifications/Verification/JavaScriptResource	数组	
trackingEvents	跟踪验证广告或配套广告的事件。 路径：/avails/ads/adVerifications/trackingEvents 或 /avails/ads/companionAds/trackingEvents	数组	

属性	定义	值类型	示例
vendor	验证供应商。 路径：/avails/ads/adVerifications/vendor VAST 映射：VAST/Ad/InLine/AdVerifications/Verification/@vendor	字符串	
uri	指向可执行资产、视频资产或跟踪端点的 URI。 路径：架构中的不同位置 VAST 映射：VAST 中的各种 CDATA 元素	字符串	https://tracking.example.com/impression
verificationParameters	验证参数。 路径：/avails/ads/adVerifications/verificationParameters VAST 映射：VAST/Ad/InLine/AdVerifications/Verification/VerificationParameters	字符串	
attributes	配套广告属性，例如尺寸和渲染模式。 路径：/avails/ads/companionAds/attributes	对象	

属性	定义	值类型	示例
companionClickThrough	广告商页面的网址，当观众点击配套广告时，媒体播放器会打开该网址。 路径：/avails/ads/companionAds/companionClickThrough VAST 映射：VAST/Ad/InLine/Creatives/Creative/CompanionAds/Companion/CompanionClickThrough	字符串	https://aws.amazon.com/
companionClickTracking	该companionClickThrough 物业的追踪网址。 路径：/avails/ads/companionAds/companionClickTracking VAST 映射：VAST/Ad/InLine/Creatives/Creative/CompanionAds/Companion/CompanionClickTracking	字符串	https://myads.com/beaconing/event=clicktracking

属性	定义	值类型	示例
htmlResource	直接插入流媒体提供商的 CDATA-encoded HTML 页面中的 HTML。 路径：/avails/ads/companionAds/htmlResource VAST 映射：VAST/Ad/InLine/Creatives/Creative/CompanionAds/Companion/HTMLResource	字符串	<![CDATA[<!doctype html><html><head><meta name=\"viewport\" content=\"width=1, initial-scale=1.0, minimum-scale=1.0, ...]]>
iFrameResource	直播提供商加载到 iframe 中的 HTML 资源文件的网址。 路径：/avails/ads/companionAds/iFrameResource VAST 映射：VAST/Ad/InLine/Creatives/Creative/CompanionAds/Companion/iFrameResource	字符串	
sequence	在 VAST 响应中为广告素材指定的序列值。 路径：/avails/ads/companionAds/sequence	字符串	1

属性	定义	值类型	示例
startTime	时间位置，采用 ISO 8601 秒格式。对于 HLS 来说，这与播放会话的开始有关。对于 DASH，这与清单的 AST（可用性开始时间）有关。该响应包含整个广告效用以及每个广告和信标的开始时间。 路径：/avails/startTime 和 /avails/ads/startTime	字符串	PT18.581355S
startTimeInSeconds	时间位置，以秒为格式。对于 HLS 来说，这与播放会话的开始有关。对于 DASH，这与清单的 AST（可用性开始时间）有关。该响应包含整个广告效用以及每个广告和信标的开始时间。 路径：/avails/startTimeInSeconds 和 /avails/ads/startTimeInSeconds	数字	18.581
eventId	• HLS-与信标相关的序列号。 • DASH-广告ptsTime的开头。	字符串	
event	跟踪事件的名称。 路径：/avails/ads/adVerifications/trackingEvents/event 或 /avails/ads/companionAds/trackingEvents/event	字符串	impression，start，firstQuartile，midpoint，thirdQuartile，complete

属性	定义	值类型	示例
beaconUrls	MediaTailor 发送广告信标的网址。 路径：/avails/ads/mediaFiles/trackingEvents/beaconUrls	数组	
bitrate	视频资产的比特率。可执行资产通常不包含此属性。	字符串	2048
companionAds	一个或多个配套广告内容规范，每个规范都指定要使用的资源文件。伴随广告附带广告，并提供内容，例如广告周围的边框或横幅，以显示在视频附近。 路径：/avails/ads/companionAds	数组	[]
creativeId	广告Creative标签的Id属性值。	字符串	creative-1
creativeSequence	根据 VAST 响应中的Ad@id值，广告的播放顺序。	字符串	1
dashAvailabilityStartTime	对于 live/dynamic DASH 来说MPD@availabilityStartTime ，是起源清单。	字符串	2022-10-05T19:38:39.263Z
delivery	表示是否正在使用progressive 或streaming 协议。	字符串	progressive
eventType	信标的类型。 路径：/avails/ads/mediaFiles/trackingEvents/eventType	字符串	impression
height	视频资源的高度（以像素为单位）。	字符串	360

属性	定义	值类型	示例
hlsAnchorMediaSequenceNumber	在 HLS 起源清单中看到的 first/oldest 媒体序列的媒体序列号。	字符串	77
maintainAspectRatio	表示在缩放时是否保持视频的长宽比。	布尔值	true
mediaFilesList	指定玩家利用广告所需的视频和其他资产。 路径：/avails/ads/mediaFiles/mediaFilesList	数组	[]
mediaFileUri	指向可执行资产或视频资产的 URI。	字符串	https://myad.com/ad/ad134/vpaid.js
mediaType	素材资源或配套资源的 MIME 类型。	字符串	video/mp4
meta	广告的其他元数据。	对象	
mezzanine	夹层 MP4 素材资源的网址，如果 VPAID 广告中包含一个则指定。 路径：/avails/ads/mediaFiles/mezzanine	字符串	https://gcdn.2mdn.net/videoplayback/id/itag/ck2/file/file.mp4

属性	定义	值类型	示例
nextToken	指向下一页结果的代币的值（如果存在此类值）。	字符串	UFQzOS44NzNTXzIwMjMtMDctMzFUMTY6NTA6MDYuMzUwNjI2ODQ1Wl8x
nonLinearAds	Non-linear 与视频内容一起显示的广告。	数组	[]
nonLinearAdsList	非线性广告列表。	数组	[]
scalable	表示是否将视频缩放到其他尺寸。	布尔值	true
skipOffset	时间值，用于标识玩家何时向用户提供跳过控件。	字符串	00:00:05
staticResource	用于广告组件的静态广告素材文件的网址。 路径：/avails/ads/companionAds/staticResource	字符串	https://very-interactive-ads.com/campaign1/file.json?c=1019113602
vastAdId	Ad标签的Id属性值。	字符串	ad1
width	视频资源的宽度，以像素为单位。	字符串	640

属性	定义	值类型	示例
xPosition	图标在视频播放器中的水平位置。可以是特定的像素值或诸如 “左” 或 “右” 之类的位置。 路径：/avails/ads/icons/attributes/xPosition	字符串	left 或 10
yPosition	图标在视频播放器中的垂直位置。可以是特定的像素值或诸如 “顶部” 或 “底部” 之类的位置。 路径：/avails/ads/icons/attributes/yPosition	字符串	top 或 10
iconClicks	包含图标的点击和跟踪信息。 路径：/avails/ads/icons/iconClicks	对象	
iconClickThrough	广告商页面的网址，当观众点击该图标时，媒体播放器会打开该网址。 路径：/avails/ads/icons/iconClicks/iconClickThrough	字符串	https://advertiser.com/landing-page
iconClickTracking	该iconClickThrough 物业的追踪网址。 路径：/avails/ads/icons/iconClicks/iconClickTracking	对象	

属性	定义	值类型	示例
iconClickFallbackImages	无法显示图标时显示的备用图像数组。 路径：/avails/ads/icons/iconClicks/iconClickFallbackImages	数组	
iconViewTracking	用于跟踪查看图标时的 URL。 路径：/avails/ads/icons/iconViewTracking	字符串	https://tracking.example.com/icon-view
offset	广告播放期间图标应出现的时间间隔。 路径：/avails/ads/icons/attributes/offset	字符串	00:00:05
program	与图标相关的计划或计划，例如“AdChoices”。 路径：/avails/ads/icons/attributes/program	字符串	AdChoices
pxratio	图标或配套广告的像素比例，用于高 DPI 显示屏。 路径：/avails/ads/icons/attributes/pxratio 或 /avails/ads/companionAds/attributes/pxratio	字符串	1 或 2

属性	定义	值类型	示例
type	资源或扩展的类型。 路径：/avails/ads/extensions/type 或 /avails/ads/adVerifications/executableResource/type	字符串	text/javascript
content	扩展的内容。 路径：/avails/ads/extensions/content	字符串	
language	可执行资源的编程语言。 路径：/avails/ads/adVerifications/executableResource/language	字符串	javascript
browserOptional	表示 JavaScript 资源是否需要浏览器支持。 路径：/avails/ads/adVerifications/javaScriptResource/browserOptional	字符串	true 或 false
id	架构中各种元素的标识符。 路径：/avails/ads/companionAds/attributes/id 或 /avails/ads/icons/iconClicks/iconClickTracking/id	字符串	companion-1

属性	定义	值类型	示例
assetHeight	配套广告素材的高度。 路径：/avails/ads/companionAds/attributes/assetHeight	字符串	250
assetWidth	配套广告素材资源的宽度。 路径：/avails/ads/companionAds/attributes/assetWidth	字符串	300
expandedHeight	随播广告展开时的高度。 路径：/avails/ads/companionAds/attributes/expandedHeight	字符串	600
expandedWidth	展出后的配套广告的宽度。 路径：/avails/ads/companionAds/attributes/expandedWidth	字符串	600
renderingMode	配套广告的渲染模式。 路径：/avails/ads/companionAds/attributes/renderingMode	字符串	default 或 transparent
adSlotId	应在其中展示配套广告的广告位的 ID。 路径：/avails/ads/companionAds/attributes/adSlotId	字符串	banner-1

属性	定义	值类型	示例
creativeType	创意资源的 MIME 类型。 路径：/avails/ads/icons/staticResource/creativeType	字符串	image/png

Ad-tracking 活动时机

通过客户端报告，玩家必须发出一定精度的跟踪事件（信标）。使用 MediaTailor 客户端跟踪架构，您可以确保广告、伴随广告、叠加和跟踪事件、时间和持续时间信息以不同的形式存在，无论何种用处。

使用以下 MediaTailor key/value 配对让玩家准确协调广告事件活动（例如追踪事件）与回放位置：

- startTime
- startTimeInSeconds
- adProgramDateTime
- adID/eventId

HLS 和 DASH 的startTime价值实现startTimeInSeconds方式不同：

- HLS-这些startTime值相对于播放会话的开始。播放会话的开始被定义为零时间。广告startTime是所有EXT-INF区段持续时间累积值的总和，直到有效。广告或跟踪事件所在区段的媒体序列号也对应于客户端跟踪eventId响应中的adId或。
- DASH：
  - Live/dynamic 清单-这些startTime值是相对于 DASH 清单的。MPD@availabilityStartTimeMPD@avaibilityStartTime是所有消耗直播的 MediaTailor 会话的计时锚点。
  - VOD/static 清单-这些startTime值相对于播放会话的开始。播放会话的开始被定义为零时间。avail中的每个广告都包含在自己的Period元素中。该Period元素具有一个@start属性，其值与客户端跟踪startTime负载中的值相同。PeriodId也对应于客户端跟踪响应eventId中的adId或。

Example HLS：

在以下示例中， MediaTailor 会话已启动，以下清单是第一个提供给客户端的清单：

```
#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
```

在客户端跟踪 JSON 负载中，以下值适用：

- `startTime`: `"PT18.581355S"`
- `startTimeInSeconds`: `18.581`
- `availProgramDateTime`: `"2023-05-03T21:24:41.453Z"`
- `adId`: `4603269`

Example DASH：

在以下示例中， MediaTailor 会话在清单中出现了中间部分。请注意，第二个时段（即广告时段）的`@start`属性值是相对于该`MPD@availabilityStartTime`值的。该值是 MediaTailor 写入所有会话的客户端跟踪响应`startTime`字段的值。

```
<?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/eaa5485198bf497284559efb8172425e/</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>
```

在客户端跟踪 JSON 负载中，以下值适用：

- `startTime`: `"PT5042H27M59.931S"`
- `startTimeInSeconds`: `18152879.931`
- `availProgramDateTime`: *`null`*
- `adId`: `1683151599194_1_1`

用于跟踪客户端广告的玩家控件和功能

MediaTailor 客户端跟踪元数据支持各种玩家控件和功能。以下列表描述了流行的玩家控件。

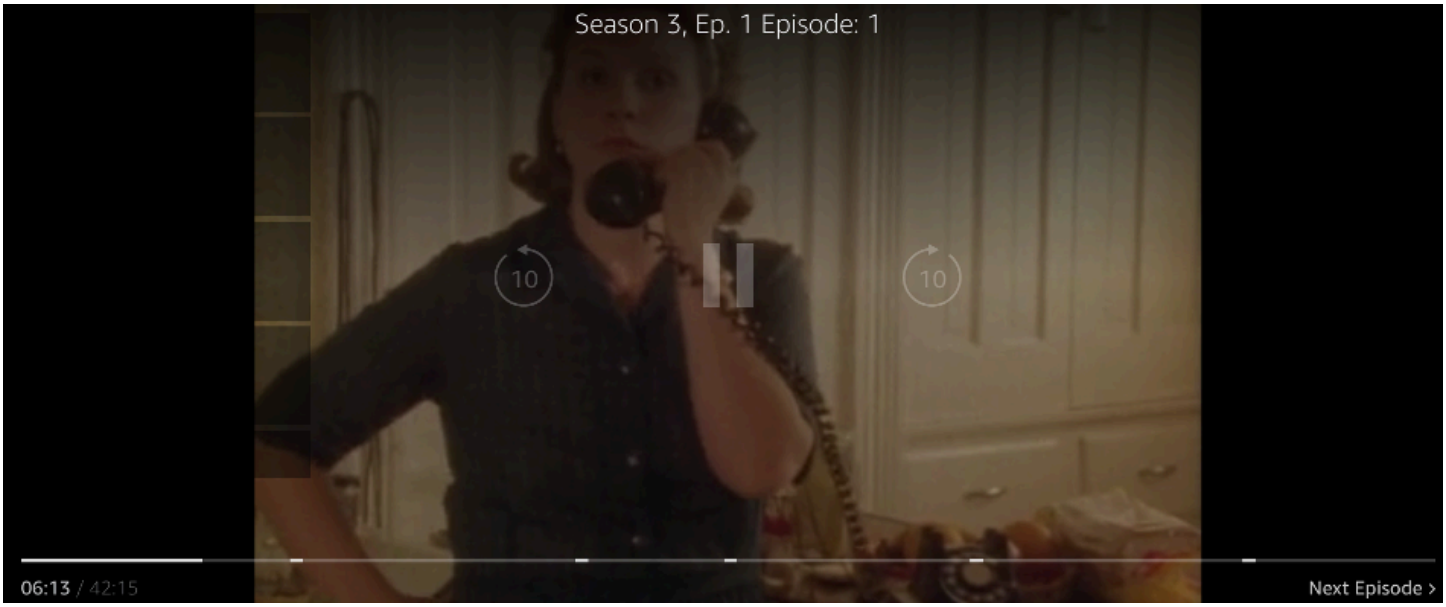
主题

- 擦洗
- 广告倒计时器
- 可跳过的广告
- 广告点击率
- 配套广告
- 互动广告 (SIMID)
- 互动广告 (VPAID)
- Google Why this Ad (WTA) 的图标

擦洗

为了增强播放体验，播放器可以在播放时间轴中显示广告位置。 MediaTailor 在客户端跟踪响应中以`startTimeInSeconds`值的形式提供这些广告位置。

> **Note**
>
> 一些流媒体提供商会阻止清理广告位置。

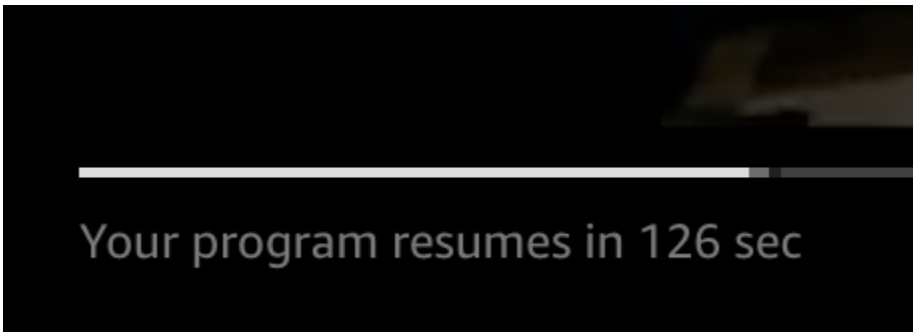


以下客户端跟踪负载 JSON 响应显示了 avails 数组的根 JSON 对象内的空闲时间（广告中断）开始时间。玩家使用这些数据在玩家时间轴上显示广告时段的位置，即 28 秒。


```
{
  "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": []
}
```

广告倒计时器

借助，MediaTailor 您可以使用广告倒计时器来帮助您的受众在广告插播观看期间保持参与度。观众可以使用计时器来了解广告中断时间何时结束以及他们的节目何时恢复。



客户端跟踪元数据中在广告倒计时器中起作用的元素是`startTime`、`startTimeInSeconds`、`duration`和`durationInSeconds`。玩家使用这些元数据以及单独跟踪的会话经过的时间来确定何时显示计时器以及应该从哪个值开始倒计时。

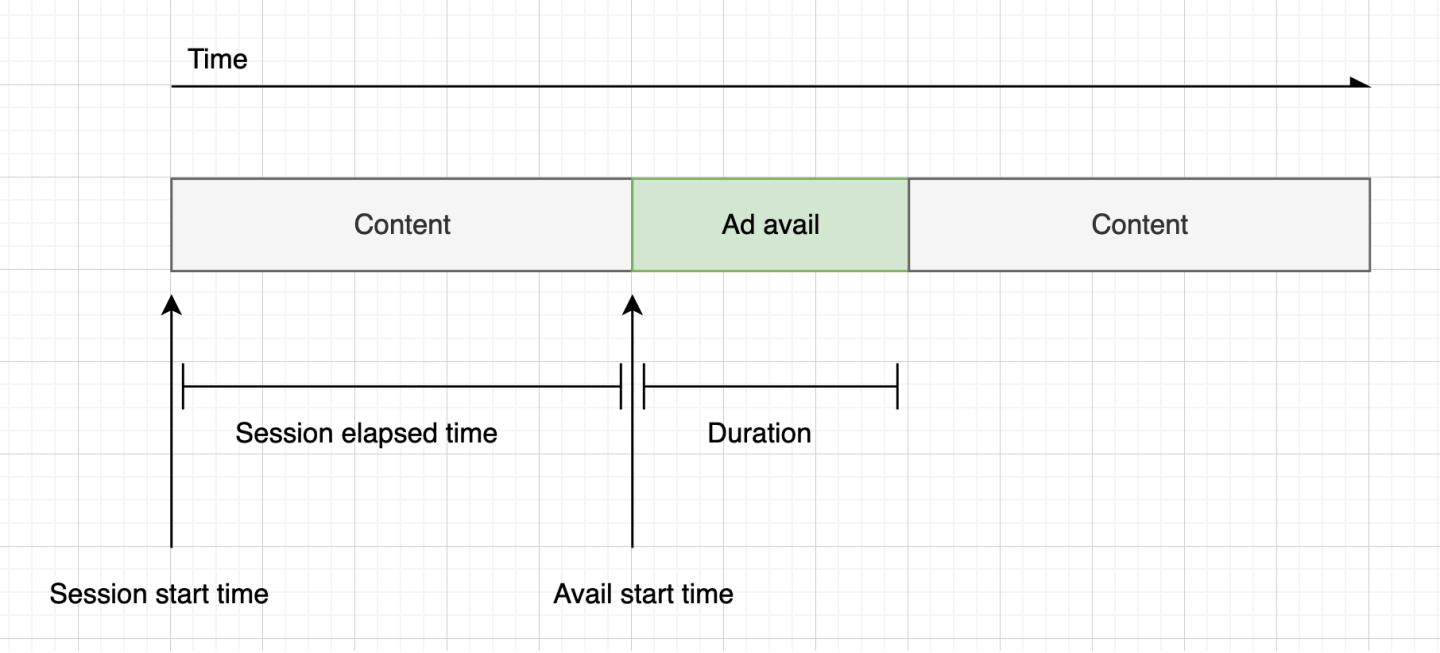
以下客户端跟踪负载 JSON 响应显示广告倒计时器所需的信息。

```
{
  "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": []
}
```

当会话的经过时间达到无效的开始时间时，玩家会显示一个倒数计时器，其值与该无效的持续时间相匹配。随着经过的时间超过无效的开始时间，倒计时计时器的值会降低。

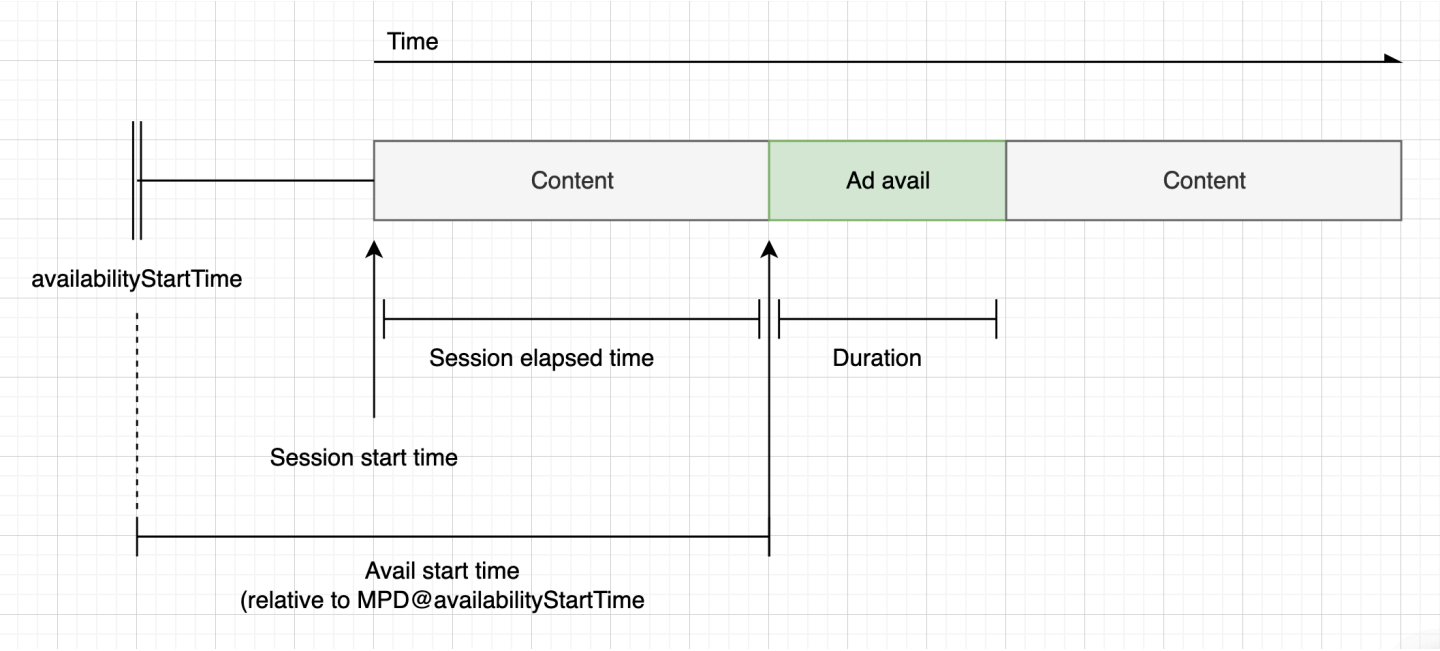
Example公式：HLS（直播和 VOD）和 DASH（VOD）的倒计时器

- session_start_time= 所有EXT-INF持续时间值的总和-三个最新EXT-INF媒体序列的持续时间值
- 计时器值 = duration-(session_elapsed_time-startTime)



Example公式：DASH 的倒计时器（直播）

- session_start_time=（最新区段的 startTime +duration）/timescale-MPD@suggestedPresentationDelay
- 计时器值 = duration-(session_elapsed_time-startTime)



可跳过的广告

可跳过的广告是允许观众跳过部分广告以恢复观看节目的广告位。在 VAST 中，该`Linear@skipOffset`属性用于标识可跳过的广告。

以下 VAST 响应显示了如何使用可跳过的广告：

```
<?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>
```

以下客户端跟踪负载 JSON 响应显示了数组内的ads广告元数据。该数组包含从 VAST 响应中 MediaTailor获得的skipOffset值。

```
{
  "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": []
}
```


广告点击率

Click-through URI 允许广告商衡量广告在吸引观众注意力方面的成功程度。观众点击正在播放的广告的活跃视频帧后，网络浏览器会打开广告商主页或活动登录页面的 URI。玩家开发者决定点击行为，例如在广告视频上叠加按钮或标签，并提供一条消息供点击以了解更多信息。玩家开发者通常会在观众点击活动视频帧后暂停广告的视频。

MediaTailor 可以解析并提供 VAST 响应中返回的任何线性视频点击事件网址。以下 VAST 响应显示了广告点击率示例。

```
<?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>
```

以下客户端跟踪负载 JSON 响应显示了如何在数 MediaTailor 组内显示点击和点击跟踪网址。`trackingEvents``clickThrough`事件类型代表点击广告，`clickTracking`事件类型代表点击跟踪网址。

```
{
  "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": "UFQzOS4zMzlTXzIwMjMtMDctMzFUMTY6NTQ6MDQuODA1Mzk2NTI5Wl8x",
  "nonLinearAvails": []
}
```

配套广告

配套广告出现在线性广告素材旁边。使用配套广告通过显示产品、徽标和品牌信息来提高广告位的效果。展示广告可以采用快速响应（QR）码和可点击区域，以提高受众参与度。

MediaTailor 在 VAST 响应中支持配套广告。它可以分别传递来自`StaticResource`、`iFrameResource`、和`HTMLResource`节点的元数据。

以下 VAST 响应显示了线性广告和配套广告的位置和格式示例。

```
<?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">
          <CcompanionAds>
            <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>
          </CcompanionAds>
        </Creative>
      </Creatives>
      ...
    </InLine>
  </Ad>
</VAST>
```

数据显示在列表的客户端跟踪响应中。/avail/x/ads/y/companionAds每个线性广告素材最多可以包含 6 个配套广告。如下例所示，配套广告显示在列表中

> **Note**
>
> 最佳做法是，应用程序开发人员应实现逻辑，在广告素材结束时明确移除或卸载配套广告。

```
{
  "avails": [
    {
```

```
      "adBreakTrackingEvents": [],
      "adMarkerDuration": null,
      "ads": [
        {
          "adId": "0",
          "adParameters": "",
          "adProgramDateTime": null,
          "adSystem": "EMT",
          "adTitle": "sample",
          "adVerifications": [],
          "companionAds": [
            {
              "adParameters": null,
              "altText": null,
              "attributes": {
                "adSlotId": null,
                "apiFramework": null,
                "assetHeight": null,
                "assetWidth": null,
                "expandedHeight": null,
                "expandedWidth": null,
                "height": "250",
                "id": "2",
                "pxratio": null,
                "renderingMode": null,
                "width": "300"
              },
              "companionClickThrough": "https://beacon.com/2",
              "companionClickTracking": null,
              "htmlResource": null,
              "iFrameResource": null,
              "sequence": "1",
              "staticResource": "https://emt.com/companion/9973499273",
              "trackingEvents": [
                {
                  "beaconUrls": [
                    "https://beacon.com/1"
                  ],
                  "eventType": "creativeView"
                }
              ]
            },
            {
              "adParameters": null,
```

```
            "altText": null,
            "attributes": {
              "adSlotId": null,
              "apiFramework": null,
              "assetHeight": null,
              "assetWidth": null,
              "expandedHeight": null,
              "expandedWidth": null,
              "height": "90",
              "id": "3",
              "pxratio": null,
              "renderingMode": null,
              "width": "728"
            },
            "companionClickThrough": "https://beacon.com/4",
            "companionClickTracking": null,
            "htmlResource": null,
            "iFrameResource": null,
            "sequence": "1",
            "staticResource": "https://emt.com/companion/1238901823",
            "trackingEvents": [
              {
                "beaconUrls": [
                  "https://beacon.com/3"
                ],
                "eventType": "creativeView"
              }
            ]
          }
        ],
        "creativeId": "1",
        "creativeSequence": "1",
        "duration": "PT10S",
        "durationInSeconds": 10,
        "extensions": [],
        "mediaFiles": {
          "mediaFilesList": [],
          "mezzanine": ""
        },
        "skipOffset": null,
        "startTime": "PT0S",
        "startTimeInSeconds": 0,
        "trackingEvents": [
          {
```

```
              "beaconUrls": [
                "https://beacon.com/impression/1"
              ],
              "duration": "PT10S",
              "durationInSeconds": 10,
              "eventId": "0",
              "eventProgramDateTime": null,
              "eventType": "impression",
              "startTime": "PT0S",
              "startTimeInSeconds": 0
            }
          ],
          "vastAdId": ""
        }
      ],
      "availId": "0",
      "availProgramDateTime": null,
      "duration": "PT10S",
      "durationInSeconds": 10,
      "meta": null,
      "nonLinearAdsList": [],
      "startTime": "PT0S",
      "startTimeInSeconds": 0
    }
  ],
  "dashAvailabilityStartTime": null,
  "hlsAnchorMediaSequenceNumber": null,
  "nextToken": "UFQxMFNfMjAyMy0wNy0wNlQyMToxMDowOC42NzQ4NDA1NjJaXzE%3D",
  "nonLinearAvails": []
}
```

互动广告 (SIMID)

SecureInteractive 媒体接口定义 (SIMID) 是互动广告局 (IAB) 在 VAST 4.x 标准中引入的互动广告标准。SIMID 将交互元素的加载与玩家身上的主要线性创作分离，在 VAST 响应中同时引用了这两个元素。 MediaTailor 在主要广告素材中拼接以保持播放体验，并将交互式组件的元数据放入客户端跟踪响应中。

在以下示例 VAST 4 响应中，SIMID 有效载荷位于`InteractiveCreativeFile`节点内部。

```
<?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>
```

在以下 VAST 3 响应中，SIMID 有效载荷位于`Extensions`节点内部。

```
<?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>
```

在以下客户端跟踪响应中，SIMID 数据显示在列表中。/avails/x/ads/y/extensions

```
{
  "avails": [
    {
      "adBreakTrackingEvents": [],
```

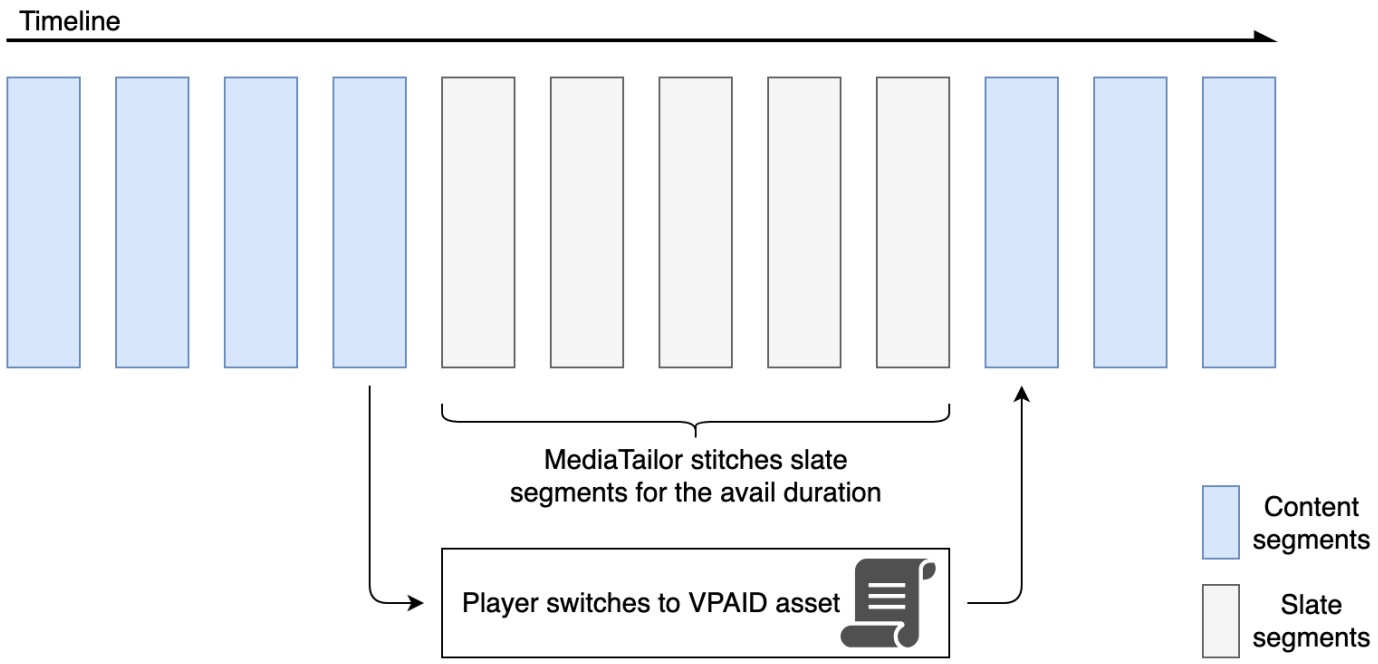
```
      "adMarkerDuration": null,
      "ads": [
        {
          "adId": "1",
          "adParameters": "",
          "adProgramDateTime": "2023-07-31T16:53:40.577Z",
          "adSystem": "2.0",
          "adTitle": "Linear SIMID Example",
          "adVerifications": [],
          "companionAds": [],
          "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": "UFQzOS4zMzlTXzIwMjMtMDctMzFUMTY6NTQ6MDQuODA1Mzk2NTI5Wl8x",
 "nonLinearAvails": []
}
```

互动广告 (VPAID)

视频播放器广告接口定义 (VPAID) 指定了广告与视频播放器之间的协议，该协议可实现广告互动和其他功能。对于直播， MediaTailor 支持VPAID格式，方法是在使用期间拼接slate片段，并在视频播放器消耗的客户端跟踪响应中放置VPAID广告素材的元数据。播放器下载VPAID文件并播放线性创意并执行客户的脚本。玩家永远不应该玩石板片段。

> **Note**
>
> 从 VAST 4.1 开始，VPAID 已过时。



以下示例显示了 VAST 响应中的 VPAID 内容。

```
<?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>
```

以下示例显示了跟踪信息。


```
{
  "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": "UFQzOS4zMzlTXzIwMjMtMDctMzFUMTY6NTQ6MDQuODA1Mzk2NTI5Wl8x",
 "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": "UFQ4TTQyLjI4OVNfMjAyMy0wOC0xNFQxOTo1MDo0MS4zOTc5MjAzODVaXzE%3D",
  "nonLinearAvails": []
}
```


Google Why this Ad (WTA) 的图标

AdChoices是一项行业标准，为观众提供有关他们看到的广告的信息，包括这些广告是如何针对他们的。

MediaTailor 客户端跟踪 API 支持 VAST 响应的 VAST 扩展节点中携带的图标元数据。有关 VAST 响应中 WTA 的更多信息，请参阅此 VAST XML 响应示例。

> **Note**
>
> MediaTailor 目前仅支持 VAST 版本 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>
```

以下示例显示了列表中的客户端跟踪响应。/avails/x/ads/y/extensions

```
{
  "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": "UFQxMFNfMjAyMy0wNy0wNlQyMDo0MToxNy45NDE4MDM0NDhaXzE%3D",
  "nonLinearAvails": []
}
```


Client-side 信标

使用客户端跟踪`startTimeInSeconds`元素，您可以使用 MediaTailor 来支持信标计时。

以下 JSON 响应显示了主要的信标类型：曝光量、起始值、四分位数和完成量。

> **Note**
>
> 互动广告局（IAB）视频曝光度衡量指南规定，展示量要求广告内容加载到客户端，并且至少需要开始向玩家呈现广告的时间。有关更多信息，请参阅 IAB 网站上的数字视频广告投放模板 (VAST)。

```
{
  "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://exampleadserver.com/tracking?event=midpoint"
          ],
          "duration": "PT0S",
          "durationInSeconds": 0.0,
          "eventId": "8104387",
          "eventType": "midpoint",
          "startTime": "PT25.367798651S",
          "startTimeInSeconds": 25.367
        },
        {
          "beaconUrls": [
            "http://exampleadserver.com/tracking?event=thirdQuartile"
          ],
          "duration": "PT0S",
          "durationInSeconds": 0.0,
          "eventId": "8104388",
          "eventType": "thirdQuartile",
          "startTime": "PT29.14279867S",
          "startTimeInSeconds": 29.142
        },
        {
          "beaconUrls": [
            "http://exampleadserver.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
    }
  ]
}
```

带有服务器端广告信标的混合模式

MediaTailor 支持会话跟踪的混合模式。在此模式下，该服务会发出与回放相关的广告跟踪事件，但会话可以使用完整的客户端跟踪负载

要使用播放前缀启用混合跟踪，请 MediaTailor 播放器根据您的协议，使用以下格式之一的请求初始化新的播放会话：

Example: HLS 格式

```
POST master.m3u8
    {
        "adsParams": {
           "deviceType": "ipad"
       },
       "reportingMode":"server"
    }
```

Example: DASH 格式

```
POST manifest.mpd
    {
        "adsParams": {
           "deviceType": "ipad"
       },
       "reportingMode":"server"
    }
```

MediaTailor 在混合模式下维护以下跟踪事件：

- 印象

- 晚于
- 第一个四分位数
- Midpoint
- 第三四分位数
- 完成
- `breakStart`(vmap)
- `breakEnd`(vmap)

Client-side 广告跟踪集成

本节介绍各种客户端广告跟踪服务器 MediaTailor 之间的集成。

主题

- 打开测量 SDK
- Datazoom 免费播放器 SDK
- Roku 广告框架 (RAF)
- TheoPlayer
- MediaTailor SDK

打开测量 SDK

互动广告局 (IAB) 开放式监测 SDK (OM SDK) 便于第三方评估向网络视频和原生应用环境投放的广告的可见度和验证度。

对于较早的 VAST 版本 3 文档，验证码应使用扩展节点加载，扩展类型为`AdVerifications`。扩展节点的根是与 VAST 4.1 元素具有相同架构的`AdVerifications`节点。

为了便于更轻松地采用OM SDK， MediaTailor 已与Datazoom合作提供免费的玩家软件开发工具包，这些软件开发工具包已针对开放式测量进行了配置和验证。有关更多信息，请参阅 Datazoom 免费播放器 SDK。

> **Note**
>
> MediaTailor 目前仅支持 VAST 版本 3。

Example: VAST 3 中的验证节点，版本低于 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 从<Extensions>节点提取AdVerifications数据并将其放入客户端跟踪adVerifications响应中的数组中。

Example: 客户端跟踪响应中的 AdVerifications 数组

```
{
  "avails": [
    {
      "adBreakTrackingEvents": [],
      "adMarkerDuration": null,
      "ads": [
        {
          "adId": "3062770",
          "adParameters": "",
          "adProgramDateTime": "2023-08-23T16:25:40.914Z",
          "adSystem": "2.0",
          "adTitle": "AD-polarbear-15",
          "adVerifications": [
            {
              "executableResource": [],
```

```
              "javaScriptResource": [
                {
                  "apiFramework": "omid",
                  "browserOptional": "true",
                  "uri": "https://verification.com/omid_verification.js"
                }
              ],
              "trackingEvents": [
                {
                  "event": "verificationNotExecuted",
                  "uri": "https://verification.com/trackingurl"
                }
              ],
              "vendor": "company.com-omid",
              "verificationParameters": "verification params key value pairs"
            }
          ],
          "companionAds": [],
          "creativeId": "00006",
          "creativeSequence": "1",
          "duration": "PT14.982S",
          "durationInSeconds": 14.982,
          "extensions": [
            {
              "content": "<AdVerifications>\n\t\t\t\t\t\t\t<Verification vendor=
\"company.com-omid\">\n\t\t\t\t\t\t\t<JavaScriptResource apiFramework=\"omid\"
 browserOptional=\"true\"><![CDATA[https://verification.com/omid_verification.js;]]></
JavaScriptResource>\n\t\t\t\t\t\t\t<TrackingEvents>\n\t\t\t\t\t\t\t<Tracking event=
\"verificationNotExecuted\"><![CDATA[;https://verification.com/trackingurl;]]></
Tracking>\n\t\t\t\t\t\t\t</TrackingEvents>\n\t\t\t\t\t\t\t<VerificationParameters><!
[CDATA[verification params key/value pairs;]]></VerificationParameters>\n\t\t\t\t\t\t</
Verification>\n\t\t\t\t\t</AdVerifications>",
              "type": "AdVerifications"
            }
          ],
          "mediaFiles": {
            "mediaFilesList": [],
            "mezzanine": ""
          },
          "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**
>
> 与 IAB 技术实验室合作，确保每年对应用程序进行认证以确保合规性。

有关 OM SDK 的更多信息，请参阅 IAB 技术实验室网站上的 O pen Measurement SDK 。

Datazoom 免费播放器 SDK

为了便于玩家软件开发工具包的更容易采用， MediaTailor 已与Datazoom合作提供免费的玩家软件开发工具包，这些软件开发工具包均使用和IAB Tech进行了配置Client-side AWS Elemental MediaTailor与谷歌广告管理器集成和测试。打开测量 SDK

Datazoom 播放器 SDK 支持以下功能：

- 直播和 VOD 播放列表
- DASH 和 HLS 规格
- 播放器供应商支持 Bitmovin、exoplayer、安卓媒体播放器、苹果 AVPlayer、Brightcove、Chromecast Receiver、hls.js、JWPlayer、 Dash.js Shaka 播放器、THEO 播放器、Roku 等 Video.js
- IAB Tech Lab 开放式测量认证，适用于特定设备
- Click-through 事件处理
- Ad-event 调度器，例如广告倒计时器、广告叠加和非线性事件、广告插播开始、广告插播结束
- Client-side 广告信标
- 谷歌编程访问库 (PAL) SDK，作为可选配置设置

Datazoom 还提供玩家 SDK 支持的付费分析和遥测服务。客户可以从 Datazoom 管理控制台选择加入和控制播放器 SDK 遥测功能。要访问Datazoom播放器软件开发工具包并了解有关增值遥测和分析服务的更多信息，请使用Datazoom网站上的联系信息。https://www.datazoom.io/partner-aws

Roku 广告框架 (RAF)

Roku广告框架（RAF）在整个Roku平台上保持了一致的广告体验。所有频道，包括视频广告，都必须满足Roku对英国皇家空军的认证要求。值得注意的是，该应用程序必须始终使用通过 RAF 触发客户端事件。 MediaTailor作为服务器端广告插入 (SSAI) 提供商，支持客户端事件触发。RAFX SSAI 适配器为 SSAI 清单服务器（即拼接器）和 RAF 提供接口。这些接口包括：

- 解析`masterURL`响应和提取 `playURL`AdURL，以及广告元数据。
- 将 MediaTailor SSAI 广告元数据转换为 RAF-usable 广告元数据，并配置 RAF 以进行播放。
- 观察直播事件和定时元数据。
- 按时匹配直播事件、广告元数据和触发事件像素。
- Pinging/polling 然后AdURL，按照 MediaTailor SSAI 清单服务器的要求，解析并重新配置 RAF。

有关适用于 RAF 的 SSAI 适配器的更多信息，请参阅 Roku 网站上的 “使用 Roku 适配器实现 Server-Side 广告插入”。

TheoPlayer

TheoPlayer 与的集成执行以下 MediaTailor 操作：

- 提供支持 VOD 和直播工作流程中 HLS 和 DASH 的 MediaTailor 客户端事件跟踪的功能。
- 仅支持为线性广告发送跟踪信标。
- 在广告期间禁用搜索。但是，当用户寻求突破广告时段时，播放广告的逻辑不成立。

要了解有关 SSAI 的更多信息 TheoPlayer，并查看其中的网络、安卓、iOS 和 tvOS 软件开发工具包 MediaTailor，请参阅网站MediaTailor上的内容。 TheoPlayer

MediaTailor SDK

AWS Elemental 维护 JavaScript-based 软件开发套件 (SDK)。 AWS Elemental 按原样提供 SDK，不含任何暗示担保。使用 SDK 作为参考演示，简化入门到使用 MediaTailor。该软件开发工具包展示了如何与 MediaTailor 客户端跟踪 API 进行交互。该软件开发工具包为玩家实现了客户端广告跟踪和报告。 HTML5-based SDK 初始化 MediaTailor 客户端报告会话，然后定期请求广告跟踪信息。在播放期间，当检测到新的广告事件时，SDK 会发出广告跟踪事件。

S MediaTailor DK 支持以下功能：

- 直播和 VOD 播放列表
- DASH 和 HLS 规格
- Click-through 事件处理
- Ad-event 调度员
- 自定义事件挂钩
- Client-side 广告信标。有关发送广告信标的更多信息，请参阅Client-side 信标。

> **Note**
>
> 提交 AWS 支持请求以获得的示例 JavaScript SDK MediaTailor。您将收到该软件包及其文件的下载链接。

使用以下方式浏览广告信标 GetTracking

使用终`GetTracking`端节点缩小向玩家返回的广告数量。例如，如果清单窗口很宽，跨越很长时间，则返回的广告信标的数量可能会影响玩家的表现。

`GetTracking`返回一个`NextToken`值，您可以使用该值通过浏览返回的信标列表来缩小返回信标的数量。您可以循环浏览这些`NextToken`值，找到广告信标`StartTimeInSeconds`字段的所需值。

- 首次调用时GetTracking，将返回所有可能落入清单窗口的广告，包括每个广告的NextToken和值。
- 如果GetTracking请求不包含NextToken，则会返回清单窗口中的所有广告。
- 如果GetTracking请求包含NextToken但没有新的信标可返 MediaTailor 回，则返回您在原始请求NextToken中发送的相同值。
- 当没有与广告相对应的信标时，将该广告从其响应中GetTracking移除。
- 来自的代币GetTracking将在 24 小时后过期。如果某个NextToken值大于 24 小时，则下次调用将GetTracking返回空值。NextToken

GetTracking 来自玩家的通用调用顺序

来自客户端玩家的GetTracking请求是一个 POST，其请求正文包含与令牌相关的广告NextToken和信标。

```
https://YouMediaTailorUrl/v1/tracking
{

     "NextToken": "value"
     .
     .
     .
}
```

GetTracking与一起使用的一般顺序NextToken如下：

1. 拨打第一个电话GetTracking。

   将返回所有广告和信标以及后续通话NextToken的第一个广告和信标。
2. 如果的值NextToken为空，则 MediaTailor 返回所有广告信标。
3. 如果已过期，NextToken则 MediaTailor 返回 HTTP 返回码 400 错误消息。

   再次致电GetTracking以检索有效的 NextToken s。
4. 扫描整个响应，找StartTimeInSeconds出在所需范围内的广告信标。
5. GetTracking使用与所需NextToken关联的值向进行新呼叫StartTimeInSeconds。
6. 如有必要，可以再次循环浏览返回的广告，直到找到想要玩的确切广告。

扩展示例

此示例说明如何使用 GetTracking's NextToken 来限制返回给玩家的广告信标的数量。

MediaTailor 收到GetTracking请求。该响应包含一个 ID 为 9935407 的广告和两个StartTimeInSeconds值分别为 52.286 和 48.332 秒的信标。

MediaTailor 发送 JSON 响应，NextToken如下所示：

```
{
"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
    }
  ]
}
```

在下一个GetTracking请求中，使用NextToken值:进行 MediaTailor 响应 JF57ITe48t1441mv7TmLKuZLroxDzfIslp6BiSNL1IJmzPVMDN0lqrBYycgMbKEb。

MediaTailor 使用与上次通话中设置的StartTimeInSeconds广告和信标相匹配NextToken的广告和信标进行响应。

假设现在的响应中除了之前的 ID 为 9935407 的广告外，还包括另一个 ID 为 9235407 的广告。ID 为 9235407 的信标有 StartTimeInSeconds s 132.41 和 70.339。

MediaTailor 对会话中的所有信标进行迭代，选择时间StartTimeInSeconds大于 52.286 秒的信标，即 ID 为 9235407 的广告中的信标 3 和信标 4：

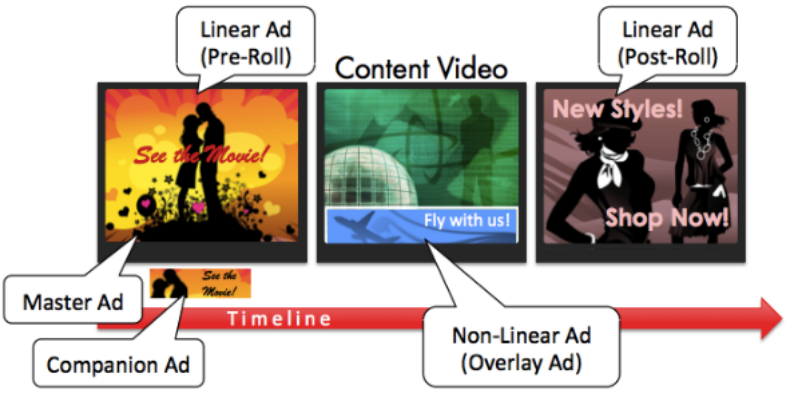
```
{
  "NextToken": ZkfknvbfsdgfbsDFRdffg12EdffecFRvhjyjfhdfhnjtsg5SDGN
  "avails": [
    {
      "ads": [
        {
          "adId": "9235407",
          "adVerifications": [],
          "companionAds": [],
          "creativeId": "",
```

```
          "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
    }
  ]
}
```

叠加广告

对于想要在不中断插播广告观看体验的情况下提高盈利能力的直播工作流程，您可以利用当前的 AWS Elemental MediaTailor 集成来指导在客户端呈现的广告格式。这种类型的广告被称为叠加广告。叠加广告是非线性视频广告，以 “广告”、“非线性视频L-band 广告”、“画中画广告”、“动态叠加”、“内容内广告” 或 “框架广告” 的形式出现。

MediaTailor 将分段类型的 SCTE-35 标记检测`id=0x38`为带内信号，以获得叠加广告插入机会。该 SCTE-35 标记会导致 MediaTailor 向广告决策服务器 (ADS) 发送请求，然后广告决策服务器 (ADS) 在 VAST 响应中使用非线性广告有效载荷进行响应。 MediaTailor 解析 VAST 响应以支持叠加广告插入。MediaTailor 不会对线性广告进行任何拼接，而是向玩家发出信号，表明有非线性叠加广告可供播放。该信号允许玩家从客户端跟踪端点获取并关联要播放的非线性广告。然后，玩家处理与这些广告相关的展示、举报和其他任务。例如，玩家的开发者可以使用供应商提供的支持叠加广告格式的设备 SDK。有关客户端跟踪集成的更多信息，请参阅。Client-side 广告跟踪集成



主题

- 使用叠加广告的先决条件 MediaTailor
- 开始使用叠加式广告 MediaTailor
- 中叠加广告的日志记录和指标 MediaTailor
- 中叠加广告的计费 MediaTailor

使用叠加广告的先决条件 MediaTailor

使用叠加式广告时，需要满足以下先决条件 MediaTailor：

- 工作流程必须是实时的，而不是视频点播 (VOD)。

- 必须将广告决策服务器 (ADS) 响应配置为在 VAST 响应中仅返回非线性广告。 MediaTailor 为了拼接广告而忽略任何线性广告。
- 清单必须使用分段类型的 SCTE-35 时间信号消息`id=0x38`来调用叠加广告功能。
- 流媒体提供商必须控制客户端设备应用程序，并与 MediaTailor 客户端跟踪 API 集成。

开始使用叠加式广告 MediaTailor

本节介绍如何开始使用的叠加广告功能。 MediaTailor您将设置 SCTE-35 信令、配置广告决策服务器 (ADS) 响应并设置会话级控制。

主题

- 启用叠加广告
- 使用客户端元数据跟踪叠加广告

启用叠加广告

MediaTailor 默认情况下，对叠加广告的支持处于启用状态。清单中特定的 SCTE-35 广告标记类型会触发叠加广告的插入。由于某些玩家可能不支持叠加广告的客户端渲染，因此您可以在会话级别禁用该功能。

要使用 HLS 或 DASH 播放前缀禁用叠加广告支持，请执行以下操作：

- 在播放器中，根据您的协议，使用以下格式之一的请求初始化新的 MediaTailor 播放会话：
  - 示例：HLS 格式

```
GET mediatailorURL/v1/master/hashed-account-id/origin-id/asset-id?
aws.overlayAvails=off
```

  - 示例：DASH 格式

```
GET mediatailorURL/v1/master/hashed-account-id/origin-id/asset-id?
aws.overlayAvails=off
```

要使用会话初始化前缀禁用叠加广告支持，请执行以下操作：

- 在播放器上，为会话初始化请求构造一个 JSON 消息正文，以 MediaTailor：

- 要禁用广告叠加层支持，请添加一个值为的overlays对象作为顶级键。off默认overlays值为on。
- （可选）提供 MediaTailor 随后传递给adsParams对象内的 ADS 的任何参数。这些参数对应于 MediaTailor 配置的 ADS 模板 URL 中的 [player_params.param] 设置。

Example HLS：

```
POST master.m3u8
    {
       "adsParams": {
           "deviceType": "ipad"
       },
       "overlayAvails": "off"
    }
```

Example DASH：

```
POST manifest.mpd
    {
        "adsParams": {
           "deviceType": "androidmobile"
       },
       "overlayAvails": "off"
    }
```

清单信号

MediaTailor 当它在清单中看到特定的 SCTE-35 标记时，就会触发叠加广告支持。所需的信号是拼接命令类型 6 或时间信号，即提供商叠加广告开始信号。此信号的分段类型 ID 为 0x38

以下示例显示了 JSON 对象中的 0x38 SCTE-35 标记。

```
{
  "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,
      "indentifier": "CUEI",
      "segmentationEventId": 158389361,
      "segmentationEventCancelIndicator": false,
      "programSegmentationFlag": true,
      "segmentationDurationFlag": true,
      "deliveryNotRestrictedFlag": false,
      "webDeliveryAllowedFlag": true,
      "noRegionalBlackoutFlag": true,
      "archiveAllowedFlag": true,
      "deviceResctrictions": 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
}
```

以下示例显示了以二进制（基数 32 /十六进制）值表示的 SCTE-35 信号：

```
0xfc30350000000000000fff00506fe001b78c8001f021d435545490970d4717fdf00000dbba009076f7665726c617
```

以下示例显示了 HLS 和 DASH 清单中的 SCTE-35 标记。

Example: HLS 清单

```
#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=0xfc30350000000000000fff00506fe001b78c8001f021d435545490970d4717fdf00000dbba009076f766572
#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 清单

```
<?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>
```

广告决策服务器 (ADS) 响应

ADS 响应必须包含一个有效的追踪事件。至少，跟踪事件可以是Impression跟踪事件。跟踪事件应至少包含一个NonLinear广告。此广告是叠加广告，采用静态、HTML 或 iFrame 资源的形式。

```
<vmap AdBreak breaktype="linear" breakId="csoverlay"
```

如果 VAST 响应是带有 breakType of 的 VMAPnonlinear，则可用元数据位于nonLinearAvails根对象内。如果 VAST 响应是带有 VMAP 的 VMA breakType Plinear，或者是没有 VMAP 的普通 VAST 响应，则可用元数据位于avails根对象内。

以下 VAST 响应是一个封装的 VMAP 响应，其breakTypelinear值为。

除了封装的 VMAP 响应外， MediaTailor 还支持breakType值为的封装 VMAP 响应和普通的 VAST 响应。nonlinear

```
<?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: DASH 清单来源到 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：包含广告 ID 装饰的 MediaTailor 个性化 DASH 清单

```
<?xml version="1.0" encoding="utf-8"?>
<MPD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns="urn:mpeg:dash:schema:mpd:2011" xmlns:scte35="urn:scte:scte35:2013:xml"
 xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 http://standards.iso.org/
ittf/PubliclyAvailableStandards/MPEG-DASH_schema_files/DASH-MPD.xsd" id="201"
 type="dynamic" publishTime="2022-11-07T19:59:05+00:00" minimumUpdatePeriod="PT2S"
 availabilityStartTime="2022-11-07T06:57:11.250000+00:00" minBufferTime="PT10S"
 suggestedPresentationDelay="PT20.000S" timeShiftBufferDepth="PT58.999S"
 profiles="urn:mpeg:dash:profile:isoff-live:2011">
  <Period start="PT46827.601S" id="0" duration="PT88.321S">
  ...
  </Period>
  <Period start="PT46915.922S" id="45" duration="PT6.006S">
  <EventStream schemeIdUri="urn:sva:advertising-wg:ad-id-signaling" timescale="90000">
    <Event presentationTime="13500000" duration="1351350">
    <![CDATA[{"version": 1,"identifiers": [{"scheme":
 "urn:smpte:ul:060E2B34.01040101.01200900.00000000","value": "adId","ad_position":
 "adId", "ad_type":"overlay","creative_id": "creativeId","tracking_uri":
 "trackingUri"}]}]]></Event>
  </EventStream>
  ...
  </Period>
  <Period start="PT46921.928S" id="49">
  ...
  </Period>
</MPD>
```

使用客户端元数据跟踪叠加广告

MediaTailor 将叠加广告置于可nonLinearAdsList用的范围内。 MediaTailor 客户端跟踪 API 有两个根对象，名为avails和。nonLinearAvails如果 VAST 响应是带有 breakType of 的

VMAPnonlinear，则可用元数据位于nonLinearAvails根对象内。如果 VAST 响应是带有 VMAP 的 VMA breakType Plinear，或者是没有 VMAP 的普通 VAST 响应，则可用元数据位于avails根对象内。

有关客户端跟踪的更多信息，请参阅Client-side 广告追踪。

以下示例显示了breakType值为的普通 VAST 响应或 VMAP 响应。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": []
}
```

以下示例显示了breakType值为的普通 VMAP 响应。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
    }
  ]
}
```

中叠加广告的日志记录和指标 MediaTailor

本节介绍中叠加广告的日志记录和指标 MediaTailor。有关设置日志记录的更多信息，请参阅监控和标记 AWS Elemental MediaTailor 资源。

主题

- CloudWatch 日志
- CloudWatch 指标

CloudWatch 日志

CloudWatch 收集有关叠加广告的以下日志信息：

- VAST_RESPONSE-显示有关非线性广告列表的信息。
- FILLED_PROVIDER_OVERLAY-显示有关非线性广告的信息。

> **Note**
>
> RAW_ADS_RESPONSE是一个可选事件，显示来自 ADS 的原始响应。使用此事件在试运行和测试环境中特别有用。要在配置或账户上启用此事件，请向 Support 提交 AWS 工单。

CloudWatch 指标

MediaTailor 将叠加广告指标与其他广告指标分开收集。 MediaTailor成功从 ADS 获取广告后收集这些指标。您不必通过轮询 GetTracking API 来收集指标。

下表描述了叠加广告的 CloudWatch 指标：

指标	说明
AdDecisionServer.OverlayAds	在您指定的 CloudWatch 时间段内，ADS 响应中包含的叠加广告数量。
AdDecisionServer.OverlayErrors	在您指定的时间段内从 ADS MediaTailor 收到的非 HTTP 200 状态码响应、空响应和超 CloudWatch 时响应的数量。

指标	说明
AdDecisionServer.OverlayFilled	成功填充至少一个叠加广告的可用数量： • 1-至少有一个有效的广告。 • 0-要么 MediaTailor 没有收到任何叠加广告，要么出现其他故障。 SampleCount 跟踪已填充的可用数量。 Sum跟踪成功填充的叠加层可用数量。
AdDecisionServer.OverlayMinSuggestedDuration	在您指定的minSuggestedDuration 时间段内从 ADS MediaTailor 收到的所有广告 CloudWatch 的时长总和（以毫秒为单位）。如果minSuggestedDuration 未指定，则显示的持续时间为计划持续时间。
AdDecisionServer.OverlayLatency	向 ADS 发 MediaTailor 出的请求的响应时间，以毫秒为单位。
AdDecisionServer.OverlayTimeouts	在您指定的时间段内向 ADS 发出的超 CloudWatch 时请求数。
AdsBilled	有关广告收费的更多信息，请参阅中叠加广告的计费 MediaTailor。
Avail.*	因为 MediaTailor 没有为叠加广告做任何规划，所以 CloudWatch 不显示任何Avail.X指标。
SkippedReason.*	因为 MediaTailor 没有为叠加广告做任何规划，所以 CloudWatch 不显示任何SkippedReason.X 指标。

中叠加广告的计费 MediaTailor

MediaTailor 根据ADS响应中的非线性广告数量向客户开具账单。此数字包括超过中断时间的非线性广告。 MediaTailor 填写空缺后，它会为其填充的广告收费。

对于预取工作流程，在检索预取时 MediaTailor 不收取广告账单，而是在该会话的消费窗口中看到兼容的广告可用时。

有关其他账单信息，请参阅https://aws.amazon.com/mediatailor/pricing/。

广告 ID 装饰

AWS Elemental MediaTailor 从内容过渡到广告插播时段时，会执行服务器端广告拼接。 MediaTailor 可以用与已拼接的广告关联的元数据来限制清单内容。这样做可以带来以下好处：

> **Important**
>
> Ad-ID 装饰与服务器引导式广告插入 (SGAI) 方法不兼容，因为只有在广告播放期间获取素材资源列表时才会知道填充`X-AD-CREATIVE-SIGNALING` 标题的字段，而不是最初写入可缓存清单时才会知道。

- 视频开始时间 (VST) 得到改善
- MediaTailor 可以支持服务器端广告插入和服务器引导式广告插入的混合模式
- 服务器端会话可以使用广告位置标记来建立播放时间表
- 对于已经使用 MediaTailor API 构建播放时间表的客户端会话，会话 VST 有所改进，因为会话不依赖调用跟踪 API 来构建时间表
- 可以利用 MediaTailor 服务器端广告插入以及场景中显示的客户端渲染广告。这样，玩家的软件开发套件 (SDK) 就无需单独集成即可直接为客户端广告调用广告投放实体。 MediaTailor 可以通过清单和客户端跟踪 API 来销售广告。

将每个创意广告素材资源与唯一标识符关联是有标准的。该关联允许广告商、代理商、供应商和发布商在其独立的工作流程中关联创意广告资产。随着直播指标和监控的不断改进，以及越来越多的发行商使用基于服务器的插入架构，因此需要在 interleaved/stitched 演示文稿中（例如个性化清单中）准确传达分配给各个创意资产的标识符。

主题

- 为会话启用广告 ID 信令
- 清单和广告元数据插入
- 广告决策服务器 (ADS) 互动

- 客户端跟踪 API

为会话启用广告 ID 信令

在会话初始化期间，必须启用广告 ID 信令功能。启用该功能的过程不同于使用 HLS/DASH 播放前缀（隐式会话初始化）创建会话，而使用会话初始化前缀（显式会话初始化）创建会话。

使用 HLS/DASH 播放前缀为会话启用广告 ID

- 在播放器中，根据您的协议，使用以下格式之一的请求初始化新的 MediaTailor 播放会话：
  - 示例：HLS 格式

```
GET <mediatailorURL>/v1/master/<hashed-account-id>/<origin-id>/<asset-id>?
aws.adSignalingEnabled=true
```

  - 示例：DASH 格式

```
GET <mediatailorURL>/v1/dash/<hashed-account-id>/<origin-id>/<asset-id>?
aws.adSignalingEnabled=true
```

使用会话初始化前缀为会话启用广告 ID

- 在播放器上，为会话初始化请求构造一个 JSON 消息正文，以 MediaTailor：
  - 在`adsParams`对象内部，提供 MediaTailor 应传递给 ADS 的任何参数。这些参数对应于 MediaTailor 配置的 ADS 模板网址中的`[player_params.param]`设置。
  - 要启用广告 ID 信号，请添加一个`adSignaling`对象作为顶级对象，然后在其中添加一个名为的`true`参数`enabled`，值为。默认`adSignaling`值为`disabled`。
  - 示例：HLS 格式

```
POST master.m3u8
    {
       "adsParams": {
           "deviceType": "ipad"
       },
       "adSignaling": {
           "enabled": "true"
```

```
        },
        "reportingMode": "client"
    }
```

- 示例：DASH 格式

```
POST manifest.mpd
    {
        "adsParams": {
           "deviceType": "ipad"
        },
        "adSignaling": {
             "enabled": "true"
         },
         "reportingMode": "client"
    }
```

清单和广告元数据插入

在广告拼接过程中，向清单中 MediaTailor 添加与正在拼接的每个广告素材关联的唯一 ID。MediaTailor 从 VAST 响应中该广告素材的`id`属性值中获取该广告素材的唯一 ID。如果广告素材缺少 ID 属性值，则 MediaTailor 会发布一个空值 (`id=""`)。

MediaTailor 使用清单中的元数据信号来解耦广告素材元数据的客户跟踪 API 与整个时间轴 timing/positioning 内的依赖关系。这种解耦可以减少播放延迟（尤其是在VOD场景中），在这种场景中，播放器的用户界面 (UI) 会在初始化播放之前在时间轴上呈现广告插播位置。

添加的元数据采用以下形式：

- 对于 HLS 清单，添加的元数据采用有效期内每个广告的`DATERANGE`标签形式。
- 对于 DASH 清单，添加的元数据在每个广告周期内采用`Event`元素的形式。

以下 JSON 消息正文显示了 VAST 响应示例：

```
{
  "version": 1,
  "identifiers": [
    {
      "scheme": "urn:smpte:ul:060E2B34.01040101.01200900.00000000",
      "value": "creativeId",
```

```
      "ad_position": "adId",
      "ad_type": "adType",
      "tracking_uri": "trackingUri",
      "custom_vast_data":"customVastData"
    }
  ]
}
```

在上述示例中：

- *creativeId*是广告Creative元素的Id属性值
- *adId*要么是与广告开头关联的 HLS 序列号，要么是广告的 DASH 时段 ID
- *adType*是avail或overlay，基于 VAST 响应
- *trackingUri*是会 MediaTailor 话的相对跟踪端点，格式为 ../../../../tracking/*hashed-account-id*/*origin-id*/*session-id*
- *customVastData*是从 V creative_signaling AST 扩展中 MediaTailor 提取的值。MediaTailor 使用 CDATA 节点的内容（如果存在）。有关更多详细信息和 VAST 响应示例，请参阅广告决策服务器 (ADS) 互动部分。

使用广告元数据个性化 HLS 清单

对于直播 HLS 直播， MediaTailor 仅当直播包含PROGRAM-DATA-TIME标签时才添加元数据，每个清单持续时间至少添加一次。对于视频点播 (VOD) 流，在个性化清单中至少 MediaTailor 添加PROGRAM-DATE-TIME一个片段，其中每个 VOD 资产的开始时间均为零纪元 ()。1970-01-01T00:00:00Z如果源清单PROGRAM-DATE-TIME包含现有内容，则 MediaTailor 保留该内容。

MediaTailor 使用广告决策服务器 (ADS) 返回的广告素材对清单进行个性化设置。对于每则广告， MediaTailor 还包括一个跨越广告持续时间的DATERANGE标签。DATERANGE标签格式与 SV A 技术出版物 2023 年版中 DASH 和 HLS 中的广告创意信号一节中描述的格式类似。

MediaTailor 生DATERANGE成的具有唯一的 ID 值。为确保唯一性（考虑到 M apping SCTE-35 中指定的指导方针 EXT-X-DATERANGE），请将第一个广告段的编MEDIA-SEQUENCE号与可用范围内的广告序列号 MediaTailor 耦合。

对于启用了 slate 的配置上填充不足的广告时段，请将 slate 区段 MediaTailor 附加到可用内容的末尾，用DISCONTINUITY标签分隔，但不包含任何元数据。DATERANGE

对于拼接到个性化清单中的每个广告，都会 MediaTailor 添加广告素材元数据，该元数据在自定义标签中以 base64 编码的数据表示。DATERANGE

Example线性 HLS 原点 (#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:/DAlAAAAAyiYAP/wFAUAAAACf+//jPl97P4AUmNiAAEBAQAAse4/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=/DAlAAAAAyiYAP/wFAUAAAACf
+//jPl97P4AUmNiAAEBAQAAse4/gA==
#EXTINF:6.006,
index_1_401.ts?m=1676054627
#EXT-X-CUE-OUT-CONT:ElapsedTime=12.145,Duration=59.993,SCTE35=/DAlAAAAAyiYAP/wFAUAAAACf
+//jPl97P4AUmNiAAEBAQAAse4/gA==
#EXTINF:6.006,
index_1_402.ts?m=1676054627
#EXT-X-CUE-OUT-CONT:ElapsedTime=18.151,Duration=59.993,SCTE35=/DAlAAAAAyiYAP/wFAUAAAACf
+//jPl97P4AUmNiAAEBAQAAse4/gA==
#EXTINF:6.006,
index_1_403.ts?m=1676054627
#EXT-X-CUE-OUT-CONT:ElapsedTime=24.157,Duration=59.993,SCTE35=/DAlAAAAAyiYAP/wFAUAAAACf
+//jPl97P4AUmNiAAEBAQAAse4/gA==
#EXTINF:6.006,
index_1_404.ts?m=1676054627
#EXT-X-CUE-OUT-CONT:ElapsedTime=30.163,Duration=59.993,SCTE35=/DAlAAAAAyiYAP/wFAUAAAACf
+//jPl97P4AUmNiAAEBAQAAse4/gA==
#EXTINF:6.006,
index_1_405.ts?m=1676054627
#EXT-X-CUE-OUT-CONT:ElapsedTime=36.169,Duration=59.993,SCTE35=/DAlAAAAAyiYAP/wFAUAAAACf
+//jPl97P4AUmNiAAEBAQAAse4/gA==
#EXTINF:6.006,
index_1_406.ts?m=1676054627
#EXT-X-CUE-OUT-CONT:ElapsedTime=42.175,Duration=59.993,SCTE35=/DAlAAAAAyiYAP/wFAUAAAACf
+//jPl97P4AUmNiAAEBAQAAse4/gA==
#EXTINF:6.006,
index_1_407.ts?m=1676054627
```

```
#EXT-X-CUE-OUT-CONT:ElapsedTime=48.181,Duration=59.993,SCTE35=/DAlAAAAAyiYAP/wFAUAAAACf
+//jPl97P4AUmNiAAEBAQAAse4/gA==
#EXTINF:6.006,
index_1_408.ts?m=1676054627
#EXT-X-CUE-OUT-CONT:ElapsedTime=54.187,Duration=59.993,SCTE35=/DAlAAAAAyiYAP/wFAUAAAACf
+//jPl97P4AUmNiAAEBAQAAse4/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线性 HLS 原点 (#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=0xFC3025000000003289800FFF0140500000000027FEFFF8CF97DECFE00526362000101010000B1EE3F80
#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线性 HLS 个性化清单（带有创意广告信号）：

MediaTailor 生DATERANGE成的具有唯一的 ID 值。为确保唯一性（考虑到 M apping SCTE-35 中指定的指导方针 EXT-X-DATERANGE），请将第一个广告段的编MEDIA-SEQUENCE号与可用范围内的广告序列号 MediaTailor 耦合。

在以下示例中， MediaTailor 将 MEDIA-SEQUENCE 421 与广告位置编号连接起来。

```
#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 起源（带有 SCTE 信号）：

```
#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 起源：

```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-TARGETDURATION:7
#EXT-X-MEDIA-SEQUENCE:1
#EXT-X-PLAYLIST-TYPE:VOD
#EXTINF:6,
index_720p1500k_00001.ts
#EXTINF:6,
index_720p1500k_00002.ts
#EXTINF:6,
index_720p1500k_00003.ts
#EXTINF:6,
```

```
index_720p1500k_00004.ts
#EXTINF:4,
index_720p1500k_00005.ts
#EXTINF:2,
index_720p1500k_00006.ts
#EXTINF:6,
index_720p1500k_00007.ts
#EXTINF:6,
index_720p1500k_00008.ts
#EXTINF:6,
index_720p1500k_00009.ts
#EXTINF:6,
index_720p1500k_00010.ts
#EXTINF:6,
index_720p1500k_00011.ts
#EXTINF:6,
index_720p1500k_00012.ts
```

Example VOD HLS 个性化清单：

MediaTailor 添加PROGRAM-DATE-TIME到 VOD 清单中，以便将其用作指示广告位置的 HLS DATERANGE 元素的锚点。

MediaTailor 生DATERANGE成的具有唯一的 ID 值。为确保唯一性（考虑到 M apping SCTE-35 中指定的指导方针 EXT-X-DATERANGE），请将第一个广告段的编MEDIA-SEQUENCE号与可用范围内的广告序列号 MediaTailor 耦合。

在以下示例中， MediaTailor 将 MEDIA-SEQUENCE 421 与广告位置编号连接起来。

```
#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/SKO-22/asset-1/hls/
index_720p1500k_00001.ts
#EXTINF:6.0,
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SKO-22/asset-1/hls/
index_720p1500k_00002.ts
#EXTINF:6.0,
```

```
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SKO-22/asset-1/hls/
index_720p1500k_00003.ts
#EXTINF:6.0,
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SKO-22/asset-1/hls/
index_720p1500k_00004.ts
#EXTINF:4.0,
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SKO-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/SKO-22/asset-1/hls/
index_720p1500k_00006.ts
#EXTINF:6.0,
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SKO-22/asset-1/hls/
index_720p1500k_00007.ts
#EXTINF:6.0,
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SKO-22/asset-1/hls/
index_720p1500k_00008.ts
#EXTINF:6.0,
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SKO-22/asset-1/hls/
index_720p1500k_00009.ts
#EXTINF:6.0,
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SKO-22/asset-1/hls/
index_720p1500k_00010.ts
#EXTINF:6.0,
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SKO-22/asset-1/hls/
index_720p1500k_00011.ts
#EXTINF:6.0,
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SKO-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
```

使用广告元数据对 DASH 清单进行个性化设置

MediaTailor 使用广告决策服务器 (ADS) 返回的广告素材对清单进行个性化设置。对于每个广告，MediaTailor 还包括一个跨越广告持续时间的`EventStream`元素。`Event`元素格式与 SV A 技术出版物 2023 年版中 DASH 和 HLS 中的广告创意信号一节中描述的格式类似。

对于已启用 slate 的配置上填充不足的广告时段，请将 slate 时段 MediaTailor 附加到可用期的末尾，但不包含任何元数据 `EventStream`

对于拼接到个性化清单中的每个广告，都会 MediaTailor 添加广告素材元数据，表示为`CDATA`元素中的一个`Event`元素。

Example线性 DASH 原点（内联 SCTE 属性）：

```
<MPD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns="urn:mpeg:dash:schema:mpd:2011" xmlns:scte35="urn:scte:scte35:2013:xml"
 xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 http://standards.iso.org/
ittf/PubliclyAvailableStandards/MPEG-DASH_schema_files/DASH-MPD.xsd" id="201"
 type="dynamic" publishTime="2023-02-10T21:08:40+00:00" minimumUpdatePeriod="PT6S"
 availabilityStartTime="2023-02-09T22:47:05.865000+00:00" minBufferTime="PT10S"
 suggestedPresentationDelay="PT20.000S" timeShiftBufferDepth="PT88.999S"
 profiles="urn:mpeg:dash:profile:isoff-live:2011">
  <Period start="PT80141.456S" id="104" duration="PT304.103S">
    <AdaptationSet id="1485523442" mimeType="video/mp4" segmentAlignment="true"
 startWithSAP="1" subsegmentAlignment="true" subsegmentStartsWithSAP="1"
 bitstreamSwitching="true">
      <SegmentTemplate timescale="60000" media="index_video_$RepresentationID$_0_
$Number$.mp4?m=1676062374" initialization="index_video_$RepresentationID$_0_init.mp4?
m=1676062374" startNumber="151" presentationTimeOffset="4808487386">
        <SegmentTimeline>
          <S t="4824975858" d="360360" r="3"/>
          <S t="4826417298" d="316316"/>
        </SegmentTimeline>
      </SegmentTemplate>
      <Representation id="1" width="960" height="540" frameRate="30000/1001"
 bandwidth="1800000" codecs="avc1.4D401F"/>
      <Representation id="3" width="640" height="360" frameRate="30000/1001"
 bandwidth="1200000" codecs="avc1.4D401E"/>
      <Representation id="5" width="480" height="270" frameRate="30000/1001"
 bandwidth="800000" codecs="avc1.4D4015"/>
    </AdaptationSet>
    <AdaptationSet id="1377232898" mimeType="audio/mp4" segmentAlignment="0"
 lang="eng">
```

```
      <Label>eng</Label>
      <SegmentTemplate timescale="48000" media="index_audio_$RepresentationID$_0_
$Number$.mp4?m=1676062374" initialization="index_audio_$RepresentationID$_0_init.mp4?
m=1676062374" startNumber="151" presentationTimeOffset="3846790126">
        <SegmentTimeline>
          <S t="3859981294" d="287744"/>
          <S t="3860269038" d="288768"/>
          <S t="3860557806" d="287744"/>
          <S t="3860845550" d="288768"/>
          <S t="3861134318" d="252928"/>
        </SegmentTimeline>
        </SegmentTemplate>
        <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"/>
          <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 Linear DASH 个性化清单（带有创意广告信号）：

```
<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/
f9f38deca3f14fc4b5ab3cdbd76cfb9e/</BaseURL>
    <Location>https://777788889999.mediatailor.us-west-2.amazonaws.com/
v1/dash/94063eadf7d8c56e9e2edd84fdf897826a70d0df/emt/out/v1/
f9f38deca3f14fc4b5ab3cdbd76cfb9e/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:smpte:ul:060E2B34.01040101.01200900.00000000","value": "155_1","ad_position":
 "155_1", "ad_type":"avail","creative_id": "123","tracking_uri": "../../../../v1/
tracking/hashed-account-id/origin-id/session-id","custom_vast_data":"123abc"}]}]]>
            </Event>
        </EventStream>
        <AdaptationSet bitstreamSwitching="false" frameRate="30000/1001"
 mimeType="video/mp4" segmentAlignment="true" startWithSAP="1"
 subsegmentAlignment="true" subsegmentStartsWithSAP="1">
            <SegmentTemplate startNumber="1" timescale="90000"/>
            <Representation bandwidth="1800000" codecs="avc1.64001f" height="540"
 id="1" width="960">
                <SegmentTemplate initialization="asset_540_1_2init.mp4"
 media="asset_540_1_2_$Number%09d$.mp4" startNumber="1" timescale="90000">
                    <SegmentTimeline>
                        <S d="180180" r="6" t="0"/>
                        <S d="90090" t="1261260"/>
                    </SegmentTimeline>
                </SegmentTemplate>
            </Representation>
            <Representation bandwidth="1200000" codecs="avc1.64001e" height="360"
 id="3" width="640">
                <SegmentTemplate initialization="asset_360_1_1init.mp4"
 media="asset_360_1_1_$Number%09d$.mp4" startNumber="1" timescale="90000">
                    <SegmentTimeline>
                        <S d="180180" r="6" t="0"/>
                        <S d="90090" t="1261260"/>
                    </SegmentTimeline>
                </SegmentTemplate>
            </Representation>
            <Representation bandwidth="800000" codecs="avc1.640015" height="270" id="5"
 width="480">
```

```
                <SegmentTemplate initialization="asset_270_0_0init.mp4"
 media="asset_270_0_0_$Number%09d$.mp4" startNumber="1" timescale="90000">
                    <SegmentTimeline>
                        <S d="180180" r="6" t="0"/>
                        <S d="90090" t="1261260"/>
                    </SegmentTimeline>
                </SegmentTemplate>
            </Representation>
        </AdaptationSet>
        <AdaptationSet lang="eng" mimeType="audio/mp4" segmentAlignment="0">
            <SegmentTemplate initialization="asset_audio_128_3init.mp4"
 media="asset_audio_128_3_$Number%09d$.mp4" startNumber="1" timescale="48000"/>
            <Label>eng</Label>
            <Representation audioSamplingRate="48000" bandwidth="128000"
 codecs="mp4a.40.2" id="6">
                <SegmentTemplate initialization="asset_audio_128_3init.mp4"
 media="asset_audio_128_3_$Number%09d$.mp4" startNumber="1" timescale="48000">
                    <SegmentTimeline>
                        <S d="98304" t="0"/>
                        <S d="96256" r="1" t="98304"/>
                        <S d="95232" t="290816"/>
                        <S d="96256" r="2" t="386048"/>
                        <S d="48128" t="674816"/>
                    </SegmentTimeline>
                </SegmentTemplate>
                <AudioChannelConfiguration
 schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
            </Representation>
        </AdaptationSet>
    </Period>
    <Period id="155_2" start="PT22H21M0.575S">
        <BaseURL>https://777788889999.mediatailor.us-west-2.amazonaws.com/
v1/dashsegment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/
emt/672ed481-4ffd-4270-936f-7c8403947f2e/155/155_2/</BaseURL>
        <EventStream schemeIdUri="urn:sva:advertising-wg:ad-id-signaling"
 timescale="90000">
            <Event presentationTime="0" duration="1351350">
                <![CDATA[{"version": 1,"identifiers": [{"scheme":
 "urn:smpte: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:smpte: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>
```

广告决策服务器 (ADS) 互动

MediaTailor 使用 VAST 响应中的广告素材id属性值作为广告 ID 信号中的值。如果id属性值为空或未出现在 VAST 响应中，则在广告 ID 信号中 MediaTailor 放置一个空值。

Example反应非常广泛：

以下示例 VAST 响应包括内联线id性的属性值Creative。 MediaTailor 从自定义 VAST Extension 元素中提取值，并将该值放在清单的素材元数据中。

```
<?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|
TVNlDDNpFTchtpRj,E5TfTtcYd5IEzvEt,ChA05OHcvWRGFY6Zp5VSSlxUEJ2B9p8GGhQIDzIQkFeQC-
Ho67FR3P9qNa6khSAGKgAyAA]]></Extension>
          </Extensions>
        </InLine>
    </Ad>
</VAST>
```

客户端跟踪 API

以下示例展示了玩家 SDK 如何使用和将清单中的广告元数据与客户端跟踪响应负载中的完整跟踪事件数据关联`creativeId`起来`adId`。

Example JSON 消息：

```
{
  "avails": [
    {
      "adBreakTrackingEvents": [],
      "ads": [
        {
          "adId": "5",
          "adParameters": "",
          "adProgramDateTime": null,
          "adSystem": "2.0",
          "adTitle": "AD-caribbean2-15",
          "adVerifications": [],
          "companionAds": [],
          "creativeId": "1234",
          "creativeSequence": "2",
          "duration": "PT15S",
          "durationInSeconds": 15,
          "extensions": [],
          "mediaFiles": {
            "mediaFilesList": [],
            "mezzanine": ""
          },
          "skipOffset": null,
          "startTime": "PT30S",
          "startTimeInSeconds": 30,
          "trackingEvents": [
            {
              "beaconUrls": [
                "https://myServer/impression"
              ],
              "duration": "PT15S",
              "durationInSeconds": 15,
              "eventId": "5",
              "eventProgramDateTime": null,
              "eventType": "impression",
              "startTime": "PT30S",
```

```
              "startTimeInSeconds": 30
            }
          ],
          "vastAdId": ""
        }
      ],
      "availId": "5",
      "availProgramDateTime": null,
      "duration": "PT15S",
      "durationInSeconds": 15,
      "meta": null,
      "nonLinearAdsList": [],
      "startTime": "PT30S",
      "startTimeInSeconds": 30
    }
  ],
  "nextToken": "UFQ1TTM0Ljk2N1NfMjAyMi0xMS0xOFQwNDozMzo1Mi4yNDUxOTdaXzE%3D",
  "nonLinearAvails": []
}
```

使用 AWS Elemental MediaTailor 创建线性组装流

AWS Elemental MediaTailor 频道汇编是一项仅限清单的服务，允许您使用现有的视频点播 (VOD) 内容和直播内容来创建线性流媒体频道。 MediaTailor 切勿触及您的内容片段，这些区段直接由您的原始服务器提供。取而代之的是，从您的来源 MediaTailor 获取清单，然后使用它们来组装一个引用基础内容段的实时滑动清单窗口。频道汇编会跟踪诸如媒体序列号之类的内容，这是使资产之间流畅播放所必需的。线性汇编流是使用现有的多比特率编码和打包的 VOD 内容以较低的运行成本创建的。

通过在节目中插入广告插播时间，您可以轻松地通过频道汇编线性直播获利，而不必使用 SCTE-35 标记来限制内容。您可以将频道汇编与 MediaTailor 广告插入服务或任何服务器端广告插入服务一起使用。

要开始使用频道汇编，请参阅the section called “MediaTailor 频道汇编入门”。

有关将 Channel Assembly 与内容分发网络 (CDN) 集成的信息，请参阅使用频道集合和 CDN 构建 MediaTailor 线性频道。

主题

- 渠道汇编工作流程类型
- 使用来源位置
- 使用通道
- 将节目添加到频道的日程安排中
- 在频道直播中插入个性化广告和广告插播时间
- 对频道的播放进行时移

渠道汇编工作流程类型

AWS Elemental MediaTailor 频道汇编支持三种用于创建线性流媒体频道的工作流程类型。每种工作流程类型使用不同的源内容组合来满足特定的用例。本主题描述了每种工作流程类型、何时使用它以及详细设置过程的链接。

工作流程类型	说明	回放模式	使用案例
VOD-to-live	将预先录制的 VOD 资源汇编成线性直播	循环或线性	24/7 来自内容库的频道

工作流程类型	说明	回放模式	使用案例
Live-to-live	在预定频道中以节目形式传送直播源源	仅限线性	新闻、体育、区域节目
VOD + Live-to-live（混合）	将视频点播（VOD）和直播源整合到一个频道的时间表中	仅限线性	主要包含 VOD 内容（包括预定直播活动）的频道

VOD-to-live

VOD-to-live 频道完全使用预编码、预打包的 VOD 内容创建线性直播。直播看起来像是带有实时滑动清单窗口的直播，但仅使用 VOD 资产。你不需要在源内容中添加 SCTE-35 标记，因为你在安排节目时定义广告时段。

何时使用

- 你有一个 VOD 资源库，想要创建一个 24/7 线性频道。
- 您想通过广告插播时间通过内容获利，但源内容中没有 SCTE-35 标记。
- 你需要一个简单、低成本的线性流，不需要实时编码基础设施。

工作原理

1. 准备您的内容 — 对您的 VOD 资源进行编码和打包。将它们托管在源服务器（亚马逊 S3 或任何 HTTP 源）上。 AWS Elemental MediaPackage
2. 创建源位置 — 向其注册您的源服务器 MediaTailor。
3. 添加 VOD 来源 — 注册每段内容并定义其包配置和源组。
4. 创建频道 — 选择播放模式（循环或线性）并配置输出。
5. 安排节目 — 按照您希望的播放顺序将节目添加到频道的时间表中。
6. 配置广告插播时间 -向每个程序添加广告插播时间。连接到 SSAI 配置以投放个性化广告。
7. 启动您的频道并配置 CDN — 启动频道。（可选）在其前面添加 CDN 以节省规模和成本。

选择播放模式

VOD-to-live 频道支持两种播放模式：

- 循环（基本层）— 程序以无休止的循环方式连续播放。当最后一个程序完成时，播放会返回到第一个节目。这是 24/7 频道最简单的选项。循环模式不支持直播源或填充列表。
- 线性（标准层）— 程序按顺序播放一次。线性模式支持基于时间的调度，并可填补程序间的空白。

内容要求

为了实现平稳过渡，同一频道中的所有 VOD 资源都必须具有：

- 源组中所有源的格式副本（视频、音频、字幕等）数量相同
- 所有资产的区段持续时间一致。我们建议 2 到 6 秒。
- 匹配相应格式副本的比特率和分辨率

> **Important**
>
> 频道汇编不支持每标题编码或自动 ABR 编码。这些方法生成的资源具有不同的再现次数和持续时间，这会导致程序过渡时播放失败。

广告中断和获利

在日程安排中添加节目时，您可以定义广告时段。每个广告插播时间都需要从程序开始算起以毫秒为单位的偏移量。偏移量必须与 VOD 源中所有轨道的分段边界对齐。

您的 VOD 源清单可能包含广告标记。对于 HLS 来说，这些是带有持续时间的`EXT-X-DATERANGE`或`EXT-X-CUE-OUT`标签。对于 DASH，请使用`EventStream`。当 MediaTailor 检测到这些标记时，它会在您设置广告插播时将它们的偏移量显示为选项。这提供了最精确的广告投放位置。

要投放个性化广告，请将您的频道与 MediaTailor SSAI 播放配置配对。将内容源设置为您频道的播放 URL。

提示

Topic	Detail
广告插播时间	与区段边界不对齐的广告插播时间偏移量会调整到最近的边界。使用较短的分段（2-4 秒）来获得更精确的广告投放。

Topic	Detail
签名网址来源	MediaTailor 首次添加 VOD 源清单时会缓存 VOD 源清单。如果您使用过期时间较短的签名网址，则缓存的网址可能会在播放前过期。调用 `UpdateVodSource` API 强制 MediaTailor 重新获取清单。
格式副本不匹配	如果节目无法完全过渡，请检查频道中的所有 VOD 源是否具有相同的再现次数，并使用相同的 ABR 阶梯。
提前安排	在节目播出时间前至少 10 分钟将节目添加到正在运行的线性频道。信道汇编开始在此窗口内处理程序。

有关创建第一个频道的说明，请参阅the section called “MediaTailor 频道汇编入门”。

Live-to-live

直播频道作为预定频道中的节目通过直播源提要传递。该频道根据您的日程安排在不同的直播源之间切换。当您的所有内容均已上线，并且您想将多个直播源汇编成一个线性频道时，请使用此工作流程。

何时使用

- 你有多个直播 encoders/packagers ，想按计划在它们之间切换。
- 您正在建立一个区域频道，可以在一天中的不同时间在全国和地区直播之间切换。
- 你有 24/7 直播（新闻、体育网络），想创建一个精选的频道，从多个来源中汲取灵感。

工作原理

1. 设置实时编码管道 — 使用 AWS Elemental MediaLive、 AWS Elemental MediaPackage或任何实时打包来源对直播内容进行编码和打包。
2. 创建源位置 -为您的直播内容注册原始服务器 MediaTailor。
3. 添加直播源 — 注册每个直播源并定义其包配置和源组。
4. 创建频道 — 选择线性播放模式（直播源必需）并配置输出。
5. 安排节目 -将直播源节目添加到频道的时间表中。指定每个直播节目的时长。
6. 启动您的频道并配置 CDN — 启动频道。（可选）在其前面添加 CDN 以节省规模和成本。

回放模式

Live-to-live 频道需要线性（标准层）播放模式。循环模式不支持直播源。

内容要求

除了以下列出的一般要求外the section called “使用来源位置”：

- HLS 实时来源 — 您必须为清单窗口中的第一个片段以及每次不连续时提供#EXT-X-PROGRAM-DATE-TIME标签。
- HLS 广告标记 — 您必须将广告标记配置为DATERANGE。
- 源清单窗口 — 使用清单窗口持续时间至少比频道集合频道的清单窗口长 30 秒。
- 目标持续时间 — 使目标持续时间与频道中现有源的持续时间相匹配。
- 子播放列表 —使子播放列表的数量与频道中其他来源的数量相匹配。

使用实时源代码配置 MediaPackage

如果您使用 AWS Elemental MediaPackage 作为包装来源，我们建议使用以下配置：

HLS 设置：

设置	值	必要性	注意
端点类型	Apple HLS 或 CMAF	必填	用Apple HLS于 TS 片段，用CMAF于 MP4 片段
ProgramDateTimeIntervalSeconds	1	必填	防止中断时出现播放问题。
PlaylistWindowSeconds	比 CA 清单窗口长 30 秒	必填	
AdMarkers	DATERANGE	必填	通过广告标记时为必填项。

设置	值	必要性	注意
IncludeIframeOnlyStream	已禁用	推荐	

DASH 设置：

设置	值	必要性	注意
ManifestLayout	FULL	推荐	
SegmentTemplateFormat	NUMBER_WITH_TIMELINE 或 TIME_WITH_TIMELINE	推荐	不支持 NUMBER_WITH_DURATION 。
ManifestWindowSeconds	比 CA 清单窗口长 30 秒	必填	
PeriodTriggers	ADS	必填	通过广告标记时为必填项。

广告时段

对于直播来源，您无需在频道集合计划中定义广告时段。取而代之的是，实时来源清单中的广告插播时间是自动传递的：

- HLS — 使用EXT-X-DATERANGE标签的广告插播间隔被传递。
- DASH — 使用广告插播EventStream时段会被传递。

要通过直通广告时段获利，请将您的频道与 SSAI 播放配置配对。 MediaTailor

提示

Topic	Detail
计划持续时间	必须为每个直播源程序指定持续时间。持续时间到期后，频道集会将过渡到计划中的下一个节目。
填充板	如果时间表中的节目之间存在间隔，则按 MediaTailor 顺序排列。如果石板的持续时间小于间隙的持续时间，则 MediaTailor 循环播放石板。在创建频道时配置填充列表。
区域渠道	使用直播资源创建区域频道。 encoder/packager 为全国性内容运行一个编码器，然后在这些区域上线时运行区域编码器。为每个地区创建一个单独的频道，每个频道都有自己的时间表。
过渡时机	MediaTailor 在公共片段边界上开始播放节目。对于绝对过渡，预计计划时间和实际过渡之间最多有一段差异。
编码成本	仅在内容上线时运行编码器，从而最大限度地降低成本。在您的频道上配置填充表，以填补直播活动之间的空白。

VOD + Live-to-live （混合）

混合频道将VOD和直播源合并到一个时间表中。这是最灵活的工作流程类型。当您的频道主要是穿插预定直播活动的 VOD 内容，或者您想弥合直播活动与预先录制的内容之间的差距时，请使用它。

何时使用

- 您的频道主要是VOD内容，但您已经安排了直播活动（晚间新闻，每周体育节目，特别广播）。
- 你想全天在 VOD 填充内容和直播间切换。
- 你正在建立一个在直播活动窗口之间使用预先录制的内容的频道。

工作原理

1. 设置内容管道 — 对您的 VOD 资产和直播内容进行编码和打包。将它们托管在原始服务器上。
2. 创建来源位置 -向注册您的源服务器 MediaTailor。您可以将单独的源位置用于 VOD 和直播内容，如果它们共享来源，则可以使用单一来源位置。

3. 添加 VOD 源和直播源 — 注册您的 VOD 资源和直播源，每个资源和直播源都有各自的包配置和源组。
4. 创建频道 — 选择线性回放模式（使用直播源时为必填项）并配置输出。
5. 安排节目 — 将 VOD 和直播源节目的混合添加到频道的时间表中。对于必须在特定时间开始的直播节目，使用绝对过渡。对填补空白的 VOD 程序使用相对过渡。
6. 配置广告插播时间 -向 VOD 程序添加广告插播时间。对于直播节目，广告插播时间是从源清单中传递的。
7. 启动您的频道并配置 CDN — 启动频道。或者，在其前面添加一个 CDN 以节省规模和成本。

回放模式

混合通道需要线性（标准层）播放模式。循环模式不支持直播源。

内容要求

混合频道必须满足 VOD 和直播源的内容要求：

- 源组中的所有来源（VOD 和直播）必须具有相同数量的格式副本。
- VOD 和直播源必须具有匹配的目标片段时长。
- 直播源必须满足直播源的要求，包括 HLS 的`EXT-X-PROGRAM-DATE-TIME`标签。

日程安排模式

混合频道支持两种节目的过渡类型：

- 相对 — 该程序在时间表中的另一个节目之前或之后播放。仅适用于 VOD 源程序。
- 绝对 — 该程序在特定的挂钟时间播放。适用于线性模式下的 VOD 和直播源程序。

常见的调度模式：

模式	说明	示例
带有预定直播窗口的 VOD 填充器	VOD 节目排满了日程安排，直播节目会在特定时间插入。	一个电影频道，在晚上 8:00 切换到体育赛事直播。

模式	说明	示例
使用 VOD 插页式广告进行初级直播	直播大部分时间都在运行，VOD 节目在休息时间或非工作时间运行。	一个新闻频道，在直播之间播放预先录制的片段。
Time-of-day 切换	不同的来源（直播或 VOD）在一天中的不同时间播放。	黄金时段全国直播，早上区域直播，隔夜VOD内容。

广告时段

广告插播行为取决于每个程序的来源类型：

- VOD 程序 — 您可以在时间表中定义广告时段，偏移量以毫秒为单位。连接到 SSAI 配置以投放个性化广告。
- 直播节目 — 广告插播时间自动从直播源清单中传递。HLS 使用`EXT-X-DATERANGE`标签；达世币使用`EventStream`。

提示

Topic	Detail
从 VOD 过渡到直播	在绝对时间从 VOD 节目过渡到直播节目时， MediaTailor 会在距离直播节目的开始时间最近的公共分段边界处截断 VOD 节目。
从直播过渡到 VOD	当直播节目的持续时间到期时，将 MediaTailor 过渡到计划中的下一个节目。如果下一个节目是VOD，则只要满足内容要求，就可以无缝播放。
填充板	在您的频道上配置填充列表。 MediaTailor 在程序之间出现任何间隙时播放 slate。如果直播源在节目中途不可用，Channel Assembly将继续提供最后的已知片段，直到节目持续时间到期。
混合广告获利	您可以对整个频道使用单个 SSAI 配置。 MediaTailor 处理同一个直播中的预定广告插播时间（VOD 节目）和直通广告时段（直播节目）。
灵活安排时间	至少在播出时间前 10 分钟添加 VOD 节目。对于使用绝对过渡的直播节目，请至少提前 15 分钟在正在播出的频道上安排这些节目。

选择正确的工作流程

使用以下决策指南为您的用例选择正确的工作流程类型：

- 如果您只有 VOD 内容且不需要基于时间的日程安排 → 在循环播放模式下VOD-to-live使用。
- 如果您只有 VOD 内容且需要基于时间的日程安排 → VOD-to-live 使用线性播放模式。
- 如果您只有直播内容 → 使用Live-to-live。
- 如果你同时有 VOD 和直播内容 → 使用 VOD + Live-to-live （混合）。

功能对比

功能	VOD-to-live（循环）	VOD-to-live（线性）	Live-to-live	混合
VOD 来源	支持	是	否	是
直播来源	否	否	是	是
绝对过渡	否	是	是	是
相对过渡	支持	是	否	是（仅限 VOD）
预定广告时段	支持	是	否	是（仅限 VOD）
Pass-through 广告插播时间	否	否	是	是（仅限直播）
填充板	否	是	是	是
24/7 循环	是	否	否	否
Time-based 日程安排	否	是	是	是
频道等级	Basic	标准	标准	标准

后续步骤

- the section called “MediaTailor 频道汇编入门”— 创建第一个频道的 Step-by-step 教程
- the section called “使用来源位置”— 管理源位置和内容的详细程序
- the section called “使用通道”— 频道配置选项和播放模式
- the section called “添加程序”— 使用 VOD 和直播资源调度节目
- the section called “ 插入广告和广告插播时间”— 使用 SSAI 获利
- the section called “使用 CDN 进行频道组装”— 整合 CDN 以节省规模和成本
- the section called “对频道的播放进行时移”— 启用追赶和重新启动功能

使用来源位置

源位置表示存储您的源内容的源服务器。来源位置可以是 Amazon S3、标准 Web 服务器、内容分发网络 (CDN)（如亚马逊 CloudFront）或包装来源（例如）。 AWS Elemental MediaPackage MediaTailor 从源位置检索您的内容清单，并使用它们来整理频道的线性直播。

本主题介绍如何使用 AWS Elemental MediaTailor 控制台创建和删除来源位置，以及如何使用 VOD 来源。

主题

- 创建源位置
- 为您的来源位置配置身份验证
- 使用 VOD 来源
- 使用实时信号源
- 使用软件包配置
- 清单缓存

创建源位置

以下过程说明了如何使用 MediaTailor控制台创建源位置。有关如何使用 MediaTailor API 创建来源位置的信息，请参阅 AWS Elemental MediaTailor API 参考CreateSourceLocation中的。

创建源位置

1. 打开 MediaTailor 控制台，网址为https://console.aws.amazon.com/mediatailor/。

2. 在导航窗格中，选择频道集合 > 来源位置。
3. 在导航栏上，选择创建来源位置。
4. 在源位置配置下，输入源服务器的名称和基本 URL：

   - 名称：您的来源位置的标识符，例如 my- origin。
   - 基本 URL：存储内容的源服务器的协议和基本 URL，例如https://111111111111.cloudfront.net。网址必须采用标准的 HTTP 网址格式，前缀为 ht tp://或 h ttps://。

     或者，如果您的源位置是 Amazon S3 存储桶，并且您想使用 AWS 签名版本 4 进行 Amazon S3 访问身份验证，请选择 “使用 Sigv 4 进行 Amazon S3 身份验证”。有关高级信息，请参阅为您的来源位置配置身份验证。

5. 在 “访问配置” 下，可以选择为您的来源位置配置身份验证：

   - 访问类型：选择 MediaTailor 用于访问存储在来源位置来源上的内容的身份验证类型。
     - 适用于亚马逊 S3 的 Sigv4-使用 MediaTailor 亚马逊签名版本 4 (Sigv4) 来授权向你的来源发出的请求。有关更多信息，请参阅 the section called “使用 Sigv4 对发往亚马逊 S3 的请求进行身份验证”。
     - S@@ ecrets Manager 访问令牌身份验证- MediaTailor 使用 Secrets Manager 以及由您创建、拥有和管理的 AWS KMS 客户托管密钥来简化与您的源 MediaTailor 之间的访问令牌身份验证。有关如何配置 S ecrets Manager 访问令牌身份验证的信息，请参阅the section called “与 AWS Secrets Manager 访问令牌身份验证”。
       - 标头名称-指定 HTTP 标头名称。 MediaTailor 使用 HTTP 标头在内容清单请求中将访问令牌发送到您的源。你可以使用任何标头名称，只要它不是以`x-amz-`或开头的`x-amzn-`。如果您要与 MediaPackage CDN 授权集成，则标头值应`X-MediaPackage-CDNIdentifier`为。
       - 密钥字符串密钥-您在 Secrets Manager 密`SecretString`钥中指定的密钥。例如，如果`SecretString`包含密钥和值对，例如:`{"MyHeaderName": "11111111-2222-3333-4444-111122223333"}`，则`MyHeaderName`是您在此字段中输入的`SecretString`密钥。
       - 秘密 ARN-存放您的访问令牌的机密的 ARN。有关分步指南，请参阅步骤 2：创建一个 AWS Secrets Manager 机密密钥。
6. 在 “区段交付服务器配置” 下，可以选择配置服务器来交付您的内容分段：

- 使用默认的分段传送服务器：输入用于交付内容分段（例如 CDN）的服务器的基本 URL。如果您想使用与源位置服务器不同的服务器来提供内容分段，请配置默认分段主机名。例如，您可以对基本 HTTP URL（用于访问清单的内容）和默认区段基础 URL（玩家用于访问内容分段） MediaTailor 使用不同的 CDN 配置，来限制玩家对原始清单的访问。如果未输入值，则 MediaTailor 默认为分段传送的源位置服务器。
- 使用指定分段传送服务器：如果您配置了默认的分段传送服务器，则还可以配置其他分段传送服务器。每一个都必须有一个唯一的名称和一个基本 URL。基本网址可以是完整的 HTTP 网址，也可以是类似的相对路径`/some/path/`。这些名称用于标识在 MediaTailor 收到内容分段请求时应使用哪个服务器。如果请求包含标头`X-MediaTailor-SegmentDeliveryConfigurationName`并且标头的值与名称匹配，则将使用相应的基本 URL 来提供内容。如果请求中未包含标头，或者标头与任何名称都不匹配，则将使用默认的分段传送服务器。

7. 选择 “创建源位置”。
8. 要添加更多源位置，请重复步骤 2-6。

为您的来源位置配置身份验证

使用访问配置为您的来源位置配置身份验证。开启访问配置后， MediaTailor 只有在请求在 MediaTailor 和您的来源之间获得授权的情况下，才会从您的来源检索源清单。默认情况下，访问配置处于关闭状态。

MediaTailor 支持以下身份验证类型：

- 适用于 Amazon S3 身份验证的 Sigv4
- AWS Secrets Manager 访问令牌
- 用于 MediaPackage 版本 2 (v2) 身份验证的 Sigv4

本章介绍如何使用适用于 Amazon S3、v MediaPackage 2 的 Sigv4 以及如何使用 AWS Secrets Manager 访问令牌进行来源位置身份验证。

有关更多信息，请选择适用的主题。

主题

- 使用 Sigv4 对发往亚马逊 S3 的请求进行身份验证
- 在版本 2 中使用 SigV4 MediaPackage

- 与 AWS Secrets Manager 访问令牌身份验证

使用 Sigv4 对发往亚马逊 S3 的请求进行身份验证

适用于亚马逊 S3 的签名版本 4 (Sigv4) 是一种签名协议，用于对通过 HTTPS 向亚马逊 S3 发出的请求进行身份验证。当您将 Sigv4 用于 Amazon S3 时，在向用作源的 Amazon S3 存储桶的 HTTPS 请求中 MediaTailor 包含签名的授权标头。如果签名的授权标头有效，则您的来源满足请求。如果无效，则请求失败。

有关 sigV4 的一般信息 AWS Key Management Service，请参阅 Amazon S3 API 参考中的身份验证请求（AWS 签名版本 4）主题。

> **Note**
>
> MediaTailor 始终使用 Sigv4 签署对这些来源的请求。

要求

如果您为来源位置激活 SigV4 for Amazon S3 身份验证，则必须满足以下要求：

- 您必须通过在 IAM 中授予 mediatailor.amazonaws.com 委托人访问权限来允许 MediaTailor 访问您的亚马逊 S3 存储桶。有关在 IAM 中配置访问权限的信息，请参阅AWS Identity and Access Management 用户指南中的访问管理。
- m ediatailor.amazonaws.com 服务主体必须有权读取 VOD 源包配置引用的所有多变体播放列表。
- API 的调用者必须具有 s3: GetObject IAM 权限才能读取您的 MediaTailor VOD 源包配置引用的所有多变体播放列表。
- 您的 MediaTailor 来源位置基本 URL 必须遵循 Amazon S3 虚拟托管式请求网址格式。例如，https://*bucket-name*.s3。*Region*.amazonaws.com/。*key-name*有关 Amazon S3 托管的虚拟式访问权限的信息，请参阅虚拟 Hosted-Style 请求。

在版本 2 中使用 SigV4 MediaPackage

v2 的签名版本 4 (Sigv4) 是一种签名协议，用于对通过 HT MediaPackage TP 向 MediaPackage v2 发出的请求进行身份验证。当你在 v MediaPackage 2 中使用 Sigv4 时，在发送给用作源的 MediaPackage v2 端点的 HTTP 请求中 MediaTailor包含签名的授权标头。如果签名的授权标头有效，则您的来源满足请求。如果无效，则请求失败。

有关 v2 版 Sigv4 的一般信息，请参阅 MediaPackage v2 API 参考中的身份验证请求（AWS 签名版本 4）主题。MediaPackage

要求

如果您为源位置激活 sig MediaPackage V4 进行 v2 身份验证，则必须满足以下要求：

- 您必须通过在终端节点的 O MediaTailor rigin 访问策略中授予 mediatailor.amazonaws.com 委托人访问权限来允许访问您的 MediaPackage v2 终端节点。
- 您的 MediaTailor 来源位置基础 URL 必须是 MediaPackage v2 终端节点。
- API 的调用者必须具有 mediapackagev2: GetObject IAM 权限才能读取源打包配置引用的所有多变体播放列表。 MediaTailor

与 AWS Secrets Manager 访问令牌身份验证

MediaTailor 支持 S ecrets Manager 访问令牌身份验证。通过 AWS Secrets Manager 访问令牌身份验证， MediaTailor使用 AWS Key Management Service (AWS KMS) 客户托管的 AWS Secrets Manager 密钥和您创建、拥有和管理的密钥来对发往您的来源的请求进行身份验证。

在本节中，我们将介绍 Secrets Manager 访问令牌身份验证的工作原理，并提供有关如何配置 Secrets Manager 访问令牌身份验证的分步信息。你可以在中使用 Secrets Manager 访问令牌身份验证， AWS 管理控制台 也可以通过编程方式使用 AWS API。

主题

- 配置 AWS Secrets Manager 访问令牌身份验证
- 与使用 CDN 授权的 MediaPackage 端点集成
- S MediaTailor ecrets Manager 访问令牌身份验证的工作原理

配置 AWS Secrets Manager 访问令牌身份验证

要使用 AWS Secrets Manager 访问令牌身份验证时，请执行以下步骤：

1. 您创建 AWS Key Management Service 客户管理的密钥。
2. 您创建了一个 AWS Secrets Manager 密钥。该密钥包含您的访问令牌，该令牌作为加密密钥值存储在 Secrets Manager 中。 MediaTailor 使用 AWS KMS 客户管理的密钥来解密密密钥值。
3. 您可以将 AWS Elemental MediaTailor 源位置配置为使用 Secrets Manager 访问令牌身份验证。

以下部分提供有关如何配置 AWS Secrets Manager 访问令牌身份验证的分步指导。

主题

- 步骤 1：创建一个 AWS KMS 对称客户托管密钥
- 步骤 2：创建一个 AWS Secrets Manager 机密密钥
- 步骤 3：使用访问令牌身份验证配置 MediaTailor 源位置

步骤 1：创建一个 AWS KMS 对称客户托管密钥

您用来 AWS Secrets Manager 以存储在密钥中的形式`SecretString`存储访问令牌。使用您创建、拥有和管理的AWS KMS 对称客户托管密钥进行加密。`SecretString` MediaTailor 使用对称的客户托管密钥来简化对密钥的访问（通过授权），并加密和解密密密钥值。

客户托管密钥允许您执行以下任务：

- 制定和维护关键策略
- 建立和维护 IAM 策略和授权
- 启用和禁用密钥策略
- 轮换加密密钥材料
- 添加 标签

  有关 Secrets Manager 如何使用 AWS KMS 来保护密钥的信息，请参阅《AWS Key Management Service 开发者指南》 AWS KMS中的 “如何 AWS Secrets Manager 使用” 主题。

  有关客户托管密钥的更多信息，请参阅《AWS Key Management Service 开发人员指南》中的客户托管密钥。

> **Note**
>
> AWS KMS 使用客户托管密钥需要支付费用有关定价的更多信息，请参阅定AWS Key Management Service 价页面。

您可以使用 AWS 管理控制台 或通过 API 以编程方式创建 AWS KMS 对称客户托管密钥。 AWS KMS

创建对称的客户托管密钥

按照《AWS Key Management Service 开发人员指南》中创建对称客户托管密钥的步骤进行操作。

记下关键的 Amazon 资源名称 (ARN)；你需要它。步骤 2：创建一个 AWS Secrets Manager 机密密钥

加密上下文

加密上下文是一组可选的键值对，包含有关数据的其他上下文信息。

Secrets Manager 在加密和解密时包含加密上下文。SecretString加密上下文包括机密 ARN，它将加密限制在该特定机密范围内。作为一项额外的安全措施，请代表您 MediaTailor 创建 AWS KMS 补助金。 MediaTailor 应用的GrantConstraints操作仅允许我们解密与 S ecrets Manager 加密上下文中包含的机密 ARN SecretString 关联的 ARN。

有关 Secrets Manager 如何使用加密上下文的信息，请参阅AWS Key Management Service 开发者指南中的加密上下文主题。

设置密钥策略

密钥政策控制对客户托管密钥的访问。每个客户托管式密钥必须只有一个密钥策略，其中包含确定谁可以使用密钥以及如何使用密钥的声明。创建客户托管密钥时，您可以使用默认密钥策略。有关更多信息，请参阅《AWS Key Management Service 开发人员指南》中的 AWS KMS的身份验证和访问控制。

要将您的客户托管密钥与您的 MediaTailor 来源位置资源一起使用，您必须向调用CreateSourceLocation或使用UpdateSourceLocation以下 API 操作的 IAM 委托人授予权限：

- kms:CreateGrant— 向客户托管密钥添加授权。 MediaTailor 为您的客户托管密钥创建授权，允许其使用该密钥创建或更新配置了访问令牌身份验证的来源位置。有关在中使用 Grant s 的更多信息 AWS KMS，请参阅《AWS Key Management Service 开发者指南》。

  这 MediaTailor 允许执行以下操作：

  - 致电Decrypt这样它就可以在呼叫时成功检索你的 Secrets Manager 密钥GetSecretValue。
  - 当源位置被删除或密钥的访问权限被撤销时，请致电RetireGrant取消授权。

以下是您可以为其添加的策略声明示例 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"
        }
    }
}
```

有关在策略中指定权限以及对密钥访问进行故障排除的更多信息，请参阅《AWS Key Management Service 开发人员指南》AWS KMS中的授权。

步骤 2：创建一个 AWS Secrets Manager 机密密钥

使用 Secrets Manager 以 AWS KMS 客户托管密钥加密`SecretString`的形式存储您的访问令牌。MediaTailor使用密钥来解密。`SecretString`有关 Secrets Manager 如何使用 AWS KMS 来保护密钥的信息，请参阅《AWS Key Management Service 开发者指南》 AWS KMS中的 “如何 AWS Secrets Manager 使用” 主题。

如果您使用源位置来源，并且想使用 S MediaTailor ecret AWS Elemental MediaPackage s Manager 访问令牌身份验证，请按照以下步骤操作the section called “与使用 CDN 授权的 MediaPackage 端点集成”。

你可以使用 AWS 管理控制台 或通过 Secrets Manager API 以编程方式创建 Secrets Manager 密钥。

创建密钥

按照《AWS Secrets Manager 用户指南》 AWS Secrets Manager中创建和管理密钥的步骤进行操作。

创建密钥时，请记住以下注意事项：

- KmsKeyId必须是您在步骤 1 中创建的客户托管密钥的密钥 ARN。
- 您必须提供 SecretString. `SecretString`应该是有效的 JSON 对象，其中包含包含访问令牌的键和值。例如，{” MyAccessTokenIdentifier “: “112233445566"}。该值的长度必须介于 8-128 个字符之间。

  使用访问令牌身份验证配置源位置时，需要指定`SecretString`密钥。 MediaTailor 使用密钥查找和检索存储在中的访问令牌`SecretString`。

  记下秘密 ARN 和密钥。`SecretString`当您将来源位置配置为使用访问令牌身份验证时，您将使用它们。

附加基于资源的密钥策略

要允许 MediaTailor 访问密钥值，您必须将基于资源的策略附加到该密钥。有关更多信息，请参阅AWS Secrets Manager 用户指南中的向 Secrets Manager 密钥附加权限策略。

以下是您可以为其添加的策略声明示例 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"
        }
    ]

}
```

步骤 3：使用访问令牌身份验证配置 MediaTailor 源位置

你可以使用 MediaTailor API AWS 管理控制台 或以编程方式配置 Secrets Manager 访问令牌身份验证。

使用 Secrets Manager 访问令牌身份验证配置源位置

按照《AWS Elemental MediaTailor 用户指南》Access configuration中的步骤操作。

与使用 CDN 授权的 MediaPackage 端点集成

如果您使用 AWS Elemental MediaPackage 源位置来源，则 MediaTailor 可以与使用 CDN 授权的 MediaPackage 端点集成。

要与使用 CDN 授权的 MediaPackage 终端节点集成，请按以下步骤操作。

要与集成 MediaPackage

1. 完成AWS Elemental MediaPackage 用户指南中设置 CDN 授权中的步骤（如果尚未完成）。
2. 完成the section called “步骤 1：创建一个 AWS KMS 对称客户托管密钥”中的过程。
3. 修改您在设置 MediaPackage CDN 授权时创建的密钥。使用以下值修改密钥：
   - `KmsKeyId`使用您在中创建的客户托管密钥 ARN 更新。the section called “步骤 1：创建一个 AWS KMS 对称客户托管密钥”
   - （可选）对于`SecretString`，您可以将 UUID 轮换为新值，也可以使用现有的加密密钥，前提是它是标准 JSON 格式的密钥和值对，例如`{"MediaPackageCDNIdentifier": "1122334455667788 99"}`。
4. 完成 the section called “附加基于资源的密钥策略” 中的步骤。
5. 完成 the section called “步骤 3：使用访问令牌身份验证配置 MediaTailor 源位置” 中的步骤。

S MediaTailor ecrets Manager 访问令牌身份验证的工作原理

在创建或更新源位置以使用访问令牌身份验证后，在从您的来源请求源内容清单时，请将访问令牌 MediaTailor 包含在 HTTP 标头中。

以下是如何 MediaTailor 使用 Secrets Manager 访问令牌身份验证进行来源位置来源身份验证的概述：

1. 当您创建或更新使用访问令牌身份验证的 MediaTailor 源位置时， MediaTailor 会向 Secrets Manager 发送DescribeSecret请求以确定与该密 AWS KMS 钥关联的密钥。您在来源位置访问配置中包含机密 ARN。
2. MediaTailor 为客户管理的密钥创建授权，以便该密钥 MediaTailor 可以使用该密钥访问和解密存储在中的访问令牌。 SecretString拨款名称将是`MediaTailor-SourceLocation-`*`your AWS ## ID-source location name`*。

   您可以随时撤销对授予的访问权限，也可以取消 MediaTailor对客户托管密钥的访问权限。有关更多信息，请参阅《AWS Key Management Service API Reference》中的 RevokeGrant。
3. 创建、更新 VOD 源或在程序中使用 VOD 源时， MediaTailor会向源位置发出 HTTP 请求，以检索与源位置中的 VOD 源关联的源内容清单。如果 VOD 源与配置了访问令牌的来源位置相关联，则请求会将访问令牌作为 HTTP 标头值。

使用 VOD 来源

VOD 来源代表您添加到来源位置的单个内容，例如视频或播客剧集。您可以将一个或多个 VOD 源添加到您的源位置，然后在创建频道后将每个 VOD 来源与节目关联。

每个 VOD 来源必须至少有一个软件包配置。包配置为您的 VOD 来源指定包格式、清单位置和来源组。创建频道时，您可以使用包配置的源组在频道上创建相应的输出。例如，如果你的源代码以两种不同的格式（HLS 和 Dash）打包，那么你需要创建两个包配置，一个用于 DASH，一个用于 HLS。然后，您将创建两个通道输出，每个通道输出对应一个包配置。每个频道的输出都提供了一个用于播放请求的端点。因此，使用前面的示例，该频道将为 HLS 播放请求提供端点，为 DASH 播放请求提供端点。

如果您想自动检测清单中广告标记的偏移量，则每个广告标记在所有套餐配置中必须以相同的偏移量显示，并且持续时间为零。对于 HLS， MediaTailor 将检测DATERANGE并 EXT-X-CUE-OUT标记。对于 DASH，HLS 将检测每个标签中的第一个事件EventStream标签。

在以下示例中，由于DATERANGE标签的持续时间为 0.0，因此将在 12000 毫秒的偏移量处检测到广告中断机会。偏移量为 0ms 的第一个DATERANGE标签不会被检测到，因为它的持续时间为 10.0。

```
#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
```

在以下示例中，将在 0 毫秒的偏移量时检测到广告中断机会，因为该EXT-X-CUE-OUT标签的持续时间为 0，紧随其后的是EXT-X-CUE-IN标签。第二个EXT-X-CUE-OUT/EXT-X-CUE-IN对不会被检测到，因为它的持续时间为 10。

```
#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
```

在以下示例中，将在 0 毫秒的偏移量时检测到广告中断机会，因为中的第一个事件EventStream发生在 PT0.000S开始的时间段内。EventStream将无法检测到Event中的第二个。

```
<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>
```

将 VOD 源添加到您的来源位置

以下过程说明了如何使用 MediaTailor 控制台将 VOD 源添加到您的来源位置并设置软件包配置。有关如何使用 API 添加 VOD 源的信息，请参阅 MediaTailor AP AWS Elemental MediaTailor I 参考CreateVodSource中的。

Important

在添加 VOD 源之前，请确保它们满足以下要求：

- 源变体必须具有相同的长度，具体取决于源清单的长度。
- 在包配置中，每个源必须具有相同数量的子流。

由于这些要求，我们不支持按标题或自动 ABR，因为这些编码方法可能会生成不同的清单长度和子流。
我们建议您使用包含最小段长度的编码模板，以确保您的编码源满足这些要求。

将 VOD 源添加到您的来源位置

1. 打开 MediaTailor 控制台，网址为https://console.aws.amazon.com/mediatailor/。
2. 在导航窗格中，选择频道集合 > 来源位置。
3. 在来源位置窗格中，选择您在创建源位置过程中创建的来源位置。
4. 选择 “添加 VOD 来源”。
5. 在 VOD 来源详细信息下，输入您的 VOD 来源的名称：
   - 名称：您的 VOD 来源的标识符，例如 m y-example-video。
6. 在 P ackage 配置 > 下，*source-group-name*输入有关软件包配置的信息：

Note

您的来源的软件包配置必须具有相同的持续时间，具体取决于来源的清单。而且，包配置中的所有源都必须具有相同数量的子流。为了满足这些要求，我们建议您为资产使用编码模板。我们建议您使用最小段长度为一秒的编码模板。 MediaTailor 不支持按标题或自动自适应比特率直播 (ABR)，因为这些编码方法违反了这些要求。

- 来源组：输入描述此包配置的源组名称，例如 HLS-4k。记下这个名字；当你创建频道的输出时，你会引用这个名字。有关更多信息，请参阅 将来源组与频道的输出配合使用。
- 类型：为此配置选择打包格式。 MediaTailor 支持 HLS 和 DASH。
- 相对路径：从源位置的基本 HTTP 网址到清单的相对路径。例如，/my/path/index. m3u8。

> **Note**
>
> MediaTailor 自动导入父多变体播放列表中包含的所有隐藏式字幕和子流。您无需为每个源代码格式副本 (DASH) 或变体流 (HLS) 创建单独的包配置。

有关软件包配置的更多信息，请参阅使用软件包配置。

7. 选择 “添加 VOD 来源”。

   如果要添加更多 VOD 源，请重复该过程中的步骤 4-7。

使用实时信号源

直播源代表您添加到来源位置的单个直播，例如直播足球比赛或新闻广播。创建频道后，您可以将一个或多个直播源添加到您的源位置，然后将每个直播源与一个节目相关联。

MediaTailor 支持以下类型的线性通道组装：

- 包含内容 VOD-to-live 的频道的 VOD 来源
- 包含直播内容与内容混合在一起的频道的直播源 VOD-to-live

VOD-to-live 内容的一个例子是将VOD素材库汇编成直播的频道。直播内容与内容混合的一个例子是，该频道主要播放 VOD VOD-to-live 内容，但夜间新闻活动或预先安排的体育赛事直播除外。直播内容与 VOD-to-live 内容混合的另一个例子是全直播到直播的频道，其来源因一天中的时间而异。

你可以使用直播资源来建立区域频道，该频道主要播放全国节目，但也包括区域节目重播，并混入 VOD 内容。为此，您需要为全国性内容运行一 encoder/packager 对编码器，然后在这些区域上线时运行区域编码器。然后，您可以创建区域频道组装频道，每个频道都有自己的时间表。这样，观众就可以根据需要来回切换。此设置可帮助您最大限度地 encoding/packaging 降低成本。

每个直播源必须至少有一个软件包配置。软件包配置为您的直播源指定包格式、清单位置和来源组。创建频道时，您可以使用包配置的源组在频道上创建相应的输出。例如，如果你的源代码以两种不同的格式（HLS 和 Dash）打包，那么你需要创建两个包配置，一个用于 DASH，一个用于 HLS。然后，您将创建两个通道输出，每个通道输出对应一个包配置。每个频道的输出都提供了一个用于播放请求的端点。在此示例中，该频道为 HLS 播放请求提供端点，为 DASH 播放请求提供终端节点。

使用直播源的一般要求

使用直播源时，您的内容必须符合以下一般要求：

- HLS Live sources-您必须为清单窗口中的第一个片段以及每个不连续部分提供`#EXT-X-PROGRAM-DATE-TIME`标签。
- HLS-您必须将广告标记配置为。`DATERANGE`
- 来源清单窗口-我们建议使用持续时间至少与 Channel Assembly MediaTailor 频道上的清单窗口一样长的清单窗口。作为最佳实践，可以考虑使用比 Channel Assembly 频道上的清单窗口时长 30 秒或更长的清单时长。
- 使目标持续时间与现有源的持续时间相匹配。
- 使子播放列表的数量与现有来源的数量相匹配。

配置

如果您在直播源工作流程中使用其他 AWS Elemental 媒体服务，我们建议您在设置 MediaPackage配置时遵循最佳实践。下表描述了如何根据您使用的直播标准配置 MediaPackage 设置。

MediaPackage 为直播源进行设置

标准	设置	值	必要性	注意
HLS	端点类型	`Apple HLS`	除非使用 CMAF，否则为必填项	为了匹配 HLS 职位 `ts` AWS Elemental MediaConvert
HLS	端点类型	`CMAF`	除非使用苹果 HLS，否则为必填项	为了匹配 HLS 职位 `mp4` AWS Elemental MediaConvert

标准	设置	值	必要性	注意
HLS	ProgramDateTimeIntervalSeconds	1	必填	您必须在每个片段#EXT-X-PROGRAM-DATE-TIME 上进行指定，以防止出现不连续播放时出现播放问题。
HLS	PlaylistWindowSeconds	比频道集合清单窗口长 30 秒	必填	
HLS	AdMarkers	DATERANGE	通过广告标记时为必填项	
HLS	IncludeIframeOnlyStream	已禁用	推荐	
DASH	ManifestLayout	FULL	推荐	
DASH	SegmentTemplateFormat	NUMBER_WITH_TIMELINE 或 TIME_WITH_TIMELINE	推荐	不支持 NUMBER_WITH_DURATION 。
DASH	ManifestWindowSeconds	比频道集合清单窗口长 30 秒	必填	
DASH	PeriodTriggers	ADS	通过广告标记时为必填项	

将直播源添加到您的来源位置

以下过程说明了如何使用 MediaTailor 控制台将直播源添加到您的源位置并设置软件包配置。有关如何使用 MediaTailor API 添加直播源的信息，请参阅 AWS Elemental MediaTailor API 参考CreateLiveSource中的。

> **Important**
>
> 在添加直播源之前，请确保在包配置中，每个源都有相同数量的子流。

将直播源添加到您的源位置

1. 打开 MediaTailor 控制台，网址为https://console.aws.amazon.com/mediatailor/。
2. 在导航窗格中，选择频道集合 > 来源位置。
3. 在来源位置窗格中，选择您在创建源位置过程中创建的来源位置。
4. 在 “直播源” 选项卡上，选择 “添加直播源”。
5. 在直播源详细信息下，输入直播源的名称：
   - 名称：直播源的标识符，例如 my-example- video。
6. 在 P ackage 配置 > 下，*source-group-name*输入有关软件包配置的信息：

   > **Note**
   >
   > 在套餐配置中，所有 VOD 源和直播源必须具有相同数量的子流。我们建议您以相同的方式配置源流。

   - 来源组：输入描述此包配置的源组名称，例如 HLS-4k。记下这个名字；当你创建频道的输出时，你会引用这个名字。有关更多信息，请参阅 将来源组与频道的输出配合使用。
   - 类型：为此配置选择打包格式。 MediaTailor 支持 HLS 和 DASH。
   - 相对路径：从源位置的基本 HTTP 网址到清单的相对路径。例如，/my/path/index. m3u8。

     > **Note**
     >
     > MediaTailor 自动导入父多变体播放列表中包含的所有隐藏式字幕和子流。您无需为每个源代码格式副本 (DASH) 或变体流 (HLS) 创建单独的包配置。

有关软件包配置的更多信息，请参阅使用软件包配置。

7. 选择 “添加直播源”。

如果要添加更多直播源，请重复该过程中的步骤 4-6。

使用软件包配置

包配置是源代码的表示形式，它包含在不同设备上播放所需的各种封装特性。例如，您的源可能有三种打包格式：带有 DRM 的 HLS、带区段时间轴寻址的 DASH 和带有 CMAF 片段的 HLS。

频道集合不会重新打包你的来源。如果要为给定来源包含多种打包格式，则必须使每种打包格式在源位置可用，并指定每种打包格式的路径。

每个软件包配置对象必须包含以下内容：

- 相对路径-源文件打包格式相对于源位置的完整路径。例如，/my/path/index. m3u8。
- 来源组-用于将包配置与频道输出关联的源组的名称。
- 类型-HLS 或 DASH。

创建频道后，还必须声明要用于频道输出的每个源组。

清单缓存

MediaTailor 定期并有机会地缓存源播放列表，以提高频道组合的性能和可靠性。有时，与源位置的原始版本相比，缓存的版本会过时。 MediaTailor 要强制刷新源代码的缓存版本，请调用UpdateVodSource。例如，当源代码中的嵌入路径发生变化时，使用此调用。即使您看到的请求很少，也要确保您的来源位置始终保持最新版本的可用来源 MediaTailor。

使用通道

频道将您的源清单汇编成线性流。每个通道包含一个或多个与您的封装配置相对应的输出。

首先创建频道，然后通过创建节目将 VOD 源和直播源添加到频道的日程安排中。每个节目都与 VOD 源或直播源相关联。

主题

- 使用 MediaTailor 控制台创建频道

- 将来源组与频道的输出配合使用
- 使用 MediaTailor 控制台删除频道

使用 MediaTailor 控制台创建频道

以下过程介绍如何使用 MediaTailor 控制台创建频道。

创建通道

1. 打开 MediaTailor 控制台，网址为https://console.aws.amazon.com/mediatailor/。
2. 在导航窗格中，选择频道集合 > 频道。
3. 在导航栏上，选择创建频道。
4. 在 “频道详情” 下，输入有关您的频道的详细信息：
   - 名称：输入频道的名称。
   - 等级：等级决定频道支持哪些功能以及运营该频道的成本。有关定价的更多信息，请参阅 Channel Assembly 定价页面。 MediaTailor 支持以下等级：
     - 基本-基本层支持线性和循环播放模式，不支持直播源。
     - 标准-标准层支持直播源，需要线性播放模式。

       当您在频道详细信息中选择标准时，可以在受众详细信息下定义受众。当你要为默认节目创建 AudienceMedia 时，这些受众将用于 Progrules。
       - 选择添加。
       - 在文本框中输入受众名称。它的长度必须介于 1 到 32 个字母数字字符之间。
       - 选择确认。
       - 选择下一步。
   - 播放模式：播放模式设置频道的播放行为。 MediaTailor 支持以下播放模式：
     - Loop-日程安排中的节目无休止 back-to-back地循环播放。按计划播放最后一个节目后，回放会循环播放到第一个节目。播放会继续循环，直到您停止频道为止。
     - 线性-时间表中的每个节目播放一次， back-to-back。
5. 在 Filler slat e 中，选择引用 slate 位置的来源位置名称和用作板块的 VOD 来源名称。 MediaTailor 使用名单来填补时间表中节目之间的空白。如果 slate 的持续时间小于程序之间间隔的持续时间，则 MediaTailor 循环使用 slate。如果您的频道使用线性播放模式，则必须将 filler slate 字段配置为。 MediaTailor 不支持循环播放模式的填充板。

6. 选择下一步。
7. 根据节目规则指定受众详细信息。
8. 当您在频道详细信息中选择标准时，可以在受众详细信息下定义受众。当你要为默认节目创建 AudienceMed ia 时，这些受众将用于 Progrule s：

   - 选择添加，然后在文本框中添加受众，然后选择确认。

   > **Note**
   >
   > 输入长度不超过 _ 32 个字母数字字符的名称。

   - 输出类型：选择频道的直播格式。支持 DASH 和 HLS。
   - 来源组：输入您在软件包配置中创建的源组的名称，如中所述将 VOD 源添加到您的来源位置。
9. 选择下一步。
10. 在清单设置下，输入有关清单设置的其他信息：

    - 清单窗口（秒）：每个清单中包含的时间窗口，以秒为单位。最小值为 30 秒，最大值为 3600 秒。
    - 广告标记类型（仅限 HLS 输出）：VOD 节目广告插播中显示的广告标签类型。选择`Daterange`在带有`EXT-X-DATERANGE`标签的 VOD 节目中 MediaTailor 插入广告片段。选择`Scte35 Enhanced`使用和`EXT-X-CUE-IN`标签在 VOD 节目中 MediaTailor 插入广告片`EXT-X-CUE-OUT`段。有关这些标签类型的更多信息，请参阅SCTE-35 广告中断消息。对于直播工作流程，无论所选的广告标记类型如何，都要 MediaTailor 始终传递标签，并且不会通过任何增强型 Scte35 标签。DATERANGE
11. 如果要配置多通道输出，请在输出下选择添加。然后，通过完成此过程中的步骤 6 和步骤 7 来配置输出的详细信息。
12. 选择下一步。
13. 在渠道政策下，选择您频道的 IAM 策略设置：

    - 请勿附加频道政策：仅允许有权访问此账户凭证的用户进行播放。
    - 附加自定义策略：定义自己的策略，将访问权限限制为任意数量或任意数量。
    - 附加公共政策：接受所有传入的客户端请求到频道的输出。如果您想使用 MediaTailor 广告插播，则必须使用此选项。
14. 选择下一步。
15. 在“查看并创建”窗格上查看您的设置。

16. 选择 Create channel (创建通道)。

> **Note**
>
> 频道是在停止状态下创建的。在您使用 MediaTailor 控制台或 MediaTailor StartChannel API 启动频道之前，您的频道才会处于活动状态。

将来源组与频道的输出配合使用

源组将包配置与频道上的输出相关联。在源上创建包配置时，可以识别源组的名称。然后，当您在频道上创建输出时，输入相同的名称以将输出与包配置相关联。添加到频道上节目的 VOD 源和直播源必须属于输出中标识的来源组。

例如：

- VOD 来源 1 和 2 都有三种包含源组的软件包配置：HLS、DASH 和 HLS- 4K。
- VOD 来源 3 有两种包配置，分别是源组 HLS 和 DASH。

如果频道 A 有两个输出以及源组 HLS 和 DASH，则频道输出可以使用所有三个 VOD 源。这是因为 VOD 来源 1、2 和 3 都具有带有源组标签 HLS 和 DASH 的软件包配置。

如果频道 B 有两个输出，其源组为 HLS 和 HL S-4K，则它可以使用 VOD 源 1 和 2，但不能使用 3。这是因为 VOD 源 1 和 2 都具有带有源组标签 HLS 和 HLS- 4K 的软件包配置。

如果频道 C 在源组 DASH 中只有一个输出，则它可以使用所有三个 VOD 源。所有三个 VOD 源都有 DASH 源组的软件包配置。

使用 MediaTailor 控制台删除频道

要删除您的频道，请完成以下步骤。

删除您的频道

1. 打开 MediaTailor 控制台，网址为https://console.aws.amazon.com/mediatailor/。
2. 在导航窗格中，选择频道集合 > 频道。
3. 选择要删除的频道。

4. 如果您的频道正在播出，请从 “操作” 下拉菜单中选择 “停止”。您必须先停止您的频道，然后才能将其删除。
5. 频道停止后，从 “操作” 下拉菜单中选择 “删除”。

将节目添加到频道的日程安排中

每个节目都包含一个 VOD 源或作为您账户中来源位置一部分的直播源。您可以将节目添加到频道的日程安排中，以控制它们在频道直播中的播放顺序。

包含 VOD 来源的节目可以配置一个或多个广告插播时间。每个广告时段都包含一个名单，即来自某个来源位置的 VOD 来源。要制作广告插播时间，您需要将偏移量（以毫秒为单位）添加到程序中。

主题

- 使用 MediaTailor 控制台在频道时间表内创建节目
- 使用节目规则定义受众群组和替代内容
- 生成针对特定受众的清单

使用 MediaTailor 控制台在频道时间表内创建节目

以下过程介绍如何使用 MediaTailor 控制台在频道的节目表中创建节目。它还描述了如何配置广告插播时间（可选）。有关如何使用 MediaTailor API 创建程序的信息，请参阅 AWS Elemental MediaTailor API 参考CreateProgram中的。

添加程序

1. 打开 MediaTailor 控制台，网址为https://console.aws.amazon.com/mediatailor/。
2. 在导航窗格中，选择频道集合 > 频道。
3. 在频道窗格中，选择您在创建通道过程中创建的频道。
4. 在计划详细信息中，输入有关您的计划的详细信息：
   - 名称：这是您添加到频道的节目的名称。
   - 来源类型：确定程序播放的源视频类型。此选项仅适用于标准频道。
     - VOD-该节目播放 video-on-demand来源，例如预先录制的电视剧集。
     - 直播-该节目播放实时来源，例如新闻直播。
   - 源位置名称：与程序 MediaTailor 关联的源位置。

- 如果选择 “选择现有来源位置”，请从 “选择来源位置” 菜单中选择源位置名称。或者，按名称搜索您的来源位置。如果您有大量的来源位置，这会很有用。
- 如果选择 “输入来源地点名称”，请按名称搜索来源地点。
- VOD 来源名称：与该节目 MediaTailor 关联的 VOD 源的名称：
  - 如果您选择选择现有 VOD 来源，请从与您的账户关联的 VOD 来源列表中选择一个 VOD 来源名称。或者，按名称搜索您的 VOD 来源。如果您有大量的 VOD 来源，这会很有用。
  - 如果您选择 “按名称搜索”，请按名称搜索直播源。
- 直播源名称：要与节目关联的直播源的名称。只有选择 “直播” 作为源类型时，此选项才可用。
  - 如果选择 “选择现有来源位置”，请从 “选择来源位置” 菜单中选择源位置名称。或者，按名称搜索您的来源位置。如果您有大量的来源位置，这会很有用。
  - 如果选择 “输入来源地点名称”，请按名称搜索来源地点。
  - VOD 来源名称：与该节目 MediaTailor 关联的 VOD 源的名称：
    - 如果您选择 “选择现有直播源”，请从与您的账户关联的直播源列表中选择直播源名称。或者，您也可以按名称搜索直播源。如果你有大量的直播源，这会很有用。
    - 如果您选择 “按名称搜索”，请按名称搜索直播源。

5. 选择 “下一步” 转至 “计划配置” 选项卡。
6. 在 “播放配置” 下，定义节目在频道的日程安排中何时播放：

- 持续时间（以毫秒为单位）：定义程序的持续时间（以毫秒为单位）。此选项仅适用于使用实时源的节目。
- 过渡类型：定义调度中从一个程序到另一个程序的过渡：
  - 相对：该节目在日程安排中的另一个节目之前或之后播放。此选项仅适用于使用 VOD 源的节目。
  - 绝对：程序在特定的挂钟时间播放。 MediaTailor 尽最大努力在您指定的时钟时间播放该节目。 MediaTailor 在前一个节目或板块之间的公共片段边界上开始播放该节目。此选项仅适用于配置为使用线性的通道播放模式：播放模式设置频道的播放行为。 MediaTailor 支持以下播放模式：。
  - 节目开始时间：对于绝对过渡类型，指节目预定播放时的挂钟时间。如果要将此程序添加到正在运行的线性通道中，则必须输入自当前时间起 15 分钟或更晚的开始时间。
  - 相对位置：选择将程序插入时间表中相对于另一个程序的位置。您可以选择 “程序之前” 或 “程序之后”。如果这是您频道节目表中的第一个节目，则此设置不适用。

- 如果您选择 “选择现有节目”，请从 “使用现有节目” 菜单中从该频道接下来播放的 100 个节目的预定义列表中选择节目名称。
- 如果您选择 “按名称搜索节目”，请输入频道中现有节目的名称。

如果您想在节目中添加广告插播时间，请继续下一步。广告时段只能为使用 VOD 来源的节目进行配置。对于直播来源，DASH 清单中的广告插播和使用该EXT-X-DATERANGE标签的 HLS 清单中的广告插播时间会自动传递。

7. 选择 “下一步” 转至 “添加广告插播时间”。
8. 选择 “添加广告插播时间”。在 “广告插播时间” 下，配置广告插播时间点的设置：
   - Slate 来源位置名称：选择 “选择现有源位置”，然后选择您之前在本任务中创建的 Slate 存储来源位置。
   - VOD 来源名称：选择 “选择现有 VOD 来源”，然后选择您在本任务之前添加的 Slate 所用的 VOD 来源。广告的持续时间决定了广告插播的持续时间。
   - 偏移量（以毫秒为单位）：此值确定广告插播开始时间（以毫秒为单位），即相对于节目开头的偏移量。输入任何小于 VOD 源持续时间且与节目 VOD 源内所有轨道（所有音频、视频和隐藏字幕轨道）上的片段边界对齐的值，否则广告插播将被跳过。例如，如果您输入 0，则会创建一个在节目开始之前播放的前贴片广告插播时间。
   - 可用编号：按照 SCTE-35 规范 9.7.3.1 节 “数字节目插入提示消息” 中的定义， MediaTailor 写入此值。`splice_insert.avail_num`默认值是 0。值必须介于 0 和 256 之间（含）。
   - 预期可用：按照 SCTE-35 规范第 9.7.3.1 节的定义`splice_insert.avails_expected`，将此值 MediaTailor 写入。默认值是 0。值必须介于 0 和 256 之间（含）。
   - 拼接事件 ID：按照 SCTE-35 规范第 9.7.3.1 节中的定义`splice_insert.splice_event_id`，将此值 MediaTailor 写入。默认值是 1。
   - 唯一程序 ID：按照 SCTE-35 规范第 9.7.3.1 节的定义`splice_insert.unique_program_id`，将此值 MediaTailor 写入。默认值是 0。值必须介于 0 和 256 之间（含）。
9. 对于标准线性通道，请选择 “下一步” 转至 “设置备用媒体”。

   有关使用创建备用媒体 MediaTailor 的更多信息，请参阅创建备用媒体。

   有关使用对广告插播时间 MediaTailor 进行个性化设置的更多高级信息，请参阅在频道直播中插入个性化广告和广告插播时间。
10. 选择 “下一步” 转至 “查看并创建”。

11. 选择 “添加程序”。

    有关使用对广告插播时间 MediaTailor 进行个性化设置的更多高级信息，请参阅[在频道直播中插入个性化广告和广告插播时间](#)。

12.

> **Important**
>
> 对于循环频道，如果您修改了计划在接下来的 10 分钟内的节目的节目列表，则编辑要等到下一个循环才会显现出来。

    在 “计划详情” 下，输入有关您的计划的详细信息：

    - 名称：这是您添加到频道的节目的名称。
    - 来源类型：确定节目播放的源类型。此选项仅适用于标准频道。
      - VOD-该节目播放 VOD 来源，例如预先录制的电视剧集。
      - 直播-该节目播放实时来源，例如新闻直播。
    - 源位置名称：要与程序关联的源位置。

      如果选择 “选择现有来源位置”，请从 “选择来源位置” 下拉菜单中选择来源位置名称。或者，您也可以按名称搜索来源位置。如果您有大量的来源位置，这会很有用。

      如果选择 “输入来源地点名称”，请按名称搜索来源地点。
    - VOD 来源名称：要与节目关联的 VOD 源的名称。

      如果您选择选择现有 VOD 来源，请从与您的账户关联的 VOD 来源列表中选择一个 VOD 来源名称。您也可以按名称搜索您的 VOD 来源。如果您有大量的 VOD 来源，这会很有用。

      如果您选择 “按名称搜索”，请按名称搜索您的 VOD 来源。
    - 直播源名称：要与节目关联的直播源的名称。只有选择 “直播” 作为源类型时，此选项才可用。

      如果您选择 “选择现有直播源”，请从与您的账户关联的直播源列表中选择直播源名称。或者，您也可以按名称搜索直播源。如果你有大量的直播源，这会很有用。

      如果您选择 “按名称搜索”，请按名称搜索直播源。

13. 在 “播放配置” 下，定义节目在频道的日程安排中何时播放：

    - 持续时间（以毫秒为单位）：定义程序的持续时间（以毫秒为单位）。此选项仅适用于使用实时源的节目。

- 过渡类型：定义调度中从一个程序到另一个节目的过渡。
  - 相对节目-节目在日程安排中的另一个节目之前或之后播放。此选项仅适用于使用 VOD 源的节目。
  - 绝对-节目在特定的挂钟时间播放。 MediaTailor 尽最大努力在您指定的时钟时间播放该节目。我们开始在前一个节目或板块之间的公共片段边界上播放该节目。此选项仅适用于配置为使用linear playback mode。

    > **Note**
    >
    > 请注意绝对过渡类型的以下行为：
    >
    > - 如果调度中的前一个节目的持续时间超过挂钟时间，则在最接近挂 MediaTailor 钟时间的公共段边界上截断前一个节目。
    > - 如果日程安排中的节目之间存在间隔，则 MediaTailor 播放filler slate。如果石板的持续时间小于间隙的持续时间，则 MediaTailor 循环石板。

- 节目开始时间-对于绝对过渡类型，指节目预定播放时的挂钟时间。如果要将此程序添加到正在运行的线性通道中，则必须输入自当前时间起 15 分钟或更晚的开始时间。
- 相对位置：选择将程序插入计划相对于另一个程序的位置。您可以选择 “程序之前” 或 “程序之后”。如果这是您频道节目表中的第一个节目，则此设置不适用。
- 相对程序：用于在之前或之后插入新程序的程序的名称。如果这是您频道节目表中的第一个节目，则此设置不适用。

  如果您选择 “选择现有节目”，请在 “使用现有节目” 下拉菜单中从该频道接下来播放的 100 个节目的预定义列表中选择节目名称。

  如果您选择 “按名称搜索节目”，请输入频道中现有节目的名称。

如果您想在节目中添加广告插播时间，请继续下一步。广告时段只能为使用 VOD 来源的节目进行配置。对于直播来源，DASH 清单中的广告插播和使用该EXT-X-DATERANGE标签的 HLS 清单中的广告插播时间会自动传递。

14. 选择 “添加广告插播时间”。在 “广告插播时间” 下，配置广告插播时间点的设置：

- Slate 源位置名称：选择 “选择现有源位置”，然后选择您在本教程前面创建的 Slate 存储来源位置。
- VOD 来源名称：选择 “选择现有 VOD 来源”，然后选择在本教程前面部分添加的 VOD 来源。广告的持续时间决定了广告插播的持续时间。

- 对于 O ffset（以毫秒为单位）：此值确定广告插播开始时间（以毫秒为单位），即相对于节目开头的偏移量。输入任何小于 VOD 源持续时间且与节目 VOD 源内所有轨道（所有音频、视频和隐藏字幕轨道）上的片段边界对齐的值，否则广告插播将被跳过。例如，如果您输入 0，则会创建一个在节目开始之前播放的前贴片广告插播时间。

  > **Note**
  >
  > 如果 MediaTailor 检测到 VOD 来源中持续时间`EXT-X-CUE-OUT`为零的广告标记（例如 HLS 和 `EventStream` DASH 的广告标记），则可以从下拉菜单中选择这些广告标记的偏移量作为广告插播时间的偏移量。`DATERANGE`为了检测到广告机会，VOD 来源内所有套餐配置中的广告机会必须以相同的偏移量出现，并且其持续时间必须为零。

- 消息类型：SCTE-35 广告插入类型。选择 SPLICE_INSERT 或 TIME_SIGNA L：
  - SPLICE_INSERT：使用拼接插入参数提供有关广告插播时间的基本元数据。
  - TIME_SIGN AL：使用分段描述符提供更高级的元数据。有关消息类型之间差异的更多信息，请参阅the section called “SCTE-35 广告中断消息”。

  对于 SPLICE_INSERT 消息，请键入：

  - 对于 Avail 编号`splice_insert.avail_num`，按照 SCTE-35 规范第 9.7.3.1 节中的定义，将其写入。默认值为 `0`。值必须介于`0`和之间`256`，含在内。
  - 对于预期可用性，按照 SCTE-35 规范`splice_insert.avails_expected`第 9.7.3.1 节的定义，将其写入。默认值为 `0`。值必须介于`0`和之间`256`，含在内。
  - 对于 Splice 事件 ID，按照 SCTE-35 规范第 9.7.3.1 节中的定义`splice_insert.splice_event_id`，将其写入。默认值为 1。
  - 对于唯一程序 ID，按照 SCTE-35 规范第 9.7.3.1 节中的定义，将其写入。`splice_insert.unique_program_id`默认值为 `0`。值必须介于`0`和之间`256`，含在内。

  对于 TIME_SIGNAL 消息类型：

  - 对于分段事件 ID，按照 SCTE-35 规范第 10.3.3.1 节中的定义，将其写入。`segmentation_descriptor.segmentation_event_id`默认值为 1。
  - 对于分段类型 ID，按照 SCTE-35 规范第 10.3.3.1 节中的定义，将其写入。`segmentation_descriptor.segmentation_type_id`默认值为 48 (0x30)。值必须介于`0`和之间`256`，含在内。

- 对于分段 UPID，按照 SCTE-35 规范第 10.3.3.1 节中的定义，将其写入。segmentation_descriptor.segmentation_upid该值必须是包含字符和的十六进制字符串。0-9 A-F默认值是空字符串。
- 对于分段 UPID 类型，按照 SCTE-35 规范第 10.3.3.1 节中的定义segmentation_descriptor.segmentation_upid_type，将其写入。默认值为 14 (0x0E)。值必须介于0和之间256，含在内。
- 对于分段编号，按照 SCTE-35 规范第 10.3.3.1 节中的定义，将其写入。segmentation_descriptor.segment_num默认值为 0。值必须介于0和之间256，含在内。
- 对于预期的区段，按照 SCTE-35 规范segmentation_descriptor.segments_expected第 10.3.3.1 节中的定义，将其写入。默认值为 0。值必须介于0和之间256，含在内。

15. 选择 “添加程序”。

有关使用对广告插播时间 MediaTailor 进行个性化设置的更多高级信息，请参阅在频道直播中插入个性化广告和广告插播时间。

> **Note**
>
> 如果您的频道至少有一个Enhanced Scte35广告标记类型的输出，则可以提交广告插播元数据。 MediaTailor 将提交的键值对写入广告插播EXT-X-ASSET时间段的标签。

使用节目规则定义受众群组和替代内容

使用节目规则，您可以为频道定义受众群组，并指定要为这些受众播放的替代媒体。您可以将一个或多个备用内容源与节目的受众相关联。节目结束后，除非您指定其他替代媒体，否则将播放默认的受众内容。

节目规则适用于采用线性播放模式的标准级频道。 MediaTailor频道支持所有 VOD 来源和直播源的替代媒体。

有关使用示例，请参阅将程序规则与一起使用 AWS MediaTailor。

定义受众

在配置频道时，通过逐个键入受众来定义 MediaTailor频道上的受众。您可以通过 MediaTailor 控制台或 MediaTailor `CreateChannel` API 执行此操作。每个受众的长度必须介于 1 到 32 个字母数字字符之间。如果为受众提供的值无效，则请求将失败。

您只能使用线性播放模式在标准级别的频道上定义受众。

当你需要更新受众时，你可以使用 MediaTailor 控制台或 MediaTailor `UpdateChannel` API 来完成此操作。

如果您正在使用该`ProgramRules`功能，请确保在`CreateProgram`或`UpdateProgram`请求中`AudienceMedia`定义的内容包含频道中定义的现有受众。

创建备用媒体

以下任务说明如何使用 MediaTailor 控制台定义备用媒体。有关如何使用 MediaTailor API 定义替代媒体的信息，请参阅 AWS Elemental MediaTailor API 参考*`CreateProgram`*中的。

要在新节目中定义备用媒体，请执行以下操作：

1. 打开 MediaTailor 控制台，网址为https://console.aws.amazon.com/mediatailor/。
2. 在导航窗格中，选择频道集合 > 频道。
3. 选择要向其添加备用媒体的频道名称。
4. 创建程序。有关更多信息，请参阅 使用 MediaTailor 控制台在频道时间表内创建节目。
5. 配置备用媒体：
   - 在 “受众” 框中选择 “添加”，选择要为其定义替代媒体的受众。
   - 从 “受众” 菜单中选择在频道上定义的受众。
   - 选择 “添加备用媒体”，开始为节目定义备用媒体。
   - MediaTailor 创建一个 “备用媒体 1” 盒子。这是第一个在节目 MediaTailor 中作为备用媒体播放的内容。
   - 在 “备用媒体 1” 框中：
     - 选择来源位置。
     - 选择 VOD 或 Live Source 类型：

       适用于 VOD
       - 为来源类型选择 VOD。

- （可选）指定剪辑范围。使用 VOD 源（包括备用媒体 VOD 源），您可以指定要播放 VOD 源的一部分，从来源的开头开始 and/or 剪辑。指定起始和结束偏移以毫秒为单位。
- （可选）a\ 添加广告插播时间。这与创建程序时的方式相同。有关更多信息，请参阅 使用 MediaTailor 控制台在频道时间表内创建节目。

为了直播

- 为来源类型选择 “直播”。
- 选择直播源。
- 输入开始时间，以此直播源应启动的挂钟时间为单位的毫秒。直播源只能在定义它的默认程序的时间范围内播放。如果启动时间早于默认程序的启动，则要等到默认程序启动后才会开始。如果开始时间是在默认程序结束之后，则 MediaTailor 不会播放直播源。
- 输入持续时间（以毫秒为单位）。持续时间必须至少为 10 分钟。
- 通过再次选择 “添加备用媒体”，可以将其他备用媒体添加到该节目中，供观众使用。这将创建另一个标有 “备用媒体 2” 的盒子。每个节目、每个受众最多可以指定 5 个替代媒体来源。
- 为所有所需受众定义替代媒体后，选择 “下一步” 并继续创建节目。

  有关更多信息，请参阅 使用 MediaTailor 控制台在频道时间表内创建节目。

> **Note**
>
> 备用媒体只能在定义的节目时间范围内播放。如果所有备用内容都超出了默认内容，则 MediaTailor 会将其截断。 MediaTailor 按定义的顺序播放备用媒体。直播备用媒体的开始时间将始终优先，并将截断之前安排的 VOD 源或直播源。任何没有为观众准备的替代媒体时，都将填满频道定义的填充名单

- 要为其他受众定义受众媒体，请再次选择受众旁边的添加。如上所述，选择新创建的受众，设置受众 ID 并添加备用媒体。任何一个节目最多可以有 5 位观众使用备用媒体。

生成针对特定受众的清单

要检索特定受众的清单，请使用`aws.mediatailor.channel.audienceId`查询参数。此查询参数可以由您的 CDN 动态追加，也可以通过调用您的内容或客户管理系统来添加。您必须保持给定播放会话与`audienceId`外部的关联 MediaTailor。这将检索特定受众的清单，其中包含为该受众定义的任何替代媒体来代替默认内容。重要的是，在为特定受众请求清单后，玩家必须始终使用相同的受众 ID 请求清单，否则可能会出现播放错误。

如果向频道中不存在的观众发出了请求，则会 MediaTailor 返回 404 错误。

Example为受众获取清单

```
https:// prefix>.channel-assembly.mediatailor.us-
west-2.amazonaws.com/v1/channel/ExampleChannel/index_dash.mpd?
aws.mediatailor.channel.audienceId=Seattle
```

在频道直播中插入个性化广告和广告插播时间

使用 MediaTailor，您可以通过在节目中插入广告插播来通过频道组装线性流获利，而无需使用 SCTE-35 标记来调整内容。您可以将频道组合与 MediaTailor 广告插入服务或任何服务器端广告插入 (SSAI) 配合使用。

以下主题介绍如何在频道的线性直播中插入个性化广告和广告插播时间。

主题

- 使用设置广告插入 MediaTailor
- SCTE-35 广告中断消息

使用设置广告插入 MediaTailor

要将个性化广告插入频道的直播中，您的频道的端点网址就是其内容来源 AWS Elemental MediaTailor。本指南介绍如何设置广告 MediaTailor 插播。

先决条件

在您开始之前，确保您满足以下要求：

- 为 HLS 和 DASH 直播做好 MediaTailor 广告投放准备。
  - 如果您尚未准备好内容流，请参阅 MediaTailor 广告插入入门主题第 2 步：准备直播中的。
- 拥有广告决策服务器 (ADS)。
- 在程序中配置广告插播设置。有关更多信息，请参阅Configuring ad breaks for your program步骤。

作为最佳实践，可以考虑在渠道组装和 MediaTailor 广告插入之间使用内容分发网络 (CDN)。MediaTailor 广告插入服务可以生成其他来源请求。因此，最佳做法是将 CDN 配置为代理来自频道集合的清单，然后在内容源 URL 上使用 CDN 前缀的 URL。

MediaTailor 为广告插入进行配置

以下内容显示了如何配置 MediaTailor 主机设置，以便您可以将个性化广告插入频道的直播中。

为广告插入 MediaTailor 进行配置

1. 打开 MediaTailor 控制台，网址为https://console.aws.amazon.com/mediatailor/。
2. 在导航窗格中，选择配置。
3. 在 “必填设置” 下，输入有关您的配置的基本必填信息：
   - 名称：您的配置名称。
   - 内容来源：输入频道输出中的播放网址，减去文件名和扩展名。有关 MediaTailor配置的高级信息，请参阅必需的设置。
   - 广告决策服务器：输入广告的网址。
4. 您可以选择配置别名、个性化详细信息和高级设置。有关这些设置的信息，请参阅可选配置设置。
5. 在导航栏上，选择创建配置。

既然你已经设置了广告 MediaTailor 插播，你还可以设置广告插播时间。有关详细说明，请参阅MediaTailor 广告插入入门。

SCTE-35 广告中断消息

使用 MediaTailor，您可以根据来源位置和 VOD 源资源创建内容频道。然后，您可以根据频道的时间表为每个节目设置一个或多个广告时段。您可以使用基于 SCTE-35规范的消息来限制内容是否有广告插播时间。例如，您可以使用 SCTE-35消息来提供有关广告插播时间的元数据。有关 SCTE-35规范的更多信息，请参阅数字节目插入提示消息。

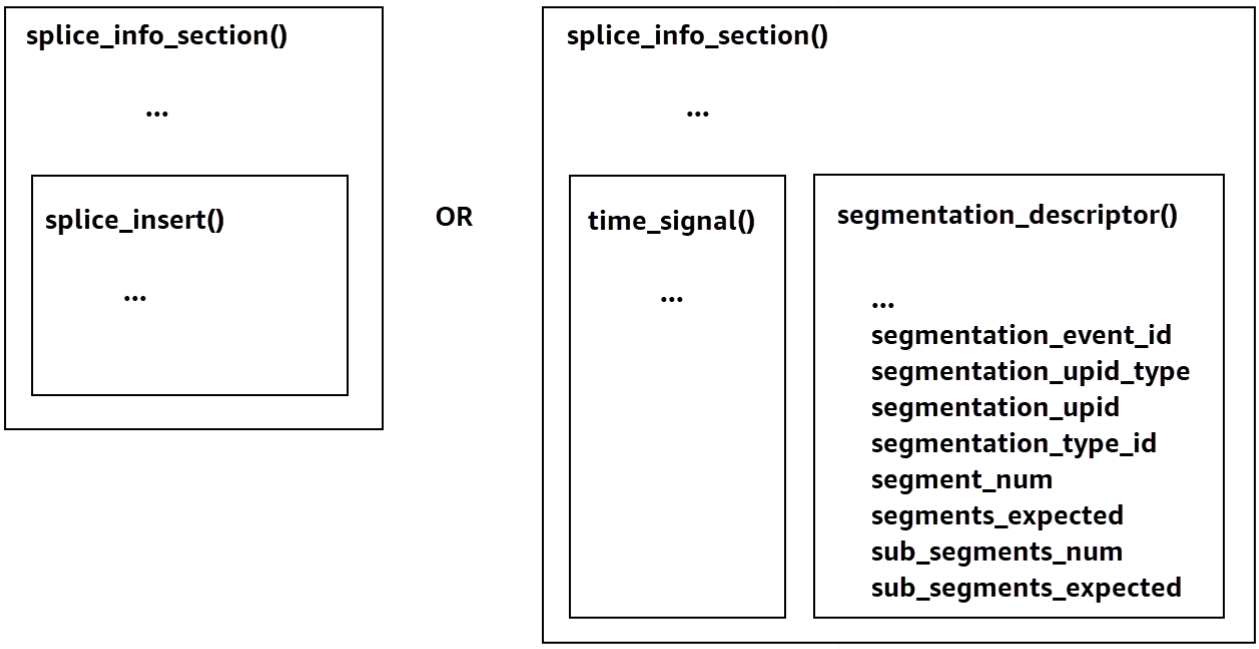
您可以通过以下两种方式之一来设置广告插播时间：

- 附上带有`time_signal` SCTE-35 消息的`segmentation_descriptor`消息。此`segmentation_descriptor`消息包含更高级的元数据字段，例如内容标识符，用于传达有关广告插播时间的更多信息。 MediaTailor 将广告元数据作为 `EXT-X-DATERANGE` (HLS) 或 `EventStream` (DASH) 广告标记 SCTE-35 数据的一部分写入输出清单。
- 附上一`splice_insert` SCTE-35 条消息，提供有关广告插播时间的基本元数据。
- HLS：
  - 当广告标记类型为时`Daterange`，在清单中将广告插播`EXT-X-DATERANGE`时间 MediaTailor 指定为标签。

- 当广告标记类型为时Scte35 Enhanced，使用以下标签 MediaTailor 指定广告插播时间：
  - MediaTailor EXT-X-CUE-OUT在广告板的第一部分放置一个，表示从内容到广告插播时间已被切断。它包含广告插播的预期持续时间，例如EXT-X-CUE-OUT:Duration=30。
  - >EXT-X-ASSET：此标签与创建或更新节目 AdBreak 时提供的广告插播元数据出现在同一片段上，EXT-X-CUE-OUT并且包含该广告插播元数据。它总是包含CAID。
  - EXT-OATCLS-SCTE35：此标签与消息出现在同一段中，EXT-X-CUE-OUT并且包含消息的 base64 编码字节。 SCTE-35
  - EXT-X-CUE-OUT-CONT：此标签显示在广告列表中的每个后续细分中，并包含持续时间和经过的时间信息。它还包含 base64 编码的 SCTE-35 消息，以及. CAID
  - EXT-X-CUE-IN：广告插播结束后，此标签会出现在第一段内容上，表示从广告插播到内容的剪辑。

下图显示了使用 SCTE-35 消息在频道中设置广告时段的两种方法：

- 使用splice_insert()消息设置带有基本元数据的广告时段。
- 将消息与time_signal()消息结合使用，设置带有更详细元数据的广告时段。segmentation_descriptor()



有关使用的信息`time_signal`，请参阅 2022 年 SCTE-35 规范第 9.7.4 节 “数字节目插入提示消息”。

广告插播信息显示在输出`splice_info_section` SCTE-35数据中。使用 MediaTailor，您可以将一条`segmentation_descriptor`消息与一条`time_signal`消息配对。

> **Note**
>
> 如果您发送`segmentation_descriptor`消息，则必须将其作为`time_signal`消息类型的一部分发送。该`time_signal`消息仅包含使用给定时间戳 MediaTailor 构造的`splice_time`字段。

下表描述了每`segmentation_descriptor`条消息 MediaTailor 所需的字段。有关更多信息，请参阅 2022 年 SCTE-35 规范第 10.3.3.1 节，您可以在 ANSI 网上商店网站上购买。

分段描述符消息的必填字#

字段	Type	默认 值	说明
segmentation_event_id	integer	1	这是写给的segmentation_descriptor.segmentation_event_id 。
segmentation_upid_type	integer	14 (0x0E)	这是写给的segmentation_descriptor.segmentation_upid_type 。该值必须介于 0 和 256 之间（含）。
segmentation_upid	字符串	""（空字符串）	这是写给的segmentation_descriptor.segmentation_upid 。该值必须是包含字符和的十六进制字符串。0-9 A-F
segmentation_type_id	integer	48 (0x30)	这是写给的segmentation_descriptor.segmentation_type_id 。该值必须介于 0 和 256 之间（含）。

字段	Type	默认 值	说明
segment_num	integer	0	这是写给的segmentation_descriptor.segment_num 。该值必须介于 0 和 256 之间（含）。
segments_expected	integer	0	这是写给的segmentation_descriptor.segments_expected 。该值必须介于 0 和 256 之间（含）。
sub_segment_num	integer	null	这是写给的segmentation_descriptor.sub_segment_num 。该值必须介于 0 和 256 之间（含）。
sub_segments_expected	integer	null	这是写给的segmentation_descriptor.sub_segments_expected 。该值必须介于 0 和 256 之间（含）。

下表显示了为segmentation_descriptor消息的某些字段 MediaTailor 自动设置的值。

为 s **egmentati MediaTailor on_d** escriptor 消息的字段设置的值

字段	Type	值
segmentation_event_cancel_indicator	布尔值	True
program_segmentation_flag	布尔值	True
delivery_not_restricted_flag	布尔值	True

MediaTailor 始终将设置segmentation_duration_flag为True。 MediaTailor用状态内容的持续时间（以刻度为单位）填充该segmentation_duration字段。

> **Note**
>
> MediaTailor 发送time_signal消息时，它将splice_info_section消息中的splice_command_type字段设置为 6 (0x06)。

在 HLS 输出中，对于AdBreak带有time_signal消息的，输出EXT-X-DATERANGE标签包括一个设置为消息序列化版本的SCTE-35splice_info_section字段。例如，以下EXT-X-DATERANGE标签显示splice_info_section消息的序列化版本：

```
#EXT-X-DATERANGE:ID=\"1\",START-DATE=\"2020-09-25T02:13:20Z\",DURATION=300.0,SCTE35-
OUT=0xFC002C0000000000000FFF0050680000000000016021443554549000000100E000019BFCC00E00300000000
```

在 DASH 输出中，对于AdBreak带有time_signal消息的，输出EventStream元素包括一个以scte35:TimeSignal和scte35:SpliceInfoSection元素为子scte35:SegmentationDescriptor元素的元素。该scte35:TimeSignal元素有一个子scte35:SpliceTime元素，而该scte35:SegmentationDescriptor元素有一个子scte35:SegmentationUpid元素。例如，以下 DASH 输出显示了EventStream元素的子元素结构：

```
<EventStream schemeIdUri="urn:scte:scte35:2013:xml" timescale="90000">
    <Event duration="27000000">
```

```
        <scte35:SpliceInfoSection protocolVersion="0" ptsAdjustment="0" tier="4095">
            <scte35:TimeSignal>
                <scte35:SpliceTime ptsTime="0" />
            </scte35:TimeSignal>
            <scte35:SegmentationDescriptor segmentNum="0"
 segmentationDuration="27000000" segmentationEventCancelIndicator="false"
 segmentationEventId="1" segmentationTypeId="48" segmentsExpected="0">
                <scte35:SegmentationUpid segmentationUpidFormat="hexBinary"
 segmentationUpidType="14">012345</scte35:SegmentationUpid>
            </scte35:SegmentationDescriptor>
        </scte35:SpliceInfoSection>
    </Event>
</EventStream>
```

您学习了使用 SCTE-35 消息在频道集合中设置广告时段、这些消息的结构和必填字段，以及包含消息的 HLS 和 DASH 输出示例。 SCTE-35

对频道的播放进行时移

通过 MediaTailor 频道组合，您可以对频道的播放进行时移。时移允许观众在原始播出时间以外的时间观看频道的内容。例如，观众可以从头开始观看节目，即使他们在节目开始后才加入广播。或者，观众可以暂停节目，稍后再继续观看。

要对频道的播放进行时移，可以在该频道的播放 URL 中添加一个`start`参数。该`start`参数指定应开始播放的时间（相对于当前时间）。例如，如果当前时间是下午 2:00，而您想从下午 1:00 开始播放，则应将`start`参数设置为`-3600`，即 -1 小时（以秒为单位）。

该`start`参数接受正值和负值，以秒为单位：

- 负值表示相对于当前时间的过去时间。例如，`start=-3600`表示 “从 1 小时前开始播放”。
- 正值表示相对于当前时间的 future 时间。例如，`start=3600`表示 “将来 1 小时后开始播放”。

以下示例说明如何将`start`参数添加到频道的播放 URL：

```
https://a1b2c3d4e5f6.mediapackage.us-west-2.amazonaws.com/out/v1/examplechannel/
index.m3u8?start=-3600
```

在此示例中，播放从 1 小时前开始。

该`start`参数受以下限制：

- 过去或将来的最大时移窗口为 24 小时。
- 时移仅适用于使用线性回放模式的频道。
- 时移不适用于使用循环播放模式的频道。

如果您指定的`start`参数超出了可用的时移窗口，则 MediaTailor 会返回错误。

使用 CDN 优化 MediaTailor 广告个性化和内容交付

AWS Elemental MediaTailor 作为一项独立服务可以有效地运行，但是将其与内容分发网络 (CDN)（例如Amazon CloudFront 或其他第三方 CDNs）集成可以显著增强您的流媒体工作流程。当你需要向分布在不同地理位置的大量受众提供内容时，或者当你想确保在不同 AWS 地区实现一致的广告投放时，CDN 集成尤其有价值。

如果没有 CDN，您的观众会直接连接到以 MediaTailor 获取个性化的清单和广告细分，这可能会导致延迟增加，特别是对于远离您的 MediaTailor 配置部署 AWS 区域的观众而言。此外，在高流量事件期间，直接连接的负载 MediaTailor 可能会增加，从而可能影响性能。

有关 MediaTailor 概念和工作流程的更多信息，请参阅什么是 AWS Elemental MediaTailor?。

将 CDN 与集成时 MediaTailor，必须配置适当的 CORS（跨源资源共享）处理，以防止出现可能导致基于 Web 的播放器播放失败的问题。正确的 CORS 配置对于广告细分和内容细分都是必不可少的。虽然广告区段更容易受到CORS问题的影响，但对所有细分类型应用一致的CORS处理可以确保最可靠的播放体验。有关使用适当的 CORS 处理配置 CDN 路由行为的详细指南，请参阅。Production-ready CloudFront 的配置 MediaTailor

CDN 集成还支持高级参数传递和动态路由功能。有关通过传递 CDNs 用于授权和路由的查询参数的信息，请参见MediaTailor 清单查询参数。有关使用配置别名的动态广告服务器和源站路由，请参阅MediaTailor ADS 请求的动态广告变量。

在观众之间放置 CDN 并 MediaTailor 具有以下好处：

- 通过在离观众更近的边缘位置提供内容来减少延迟
- 通过在 CDN 的全球基础设施中分配负载来提高可扩展性
- 通过冗余的交付路径提高可靠性
- 通过减少源站流量来优化成本
- 实现媒体质量感知路由 (MQAR) 等高级功能，以提高直播质量

主题

- 根据您的需求选择合适的 CDN
- 规划您的 CDN 集成 AWS Elemental MediaTailor
- 通过以下方式设置 CDN 集成 MediaTailor
- 使用 CDN 设置 SSAI 以进行个性化视频广告

- 使用频道集合和 CDN 构建 MediaTailor 线性频道
- MediaTailor 与 CDN MediaPackage 集成
- AWS Elemental MediaTailor 与 Amazon 集成 CloudFront
- 为 MediaTailor 广告投放设置第三方 CDNs
- CDN 和 MediaTailor 集成的性能优化指南
- 监控 MediaTailor CDN 的运营和性能
- CDN 和集成的测试和 MediaTailor 验证
- 对 CDN 集成 MediaTailor 和 CDN 集成问题进行故障排除
- CDN 集成日志分析和错误代码参考 MediaTailor
- 使用自动化 MediaTailor 和 CDN CloudFormation
- Production-ready CloudFront 的配置 MediaTailor
- 获取 CDN 和集成的支持和 MediaTailor 故障排除帮助

根据您的需求选择合适的 CDN

选择合适的内容分发网络 (CDN) 提供商是一个重要的决定，它可能会影响您的内容交付性能、成本和观众体验 AWS Elemental MediaTailor。在为您的 MediaTailor实施选择 CDN 时，请考虑以下因素：

地理覆盖范围

选择在受众所在地区具有强大影响力的 CDN。不同的 CDN 提供商在不同的地理区域具有不同的优势。

与 AWS 服务集成

Amazon CloudFront 提供了与其他 AWS 服务最紧密的 MediaTailor 集成，这可以简化设置和管理。第三方 CDNs 可能会提供其他优势，例如专门的视频交付功能或在特定地区的影响力。

特定于视频的功能

寻找 CDNs 专门为视频交付设计的功能，例如自适应比特率优化、视频压缩和以观众体验为重点的分析。

成本结构

考虑流量量、地理分布和功能要求等因素，比较不同提供商的定价模式。有些 CDNs 提供批量折扣或承诺使用折扣，这些折扣可能与您的使用模式一致。

对高级功能的支持

确认您选择的 CDN 支持您需要的功能，例如令牌身份验证、地理限制、请求折叠和正确的标头转发。

有关特定 CDN 提供商及其与之集成的更多信息 MediaTailor，请参阅以下资源。

- CloudFront 集成适用于亚马逊 CloudFront
- 第三方 CDN 设置适用于第三方 CDN 提供商

以下主题提供了有关使用 CDN 进行配置 MediaTailor 以实现最佳性能的全面指导。

规划您的 CDN 集成 AWS Elemental MediaTailor

您可以通过集成 CDN 来改善观看者体验并减少延迟。 AWS Elemental MediaTailor在实施内容分发网络 (CDN) 时，您可以从离观众更近的地方提供内容。这可确保更快的加载时间、更好的可扩展性以及跨不同地理区域的广告投放一致性。

在使用实施 CDN 之前，您需要进行 AWS Elemental MediaTailor适当的规划。本节将引导您了解关键的规划领域。在开始实际配置之前，您需要先解决这些问题。这些步骤可帮助您为受众创造最佳的观看体验。

完成规划后，请参阅将 CDN 与集成，了解 step-by-step实施 MediaTailor说明。

有关可能影响您的 CDN 规划的 MediaTailor 配额的信息，请参阅中的配额 AWS Elemental MediaTailor。有关 CloudFront 配额的信息，请参阅《 CloudFront 开发人员指南》中的配额。

在开始计划之前，请先了解如何与 CDN MediaTailor 交互：

1. 观众通过您的 CDN 而不是直接从 CDN 请求内容 MediaTailor
2. CDN 会将清单请求转发到以 MediaTailor 进行个性化设置。
3. CDN 从边缘位置缓存和提供内容区段和广告区段。

这种架构可以减少负载， MediaTailor 同时确保观众以最小的延迟收到个性化广告。

了解清单术语有助于您正确配置 CDN。不同的流媒体协议使用特定的清单结构，这些结构会影响您设置缓存和路由的方式：

- HLS 清单 ——当你处理 HLS 直播时，你需要处理：
  - 多变体播放列表：将您的 CDN 配置为将这些顶级清单路由到以 MediaTailor 进行个性化设置。
  - 媒体播放列表：为这些包含内容分段链接的清单设置适当的缓存规则。
- DASH 清单 ——当你使用 DASH 直播时，你需要处理：
  - MPD（媒体演示描述）：配置您的 CDN 以根据您的个性化要求处理这些清单。

CDN 规划过程包括以下关键步骤，每个步骤都侧重于一项特定的任务：

- 估计 CDN 和 MediaTailor 集成的流量需求：计算您的预期观看者并发量和带宽需求。
- 为 CDN 和 MediaTailor 集成配置优化策略: 配置您的 CDN 以实现最佳内容交付和广告个性化。
- 定制 CDN 和 MediaTailor 集成的规划：根据您的特定 MediaTailor 工作流程调整您的 CDN 策略。
- 为 CDN 和集成设置监控和 MediaTailor 扩展: 实施监控和扩展策略以实现可靠的性能。
- 优化 CDN 和 MediaTailor 集成的成本：在性能与成本效益之间取得平衡。
- 测试您的实现 CDN 和集成 MediaTailor: 在生产部署之前，请彻底测试您的 CDN 集成。

估计 CDN 和 MediaTailor 集成的流量需求

要准确调整您的内容分发网络 (CDN) 规模，请与 AWS Elemental MediaTailor：

1. 使用历史数据或类似事件计算预期的观看者并发度。计划在基准之外增加容量，以应对意想不到的峰值。有关当前的扩展建议，请咨询您的 AWS 客户团队。你也可以看到中的配额 AWS Elemental MediaTailor。
2. 确定内容日程安排的高峰流量模式和潜在的峰值。考虑以下因素：
   - 体育赛事直播或赛季首播
   - 营销活动或促销活动
   - 全球受众的时区差异
   - 假日或季节性观看模式
3. 通过将观看者数量乘以直播比特率来确定您的带宽需求。与您的 CDN 提供商合作计算适当的容量。此计算基于您的特定内容比特率和预期的受众规模。按照提供商的建议，为广告细分和清单请求增加开销。
4. 与您的 CDN 提供商合作，确保目标区域有足够的边缘容量。

采取以下具体措施，确保您的广告投放能力满足观众的需求：

1. 在 Service Quotas 控制台中查看您当前的广告插入请求配额。查看当前的服务限制，了解您的配置可以支持多少并发查看器。
2. 对于高流量事件，请通过 Service Quotas 控制台申请增加配额。
3. 如果您预计同时观看人数将超过 500,000 人，请在活动开始前至少 2 周联系 AWS S upport。这样可以确保有足够的容量 AWS 来满足您的广告个性化需求。

有关在工作流程中实施容量规划的更多信息，请参阅使用预取计划来优化高流量事件的广告投放。

为 CDN 和 MediaTailor 集成配置优化策略

完成流量估算后，请配置内容分发网络 (CDN) 以优化内容交付和广告个性化。 AWS Elemental MediaTailor这些优化有助于确保流畅播放，同时保持定向广告。

实施以下特定的 CDN 优化：

1. 在 CDN 中配置 Origin shield 功能，以减少负载 MediaTailor 并提高缓存效率。Origin shield 充当中间缓存层，它：
   - 将多个查看者请求合并为一个原始请求
   - 将冗余请求的数量减少到 MediaTailor
   - 缩短缓存内容的响应时间

   有关使用设置 Origin Shield 的实现详情 CloudFront，请参阅《 CloudFront 开发者指南》中的 “使用 Origin Shield”。
2. 为不同的内容类型设置适当的生存时间 (TTL) 值。TTL 决定了 CDN 缓存内容的时间长度。在这段时间之后，CDN 会从源站请求一份全新的副本：
   - 清单：
     - 广告插播时间 0 秒
     - 频道组装 5-10 秒

     在广告插入中， MediaTailor 为清单提供针对观众的个性化广告。如果播放列表或 MPD 被缓存到错误的播放设备上，则设备可能会遇到播放或跟踪问题。
   - 内容片段：24 小时或更长时间（这些时间很少更改，您可以积极缓存它们以减少源负载）
   - 广告细分：24 小时或更长时间（广告内容通常会被观众重复使用，您可以将其缓存很长时间）

   有关所有 MediaTailor 工作流程的全面的 TTL 建议和缓存优化策略，请参阅针对 CDN 和 MediaTailor 集成的缓存优化。

有关在中配置缓存行为的详细说明 CloudFront，请参阅《 CloudFront 开发人员指南》中的缓存行为设置。

3. 在观众群体附近部署 CDN 边缘节点。与您的 CDN 提供商合作：
   - 根据观众的人口统计数据确定最佳边缘节点位置
   - 确保每个区域都有足够的容量
   - 监控边缘性能并根据需要进行调整

   有关实施指南，请参阅CloudFront 边缘位置，以确定受众区域的可用边缘位置。
4. 对于全球受众，可以考虑实施多 CDN 策略。这种方法：
   - 使用多个 CDN 提供商来提高可靠性
   - 根据其所在位置将观众引导至性能最佳的 CDN
   - 在 CDN 中断期间提供故障转移选项
   - 可以通过利用不同的定价模式来优化成本

   有关实现的详细信息，请参阅AWS 网络和内容交付博客上的多 CDN 策略。

定制 CDN 和 MediaTailor 集成的规划

不同 AWS Elemental MediaTailor 的工作流程有独特的要求，这些要求会影响内容交付网络 (CDN) 的规划。有关特定于工作流的指导，请参阅使用配置。

根据您的特定 MediaTailor 工作流程调整容量计划：

用于 MediaTailor 广告插入工作流程

1. 将您的 CDN 配置为无需缓存即可处理个性化清单。这样可以确保每位观众都能收到独特的定向广告。广告基于其个人资料和观看背景。
2. 调整广告决策服务器 (ADS) 的大小，以应对峰值请求量。有关 ADS 配置的指导，请参阅广告插入 MediaTailor。考虑：
   - 您的用例的响应时间要求
   - 预期的并发查看器容量
   - 冗余和故障转移要求
   - 地理分布需求

3. 在 CDN 级别实现请求折叠，以处理同步的广告插播请求。请求折叠将多个相同的请求合并为一个原始请求。这在以下情况下至关重要：
   - 许多观众同时观看广告时段的体育赛事
   - 热门电视节目首播并同步广告片段
   - 突发新闻事件吸引观众突然激增

   有关实现的详细信息，请参阅 Origin 故障转移以配置高峰负载期间的请求处理。

适用于 MediaTailor 通道组装工作流程

1. 根据信道数量及其比特率计算容量需求。有关渠道装配容量规划的指导，请参阅中的渠道装配 MediaTailor。考虑：
   - 频道总数
   - 每个通道的比特率要求
   - 预期的并发查看器负载
   - 地理分布需求
2. 配置您的 CDN 以根据已发布的时间表处理可预测的流量模式。渠道组合通常比广告插播具有更可预测的模式，因为：
   - 编程时间表是事先知道的
   - 观看者的行为遵循既定模式
   - 内容不会因观看者而动态变化
3. 确保您的源站有足够的带宽来保持一致的频道输出。实施：
   - 用于高可用性通道的冗余源服务器
   - 在主源和备份源之间自动进行故障转移
   - 监控以检测源站性能问题

   有关实施指南，请参阅设置源站冗余以创建弹性源基础架构。

适用于组合 MediaTailor 工作流程

1. 调整基础架构的规模，以应对两种服务的组合流量模式。有关组合工作流程的指导，请参阅使用 AWS Elemental MediaTailor 插入广告。考虑：
   - 渠道组装基准要求

- 广告插入开销要求
- 高峰流量模式
- 冗余需求

2. 为线性内容交付和动态广告插入配置单独的 CDN 行为。这种分离使您能够：
   - 单独优化每种内容类型的缓存策略
   - 根据内容类型将请求路由到适当的来源
   - 分别监控每个工作流程的性能指标
3. 在边缘和原点之间设置适当的布线 CDNs ，以保持最佳性能。考虑使用：
   - 内容细分 (/content/*) 和广告细分 (/ads/*) 的来源路径不同
   - 将清单和区段的缓存行为分开
   - 地理路由，可优化不同区域的延迟

   有关实现的详细信息，请参阅配置缓存行为以设置基于路径的路由和缓存规则。

为 CDN 和集成设置监控和 MediaTailor 扩展

有效的监控和扩展策略对于通过 AWS Elemental MediaTailor 内容交付网络 (CDN) 集成保持最佳性能和观众体验至关重要。实施这些方法可确保您的 CDN 集成大规模可靠运行。

实施以下监控和扩展策略：

1. 为这些关键指标配置监控。有关适当目标值的指导，请参阅 Amazon 监控 MediaTailor CloudWatch并查阅 CDN 提供商的最佳实践：
   - CDN 缓存命中率（根据您的内容类型和交付模式建立基准指标和目标）
   - 原始请求量（在正常操作期间监控模式以建立异常检测基准）
   - 按错误类型划分的错误率（根据您的服务级别目标和 MediaTailor 最佳实践定义阈值）
   - 响应时间（根据您的观众体验要求和地理分布设置适当的延迟目标）

   有关详细的实施说明，请参阅创建 CloudWatch 仪表板以将您的指标 MediaTailor 和 CDN 指标一起可视化。
2. 针对意外流量模式或性能下降设置警报。根据您的基准指标和服务级别目标配置阈值。有关设置提醒的指导，请参阅创建 Amazon CloudWatch 警报。考虑监控：
   - 与基准缓存命中率存在显著偏差（通常在低于 85-90% 时发出警报）
   - 源站请求量突然增加（比基线增加 30% 或更多时发出警报）

- 错误率峰值超过您定义的阈值（4xx 错误通常为 1-2%，5xx 错误通常为 0.5%）
- 响应时间降级超过可接受的水平（清单通常大于 500 毫秒，区段通常超过 200 毫秒）

有关实施示例，请参阅创建有效监控仪表板和警报的CloudWatch 概念。

3. 为可预测的高流量事件制定扩展计划。您的计划应包括以下关键要素：
   - 活动前容量增加（活动开始前 24-48 小时）
   - 逐步增加观众的时间表（通常每 5 分钟间隔占预期观众的 10-20%）
   - 基于受众的区域容量分布（根据预期的区域收视率按比例分配容量）
   - 事件后扩展程序（活动结束后在 30-60 分钟内保持峰值容量）

   有关针对高流量事件进行扩展的实施指南，请参阅 AWS 媒体博客上的 “设置弹性 end-to-end实时工作流程”。
4. 为关键数据流实施故障转移和冗余措施，包括：
   - 多区域 CDN 部署（至少两个区域用于关键内容）
   - Backup 源服务器（配置为每 30 秒自动进行一次运行状况检查）
   - 根据运行状况检查自动触发故障转移（通常在 2-3 次检查失败之后）
   - 针对不同故障情形的恢复程序（记录在具体的响应时间目标中）

   有关详细的实施步骤，请参阅使用 CloudFront 源故障转移优化高可用性。

优化 CDN 和 MediaTailor 集成的成本

内容分发网络 (CDN) 的成本可能会因流量模式、地理分布和功能使用情况而有 AWS Elemental MediaTailor很大差异。有关现行定价信息，请参阅 CloudFront 定价。您也可以查阅 CDN 提供商的文档。

使用以下策略平衡性能和成本效率：

1. 分析您的 CDN 流量模式，选择最具成本效益的定价等级。与您的 CDN 提供商一起查看以下因素：
   - 按地区和时间段划分的数据传输量
   - 清单和区段的请求模式
   - 地域分布要求
   - 峰值与平均使用模式

~~如需成本分析方面的帮助，请使用AWS 定价计算器根据您的特定使用模式估算 CloudFront 成本。~~

2. 对于可预测的工作负载，请与您的 CDN 提供商评估预留容量协议。这些协议可以提供以下好处：
   - 承诺使用量的折扣费率
   - 可预测的每月预算成本
   - 优先支持和容量分配

   请咨询您的 CDN 提供商，以确定预留容量是否适合您的使用模式。有关详细信息 CloudFront，请参阅CloudFront 高级功能。这提供了有关预留容量选项的信息。
3. 通过平衡与 CDN 提供商 MediaTailor 之间的流量来优化出口成本。策略包括：
   - 最大限度地提高缓存命中率以减少源站请求
   - 使用原点盾牌整合请求
   - 实施压缩以减少数据传输量
   - 选择与区域定价一致的 CDN AWS 区域

   有关压缩的实施指南，请参阅《 CloudFront 开发人员指南》中的提供压缩文件。
4. 针对不同的内容类型实施适当的缓存策略，以减少源站请求。有关缓存优化的指导，请参阅提高缓存命中率。重点关注：
   - 内容分段（您可以将其缓存很长一段时间）
   - 广告细分（通常在观众中重复使用）
   - 静态资产，例如玩家文件和图像

   提高的缓存命中率可显著降低源站成本。与您的 CDN 提供商合作，针对您的特定内容模式优化缓存配置。有关详细的实现步骤，请参阅配置缓存行为。

测试您的实现 CDN 和集成 MediaTailor

在生产部署之前，通过全面测试 AWS Elemental MediaTailor 内容交付网络 (CDN) 集成，确保可靠的观众体验。适当的测试有助于在问题影响受众之前识别和解决问题。有关测试方法的指导，请参阅测试 CloudFront 发行版。您也可以查阅 CDN 提供商的测试文档。

请按照以下测试步骤验证您的 CDN 集成：

1. 创建镜像您的生产配置的测试环境。Include：
   - 相同的 CDN 设置和缓存行为
   - 包括具有不同比特率和格式的代表性内容
   - 使用示例广告响应配置测试广告决策服务器

- 设置监控和警报配置

有关 step-by-step实施指南，请参阅《 CloudFront 开发人员指南》中的创建暂存发行版。

2. 进行负载测试以验证您的容量估计。有关负载测试的指导，请参阅监控 MediaPackage。测试场景应包括：
   - 观众人数逐渐增加（通常每 5 分钟增加预期峰值的 10-20%）
   - 根据您的预期峰值负载突然出现流量峰值（模拟 50% 的峰值受众在 60 秒内加入）
   - 延长峰值负载周期（将峰值负荷保持至少 30-60 分钟）
   - 与您的受众群体相匹配的地理分布（根据预期的观众位置分配测试流量）

   验证响应时间是否保持在目标阈值以下。通常，这意味着清单小于 500 毫秒，分段小于 200 毫秒。错误率应保持在 1% 以下。有关负载测试工具和方法的实施细节，请参阅 AWS 网络和内容交付博客 CloudFront上的负载测试。
3. 测试故障转移场景以确保可靠性。模拟：
   - 源服务器故障（完全中断和部分降级情况）
   - CDN 边缘位置中断（使用流量路由到备份位置进行测试）
   - 广告决策服务器不可用（测试时超时 5-10 秒）
   - 网络连接问题（模拟数据包丢失和延迟增加）

   与您的 CDN 提供商合作，为您的用例建立适当的故障转移响应时间目标。通常，这意味着故障转移完成时间不到 3 秒。有关故障转移测试的实施指南，请参阅《 CloudFront 开发人员指南》中的 Origin 故障转移。
4. 对于重大事件，请根据AWS 负载测试指南实施渐进式提升策略：
   - 错开不同受众群体的开始时间。例如，首先允许高级订阅者在受众群体之间间隔15分钟。
   - 使用预热技术逐渐增加负荷。预热涉及：
     - 活动开始前 2-4 小时用热门内容填充 CDN 缓存
     - 逐渐将合成流量增加到预热系统预期峰值的 20-30%
     - 在实际负载条件下使用实际内容测试所有组件
   - 在整个升级期间监控系统性能，跟踪：
     - 缓存命中率和响应时间（目标命中率 > 90%，响应时间小于 500 毫秒）
     - 错误率和源站负载（保持错误率 < 1%，原点 CPU < 70%）
     - 广告个性化成功率（目标个性化成功率 > 98%）
     - 观看者体验指标（目标 <2 秒启动时间，< 0.5% 的重新缓冲）

- 制定应对意外交通激增的应急计划。您的计划应包括以下基本组成部分：
  - 紧急容量扩展程序，其中包含记录在案的步骤，可在 15 分钟内将容量提高 50-100%
  - Backup CDN 激活协议，能够将 20-50% 的流量转移到辅助 CDN
  - 简化了广告插入后备功能，将定位参数从 10 个或更多减少到 3-5 个基本参数
  - 使用预定义的通知模板和联系人列表为利益相关者制定沟通计划

完成测试后，继续为生产部署步骤实施 CDN 集成。

排除 CDN 和 MediaTailor 集成的常见问题

在常见的内容交付网络 (CDN) 集成难题影响您的观众 AWS Elemental MediaTailor 之前，解决这些难题。本节可帮助您识别和解决 CDN 与 AWS Elemental MediaTailor集成期间发生的典型问题。

有关全面的故障排除指南，请参阅故障排除 MediaTailor和故障排除 CloudFront 发行版。

解决 MediaTailor 舱单配送问题

如果观众遇到播放问题或看到的广告不正确，请查看以下与清单相关的常见问题：

- 缓存设置不正确：如果您的 CDN 缓存个性化清单，则观众可能会看到面向其他用户的广告。

  解决方案：将 CDN 的清单请求配置为 0 的缓存 TTL。 MediaTailor
- 源站请求失败：如果您的 CDN 无法访问 MediaTailor，则清单请求将失败。

  解决方案：检查您的 CDN 和 MediaTailor之间的网络连接。验证您的 CDN 是否正确转发了正确的标头。
- 会话参数问题：会话参数缺失或不正确可能会导致个性化失败。

  解决方案：确保您的玩家正确地将所有必需的会话参数附加到清单请求中。

修复 MediaTailor 区段交付问题

如果内容或广告区段加载不正确，请调查以下常见问题：

- 分段路径重写：不正确的 CDN 配置可能会不正确地重写分段。 URLs

  解决方案：验证您的 CDN 是否正确处理分段 URLs。确保它不会以破坏引用的方式修改路径。
- CORS 配置：CORS 标头缺失或不正确可能会导致浏览器无法加载区段。

解决方案：将您的 CDN 配置为为分段请求传递相应的 CORS 标头。

- 缓存未命中风暴：在高流量事件期间，多个缓存失误可能会使源服务器不堪重负。

  解决方案：实现请求折叠和源站屏蔽功能，以减少流量高峰期间的源站负载。

解决 MediaTailor CDN 性能问题

如果观看者遇到缓冲或加载缓慢的情况，请检查以下与性能有关的问题：

- 缓存命中率低：如果您的 CDN 经常从源站请求内容，则性能将受到影响。

  解决方案：按内容类型分析缓存命中率并调整 TTL 设置以提高缓存效率。

- 地理分布：远离 CDN 边缘位置的观看者可能会遇到延迟增加的情况。

  解决方案：查看您的 CDN 边缘位置分布。在观众高度集中的区域增加容量。

- 源站容量限制：如果您的源服务器过载，则响应时间将增加。

  解决方案：实现源站请求限制。您还可以增加源容量或改善缓存以减少源负载。

如需其他疑难解答帮助，请参阅疑难解答 MediaTailor或联系 Supp AWS ort。有关解决常见 CDN 问题的实施指南，请参阅 AWS 媒体博客上的调试内容分发网络。

通过以下方式设置 CDN 集成 MediaTailor

本节提供有关 AWS Elemental MediaTailor 与内容分发网络 (CDN) 集成的指导。

与 CDN 的有效集成 MediaTailor 对于通过大规模个性化广告提供高质量的流媒体体验至关重要。本指南将引导您完成设置、配置和优化 CDN 集成的完整过程。

有关更多信息，请参阅以下链接：

- 有关传递 CDNs 用于授权和广告定位的查询参数的信息，请参阅MediaTailor 清单查询参数。
- 有关使用动态变量和配置别名的高级路由，请参阅MediaTailor ADS 请求的动态广告变量。
- 有关创建 MediaTailor 配置的信息，请参阅使用 AWS Elemental MediaTailor 插入广告。
- 有关创建 CloudFront 分配的信息，请参阅《 CloudFront 开发者指南》中的创建分配。

为确保本指南的清晰度，使用以下术语来描述不同类型的清单：

- HLS 表现为：
  - 多变播放列表：包含媒体播放列表链接的顶级清单
  - 媒体播放列表：包含内容分段链接的二级清单
- DASH 表现为：
  - MPD（媒体演示文稿描述）：DASH 清单的标准术语

MediaTailor 与 CDN 集成具有以下好处：

- 减少观众的延迟
- 提高了高流量事件的可扩展性
- 通过冗余传送路径提高可靠性
- 通过减少源站流量来优化成本
- 更好地防御 DDo S 攻击

CDN 集成组件和要求

要成功与 CDN 集成， MediaTailor 需要配置以下关键组件：

CDN 路由行为

用于确定不同类型的请求（清单、内容区段、广告区段）如何通过您的 CDN 路由的规则。

CDN 映射在 MediaTailor

中的配置可 MediaTailor 确保清单引用您的 CDN 域而不是直接引用源服务器。

安全设置

保护您的内容和基础架构的配置，包括传输安全、访问控制和监控。

测试和验证

在部署到生产环境之前，验证您的 CDN 集成是否正常运行的程序。

CDN 集成的先决条件

在配置 CDN 集成之前，请确保具备以下条件：

1. 具有以下设置的 MediaTailor 配置：

- 您的内容来源作为内容来源
- 你的 ADS 作为广告决策服务器

在 CDN 集成步骤 URLs 中，您还需要来源和 ADS。

2. 访问您的 CDN 的配置接口
3. 了解 CDN 关于行为、规则和缓存设置的特定术语
4. 了解您的内容结构，包括用于区段（例如.ts、.mp4 或.m4s）的文件扩展名

CDN 集成设置步骤

MediaTailor 与 CDN 集成的过程遵循以下高级步骤：

1. 配置 CDN 路由行为-将 CDN 设置为适当地路由不同类型的请求。
2. 在中配置 CDN 映射 MediaTailor-更新您的 MediaTailor 配置以使用您的 CDN 域名。
3. 实施安全最佳实践-配置安全设置以保护您的内容和基础架构。
4. 测试您的集成-验证您的 CDN 集成是否正常运行。

MediaTailor CDN 集成所需的标头

MediaTailor 为了使您的 CDN 正常运行，您必须将 CDN 配置为转发特定的 HTTP 标头。这些标题对于实现适当的功能（包括压缩、设备检测、广告个性化和地理位置定位）至关重要。

将您的 CDN 配置为将以下标头转发到 MediaTailor：

`Accept-Encoding`

用途：压缩功能所必需的

详细信息：此标头说明客户端支持 MediaTailor 哪些压缩方法。 MediaTailor 尽可能使用此信息压缩清单，从而减少带宽使用量并提高性能。不支持压缩的旧设备不会发送此标头，而是 MediaTailor 会返回未压缩的清单。

`User-Agent`

用途：设备检测和广告个性化所必需的

详细信息： MediaTailor 分析 User-Agent 标头以识别客户端设备类型、浏览器和功能。这些信息用于广告定位、设备特定的优化以及确保与不同播放客户端的兼容性。

`Host`

用途：正确请求路由所必需的

详细信息：标`Host`头可确保将请求路由到正确的 MediaTailor 端点。这在多租户环境和使用自定义域配置时尤其重要。

许多公司 CDNs（包括Amazon CloudFront）在默认情况下不转发`Host`标题。对于 CloudFront 用户：有关配置说明配置缓存行为，请参阅。

`X-Forwarded-For`

用途：客户端 IP 检测和地理定位所必需的

详细信息：当请求通过您的 CDN 时，此标头会保留原始客户端 IP 地址。 MediaTailor 使用此信息进行地理广告定位、分析和遵守区域内容限制。

> **Important**
>
> 全部 MediaTailor 功能都需要全部四个标头。缺少这些标头中的任何一个都可能导致功能降低，包括：
>
> - 无法压缩清单（缺少接受编码）
> - 广告定位不佳和设备兼容性问题（缺少用户代理）
> - 请求路由失败（缺少主机）
> - 地理定位和分析不准确（缺失）X-Forwarded-For

有关特定于 CDN 的配置说明，请参阅引用此标题列表的路由行为和缓存部分。

以下主题提供了 CDN 集成的各个方面的详细说明。

主题

- 为以下各项设置 CDN 路由行为 MediaTailor
- 在中设置 CDN 映射 MediaTailor
- CDN 集成安全最佳实践 MediaTailor

为以下各项设置 CDN 路由行为 MediaTailor

本节介绍如何设置内容分发网络 (CDN)，以适当地路由不同类型的请求以进行 AWS Elemental MediaTailor 集成。正确的路由配置可确保清单请求、内容区段和广告区段得到正确处理。

配置 CDN 路由行为是创建高效内容交付管道的关键步骤。通过为不同的内容类型设置特定的路由规则，您可以优化缓存，提高投放性能，并确保个性化广告插入正常运行。

有关使用动态变量和配置别名的高级路由方案，请参阅MediaTailor ADS 请求的动态广告变量。有关在 CDN 路由中保留查询参数的信息，请参阅MediaTailor 清单查询参数。

> **Important**
>
> 如果不在缓存的对象中或 CDN 对查看者的响应中包含 CORS 标头，可能会导致播放失败。

CDN 路由行为配置

设置您的 CDN 以适当地路由不同类型的请求。

内容分段路由

内容分段路由将对您的实际内容分段的请求定向到您的源服务器。与广告区段路由一样，内容区段路由也需要适当的 CORS 配置，以确保在基于 Web 的播放器中流畅播放。

有关详细的配置指南，请参阅中的 CloudFront 示例优先级 4：内容来源路径行为。此示例提供了您应遵循 CloudFront 或适应其他设置的特定设置 CDNs。

内容分段路由的关键配置要求包括：

- 使用与内容分段文件扩展名相匹配的路径模式（比如`*.ts`、`*.mp4`或`*.m4s`）
- 将请求路由到您的内容来源（例如 Amazon S3 存储桶或 MediaPackage终端节点）
- 为了获得最佳的缓存命中率，请仅包含查询字符串参数，这些参数会使您的源修改缓存密钥中的响应并转发源请求
- 应用适当的 TTL 值大于 24 小时的缓存策略
- 向您的查看者添加 CORS 响应标题

广告区段路由

广告细分路由对于向观众投放个性化广告至关重要。配置广告区段路由时，您必须实施适当的 CORS（跨源资源共享）处理，以防止出现可能导致基于 Web 的播放器播放失败的问题。

有关详细的配置指南，请参阅中的 CloudFront 示例优先级 0：广告区段路径行为。此示例提供了您应遵循 CloudFront 或适应其他设置的特定设置 CDNs。

广告区段路由的关键配置要求包括：

- 使用`/tm/*`专门针对 MediaTailor 广告细分的路径模式
- 将请求路由到 `segments.mediatailor.region.amazonaws.com`
- 为了获得最佳的缓存命中率，请勿在缓存密钥或原始请求中包含任何查看者请求标头、Cookie 或查询字符串参数
- 应用适当的 TTL 值大于 24 小时的缓存策略
- 向您的查看者添加 CORS 响应标题

清单请求路由

要将多变体播放列表、媒体播放列表和 MPD 请求路由到 MediaTailor，请使用以下常规设置。有关 CloudFront 配置，请参阅配置清单缓存行为。

1. 在您的 CDN 配置界面中，为不同的清单类型创建行为。
2. 设置路径模式以匹配多变体播放列表和媒体播放列表文件扩展名（`*.m3u8`对于 HLS）和 MPD 文件扩展名（`*.mpd`对于 DASH）。
3. 在 CDN 中配置源设置，使其指向您的 MediaTailor配置端点。
4. 要插入广告，请禁用缓存个性化多变体播放列表、媒体播放列表和。 MPDs由于广告插入会提供个性化清单，因此您的 CDN 不应将其缓存。如果与预期设备不同的播放设备接收到缓存的播放列表或 MPD，则可能会导致播放或跟踪出现问题。有关全面的缓存指南，包括针对所有内容类型的 TTL 建议，请参阅针对 CDN 和 MediaTailor 集成的缓存优化。
5. 为所有标头配置标头转发。有关最低要求，请参阅MediaTailor CDN 集成所需的标头。
6. 启用查询字符串转发功能以传递广告定位参数。

HLS multivariant playlist

HLS 多变体播放列表请求遵循以下格式：

```
https://<playback-endpoint>/v1/index/<hashed-account-id>/<origin-id>/<index>.m3u8
```

示例：

```
https://777788889999.mediatailor.us-east-1.amazonaws.com/v1/master/
a1bc06b59e9a570b3b6b886a763d15814a86f0bb/Demo/assetId.m3u8
```

HLS media playlist

HLS 媒体播放列表请求遵循以下格式：

```
https://<playback-endpoint>/v1/manifest/<hashed-account-id>/<session-id>/
<manifestNumber>.m3u8
```

玩家对https://CDN_Hostname/some/path/asset.m3u8的请求将https://mediatailor.us-west-2.amazonaws.com/v1/session/configuration/endpoint根据请求*.m3u8中的关键字路由到 MediaTailor 路径。

示例：

```
https://777788889999.mediatailor.us-east-1.amazonaws.com/v1/manifest/
a1bc06b59e9a570b3b6b886a763d15814a86f0bb/c240ea66-9b07-4770-8ef9-7d16d916b407/0.m3u8
```

DASH MPD

DASH MPD 请求遵循以下格式：

```
https://<playback-endpoint>/v1/dash/<hashed-account-id>/<origin-id>/<assetName>.mpd
```

玩家对https://CDN_Hostname/some/path/asset.mpd的请求将https://mediatailor.us-west-2.amazonaws.com/v1/dash/configuration/endpoint根据请求*.mpd中的关键字路由到 MediaTailor 路径。

示例：

```
https://777788889999.mediatailor.us-east-1.amazonaws.com/v1/dash/
a1bc06b59e9a570b3b6b886a763d15814a86f0bb/Demo/0.mpd
```

CDN 路由最佳实践

配置 CDN 路由行为时，请遵循以下最佳实践以确保最佳性能和可靠性：

使用特定的路径模式

创建与您的内容结构精确匹配的特定路径模式，以确保正确的路由。

确定行为顺序的优先级

在大多数情况下 CDNs，行为是按顺序评估的。将更具体的行为放在更一般的行为之前。

测试行为模式

在部署到生产环境之前，请验证您的路径模式是否与预期的请求正确匹配。

记录您的配置

保存 CDN 路由行为的文档，以便于故障排除和 future 更新。

后续步骤

配置 CDN 路由行为后，下一步是在中配置 CDN 映射。 MediaTailor有关说明，请参阅在中设置 CDN 映射 MediaTailor：

在中设置 CDN 映射 MediaTailor

本节介绍如何配置 AWS Elemental MediaTailor 为使用您的内容分发网络 (CDN) 域名。设置 CDN 路由行为后，您需要更新 MediaTailor 配置以确保清单引用您的 CDN 域而不是直接引用源服务器。

在中配置 CDN 映射 MediaTailor 可确保清单中的所有内容和广告区段 URLs 都指向您的 CDN，而不是直接指向源服务器。此步骤对于创建完整的 CDN 交付链和最大限度地发挥 CDN 集成的优势至关重要。

CDN 映射配置在 MediaTailor

设置 CDN 路由行为后，请配置 MediaTailor 为使用您的 CDN 域名：

1. 打开 MediaTailor 控制台。
2. 选择要更新的配置。
3. 在 CDN 配置部分的高级设置中，在 CDN 内容分段前缀字段中输入您的 CDN 域名。
4. 如果您为广告细分使用单独的 CDN 网域，请在 CDN 广告区段前缀字段中输入该域名。

5. 保存更改。

此配置可确保 MediaTailor 生成指向您的 CDN 而不是直接指向来源或广告细分存储的清单。 URLs

了解基本 URL 行为

MediaTailor 根据您 URLs 的 CDN 前缀配置确定清单的基础：

- 已配置 CDN 广告区段前缀：广告区段使用 CDN 前缀作为基本网址。
- 未配置 CDN 广告区段前缀：广告区段使用直接 MediaTailor 位置作为基本网址。
- 已配置 CDN 内容分段前缀：内容分段使用 CDN 前缀作为基本 URL。
- 未配置 CDN 内容分段前缀：内容分段引用原始内容来源。

DASH baseUrl 处理

对于 DASH 清单，对内容和广告细分的`BaseURL`设置进行不同的 MediaTailor 管理：

内容细分：

- 带有 CDN 内容分段前缀：使用您指定的前缀`BaseURL`在`MPD`级别上精确 MediaTailor 设置一个。
- 没有 CDN 内容分段前缀： MediaTailor 保留源清单中的现有`BaseURL`设置，或者如果不存在源网址，则根据来源 `MPD` URL 添加一个设置。

广告细分：

- 使用 CDN 广告区段前缀：`BaseURL`使用配置的前缀，每个广告时段恰好有一个广告周期。
- 没有 CDN 广告区段前缀：每个广告周期只有一个`BaseURL`指向 MediaTailor 广告内容服务器的广告时段。

Example CDN 映射示例

如果你的内容来源是`http://origin.com/contentpath/`，你的 CDN 内容分段前缀是`https://cdn.example.com/`，那么通常被引用的内容分段`http://origin.com/contentpath/subdir/content.ts`将在清单中显示为`https://cdn.example.com/subdir/content.ts`。

重要注意事项

在中配置 CDN 映射时 MediaTailor，请记住以下重要注意事项：

使用 HTTPS 作为 CDN 前缀

请务必使用 HTTPS URLs 作为 CDN 前缀，以确保内容交付安全。

匹配 CDN 行为路径

确保您配置的 CDN 前缀与您在 CDN MediaTailor 行为中配置的路径模式相匹配。

考虑区域性 CDNs

如果您为不同的区域使用不同的 CDN 域名，则需要为每个区域创建单独的 MediaTailor 配置。

验证域所有权

确保您可以控制您在中 MediaTailor配置的 CDN 域。

验证 CDN 映射配置

在中配置 CDN 映射后 MediaTailor，请验证您的配置是否正常运行：

1. 通过您的 CDN 索取清单。
2. 检查清单内容以验证该区段 URLs 引用您的 CDN 域。
3. 检查清单 URLs 中的内容区段是否指向你的 CDN 域。
4. 检查清单 URLs 中的广告区段是否指向你的 CDN 网域。

有关全面的测试和验证程序，请参阅CDN 和集成的测试和 MediaTailor 验证。

后续步骤

在中配置 CDN 映射后 MediaTailor，下一步是为您的 CDN 集成实施安全最佳实践。有关说明，请参阅CDN 集成安全最佳实践 MediaTailor：

CDN 集成安全最佳实践 MediaTailor

本节介绍如何为 AWS Elemental MediaTailor 内容分发网络 (CDN) 集成实施安全最佳实践。适当的安全配置有助于保护您的内容和基础设施免受未经授权的访问和潜在威胁。

保护 CDN 集成对于保护您的内容、防止未经授权的访问以及确保遵守安全要求至关重要。实施这些最佳实践有助于为您的流媒体工作流程创建稳健的安全态势。

CDN 安全配置

在配置 CDN 路由和缓存设置之前，请实施以下安全最佳实践来保护您的内容和基础架构：

运输安全

MediaTailor 仅使用 HTTPS 进行所有通信，不允许 HTTP 连接。为确保直播工作流程中所有组件之间的安全通信，请完成以下步骤：

1. 为 CDN 和 MediaTailor终端节点之间的所有通信配置 HTTPS。 MediaTailor 需要 HTTPS 并且不接受 HTTP 连接。
2. 所有 HTTPS 连接均使用 TLS 1.2 或更高版本。
3. 配置您的 CDN 以强制观看者仅通过 HTTPS 进行连接。

访问控制

要保护您的内容和基础设施免遭未经授权的访问，请实施以下访问控制：

1. 如果您需要将内容访问限制在特定区域，请配置地理限制设置。
2. 为需要查看者身份验证的内容实施签名 URLs 或签名 Cookie。
3. 设置基于 IP 的许可名单，以便对 CDN 配置进行管理访问。

安全监控

要有效地检测和响应安全事件，请实施以下监控实践：

1. 为 CDN 分发启用访问日志记录。
2. 为异常流量模式或访问尝试设置警报。
3. 定期检查安全配置并根据需要进行更新。

高级安全特征

以下高级安全功能为您的 CDN 集成提供了增强的保护：

Web 应用程序防火墙（WAF）

如果您的 CDN 支持 WAF 功能，请将其配置为防范常见的 Web 漏洞和攻击。

DDoS 保护

启用 CDN 提供的 DDo S 保护功能以缓解分布式 denial-of-service攻击。

源站访问控制

将您的源服务器配置为仅接受来自 CDN 的请求，以防止直接访问尝试。

内容加密

对于高度敏感的内容，可以考虑在传输安全性之外实施其他加密机制。

后续步骤

实施安全最佳实践后，下一步是测试您的 CDN 集成并对其进行故障排除。有关全面对 CDN 集成进行故障排除的测试和故障排除说明，请参阅。

使用 CDN 设置 SSAI 以进行个性化视频广告

本节提供了有关将 AWS Elemental MediaTailor 服务器端广告插入 (SSAI) 与内容分发网络 (CDN) 集成的全面指导。按照以下步骤设置、配置和优化 SSAI CDN 集成。

服务器端广告插入 (SSAI) 是一种在服务器级别而不是客户端级别将个性化广告无缝插入到视频流中的技术。与 CDN 结合使用时，您可以创建一个强大、可扩展的解决方案，以最小的延迟向全球受众投放个性化广告。

在本主题中，我们使用清单一词来统称多变体播放列表、媒体播放列表和。 MPDs

你需要什么

在使用 CDN 设置 MediaTailor 广告插播之前，请收集以下所需资源：

AWS 账户和权限

具有创建和管理 MediaTailor资源的适当权限的 AWS 账户

CloudFront （如果使用）和相关服务的 IAM 权限 MediaTailor

有关详细的权限要求，请参阅中的安全性 AWS Elemental MediaTailor。

必需的服务

AWS Elemental MediaTailor 已配置并正在运行

内容分发网络 (CDN) 账户（Amazon CloudFront 或第三方 CDN）

您的内容的源服务器（HLS 或 DASH）

支持 VAST 或 VMAP 的广告决策服务器 (ADS)

内容要求

内容已正确编码并以 HLS 或 DASH 格式打包

内容中的广告插播标记（适用于 VOD）或 SCTE-35 标记（用于直播）

开始前的准备工作

在使用 CDN 实现 MediaTailor 广告插入之前，请完成以下设置任务：

1. 配置 CDN 和源服务器之间的网络连接 MediaTailor
2. 设置 HTTPS 以实现安全的内容传输
3. 为您的 CDN 域名配置 DNS 设置
4. 要进行基本 MediaTailor 设置，请完成设置和中的步骤MediaTailor 广告插入入门。

知识先决条件

要成功实施此解决方案，您应该：

- 了解流媒体协议 (HLS/DASH)
- CDN 配置原理的基础知识
- 熟悉广告插入概念

CDN 集成的好处

将 SSAI 与 CDN 集成可带来以下主要好处：

改善了观众体验

正确配置 CDNs 可减少广告过渡期间的缓冲、启动时间和播放错误。这可以提高观众的参与度和满意度。

降低成本

高效的缓存策略可最大限度地减少源请求。这降低了数据传输成本和源服务器负载，这对于大量广告支持的内容尤其重要。

可扩展性

经过优化的 CDN 配置可在不降低性能的情况下处理热门活动期间的流量峰值，从而确保即使在观看高峰时段也能投放您的个性化广告。

全球影响力

正确配置 CDNs 向世界各地的观众提供低延迟的内容，无论他们身在何处，从而扩大您的潜在受众。

无缝的广告过渡

经过优化的 CDN 配置可确保内容和广告之间的平稳过渡，从而创造广播级质量的观看体验。

以下主题提供了有关使用 CDN 进行配置 MediaTailor 以实现最佳性能的全面指导。

主题

- 了解 CDN 和 MediaTailor 集成的广告插入架构
- 使用 CDN 设置基本 MediaTailor SSAI 以实现最佳广告投放
- 将 MediaTailor SSAI 与渠道组合集成，实现货币化线性通道
- 优化 CDN 和 MediaTailor 集成的性能
- 监控 CDN 和 MediaTailor 集成的运营
- 使用 MediaTailor SSAI 进行故障排除， CDNs 实现不间断的广告投放

了解 CDN 和 MediaTailor 集成的广告插入架构

本节介绍服务器端广告插入 (SSAI) 的概念和架构，内容分发网络 (CDNs) 适用于。 AWS Elemental MediaTailor您将了解动态广告插入和清单操作如何协同工作，从而实现有效的视频获利。

服务器端广告插入 (SSAI) MediaTailor 允许您：

- 在定义的广告断点将个性化广告插入到您的视频流中
- 根据观众数据精确定位广告
- 消除对客户端广告插入技术的需求

与 CDN 结合使用时，您可以向观众提供这些个性化直播，同时提高性能和可扩展性，从而增强您的视频盈利策略。

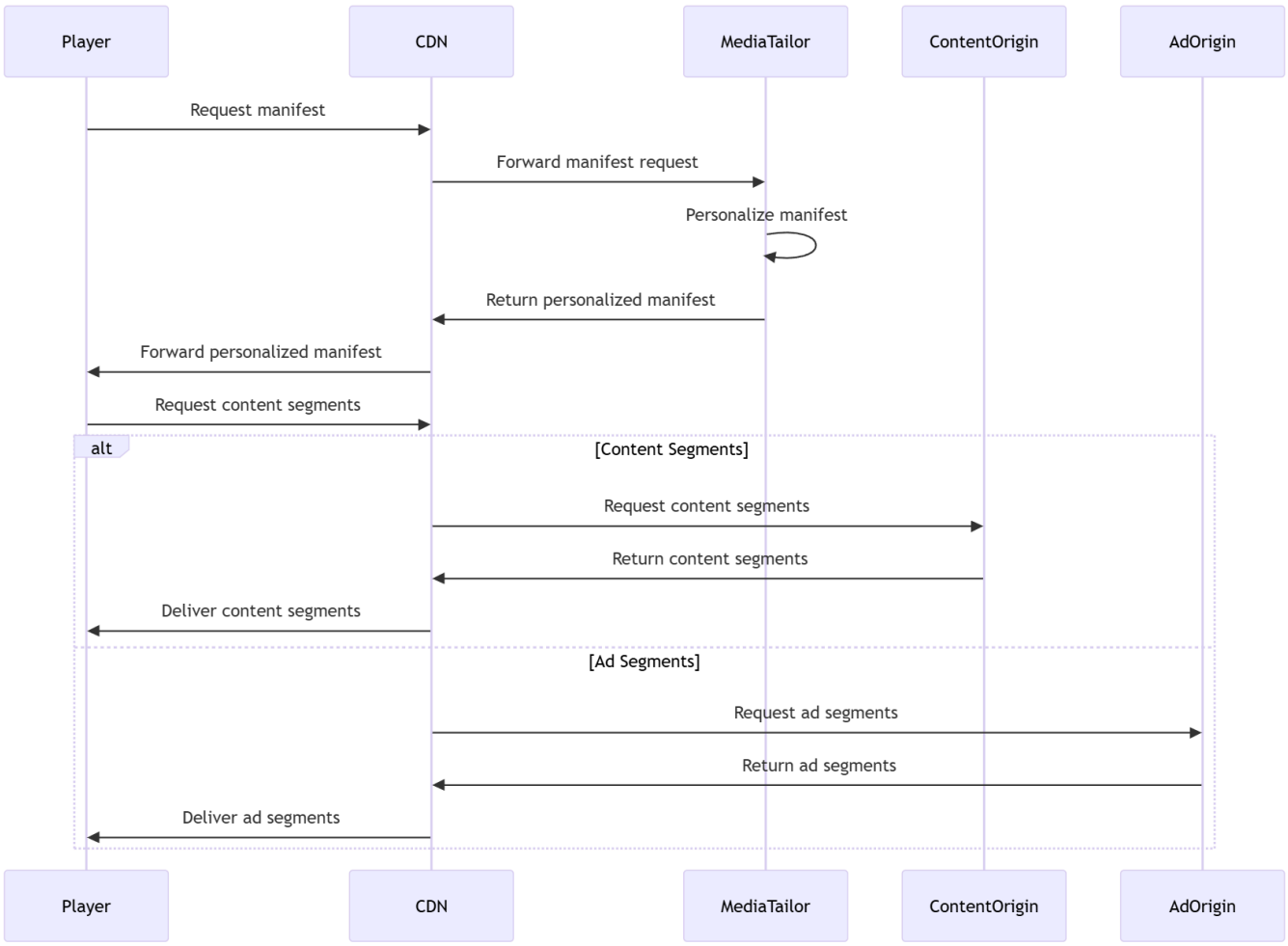
使用 CDN 进行广告插入的推荐架构将 CDN 置于观看者和广告插入之间，而广告插入则直接从您的来源访问内容。该架构为内容交付和视频盈利提供了以下好处：

- 有效缓存内容和广告细分
- 减少了请求负载 MediaTailor
- 提高了向观众传送的速度
- 简化的网址管理
- 跨设备持续投放个性化广告

在这个推荐的架构中：

1. 观众从 CDN 请求清单
2. CDN 将请求转发到广告插播中
3. 广告插入请求来自来源的内容清单
4. 广告插入请求来自广告决策服务器 (ADS) 的广告
5. 广告插入通过将广告标记（来自来源清单）替换为指向特定受众的定向广告细分（来自 ADS）的广告标记，从而对清单进行个性化设置 URLs
6. 广告插入会将包含广告细分的个性化清单返回 URLs 到 CDN，CDN 会将其转发给观众
7. 观众通过 CDN 请求区段
8. CDN 根据分段类型路由分段请求：
   - 内容分段请求将发送到内容来源
   - 广告细分请求转至 MediaTailor

这种架构可确保最佳性能，同时保持使用 CDN 的安全性和灵活性优势。



> **Note**
>
> VOD 和直播内容之间的流程略有不同。对于 VOD，清单可以缓存更长的时间，而直播内容需要更频繁地更新清单才能保持直播的连续性。

VOD 和直播内容缓存之间的主要区别：

VOD 内容

为清单设置更长的 TTL 值（从几分钟到几小时），因为它们不经常更改

直播内容

为清单设置较短的 TTL 值（秒），以确保观众收到最新的直播片段

我们不建议您在内容来源和 AWS Elemental MediaTailor之间放置 CDN。这样做可能会带来一些技术挑战：

缓存密钥冲突

配置您的 CDN 以正确处理查询参数。这样可以 MediaTailor 防止在请求具有不同查询参数的相同清单时收到不正确的清单。

Gzip 压缩问题

如果您遇到清单解析错误，请确保您的 CDN 向发送格式正确的清单。 MediaTailor有些 CDNs 可能会传送损坏的 gzip 负载，从而导致解析失败。如果发生这种情况，您可能需要禁用 CDN 之间的压缩， MediaTailor 同时保持压缩，以便在工作流程的其他地方节省成本。

表现出新鲜感

对于直播，请将您的 CDN 配置为将当前清单发送到。 MediaTailor这样可以防止内容和广告之间出现同步问题。

性能优化

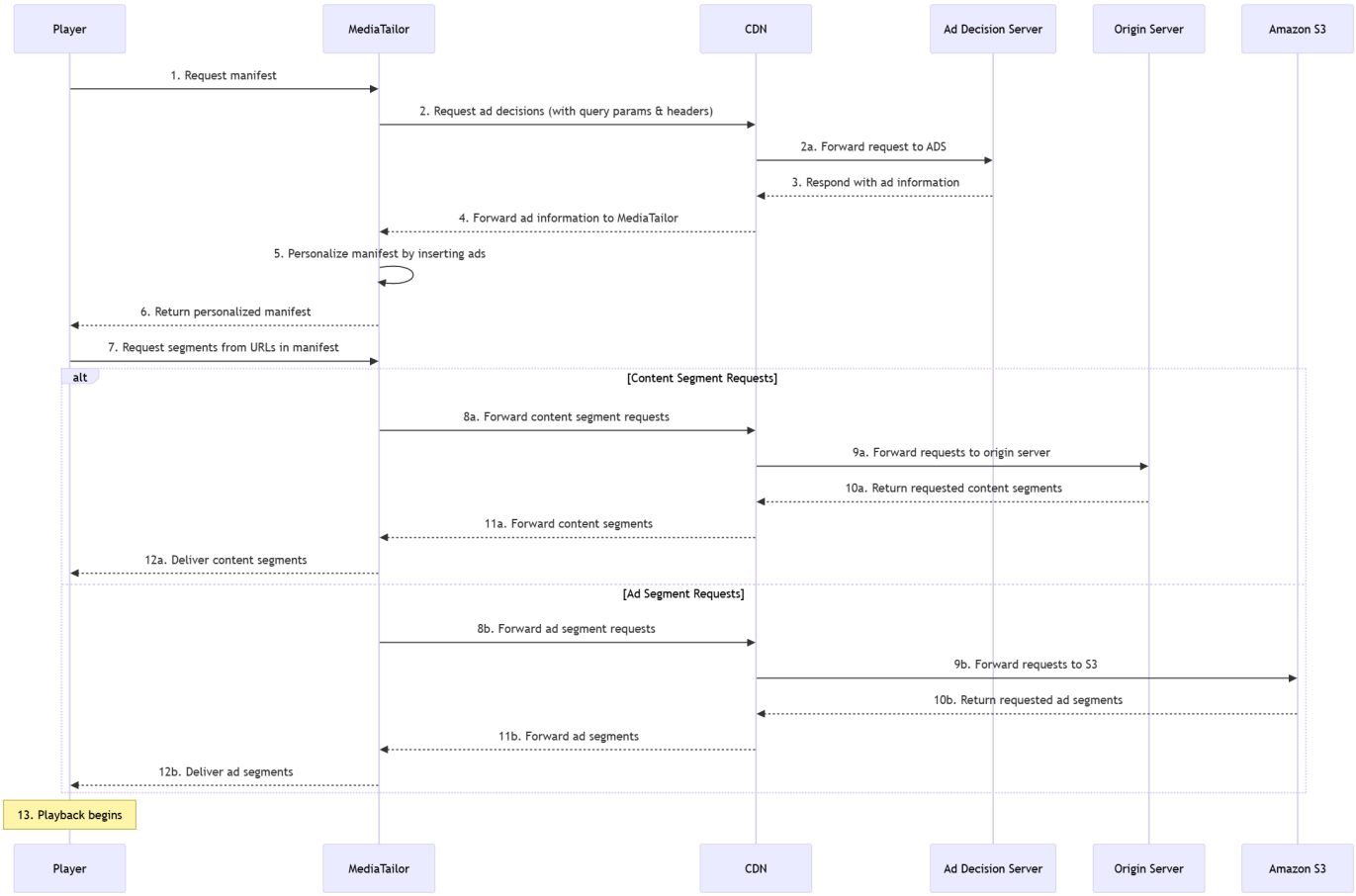
最大限度地减少网络跳跃和潜在的缓存丢失，以缩短播放启动时间。

缓存管理

实施简化的缓存失效策略，特别是对于清单经常更新的直播内容。

在这个次优架构中：

1. 观众请求多变体播放列表、媒体播放列表或 MPDs 直接从 AWS Elemental MediaTailor。
2. MediaTailor 通过 CDN 请求内容清单（多变播放列表、媒体播放列表或 MPDs）。
3. CDN 将请求转发到源服务器。
4. 源服务器返回多变体播放列表、媒体播放列表或 MPDs CDN。
5. CDN 会将多变量播放列表、媒体播放列表或转发到。 MPDs MediaTailor
6. MediaTailor 向广告决策服务器 (ADS) 请求广告。
7. MediaTailor 通过将广告插入多变体播放列表、媒体播放列表或将广告直接投放给观众，对清单 MPDs 进行个性化设置。
8. 这种架构会带来额外的延迟、潜在的缓存问题，并使故障排除变得复杂。



请求和响应流

使用 CDN 实现动态广告插入时，请将您的系统配置为支持此请求和响应流：

1. 将您的播放器配置为从您的 CDN 请求多变体播放列表 MPDs (HLS) 或 (DASH)，并将其 MediaTailor 作为清单来源。
2. 将 CDN 设置为将所有多变体播放列表、媒体播放列表和 MPD 请求转发到 MediaTailor，包括所有查询参数和标题。
3. Ensure MediaTailor 可以与您的广告决策服务器 (ADS) 通信，传递查询参数和标题。
4. 将您的广告配置为使用查询参数来确定要插入哪些广告。
5. 在 MediaTailor 播放配置中设置 CDN 前缀，这样 MediaTailor 就可以用 CDN 域名代替内容和广告区段网址前缀。

6. 配置您的 CDN 以将个性化的多变体播放列表、媒体播放列表以及 MPDs 从转发 MediaTailor 给请求的播放器。
7. 设置您的 CDN 以翻译区段 URLs，将内容分段请求转发到源服务器，将广告请求转发到存储转码广告的 Amazon S3 MediaTailor 存储桶。

用于广告插入的 CDN 术语

了解这些关键术语将有助于您实现广告插入 CDN 集成并对其进行故障排除：

原始 CDN 和边缘 CDN

Origin CDN：位于 MediaTailor 和您的内容来源之间的 CDN。它会缓存内容片段，以减少源服务器的负载。在多 CDN 架构中，这是第一个直接与源连接的 CDN 层。

边缘 CDN：位于观众和之间的 CDN。 MediaTailor它向观众提供个性化的清单和内容。在多 CDN 架构中，这是最外层的 CDN 层，直接与观众交互。

CDN 配置条款

缓存行为：确定 CDN 如何处理不同类型请求的规则。这些规则包括：

- 缓存持续时间设置
- 起点路由配置
- 请求处理参数

TTL（存活时间）：内容在需要从来源刷新之前在 CDN 缓存中保持有效的持续时间。

缓存密钥：CDN 用于存储和检索缓存内容的唯一标识符。它通常包括：

- 网址路径
- 查询参数
- 选定的标题

Origin shield：CDN 边缘站点和您的源服务器之间的中间缓存层。它减少了发往您的源站的请求数量。

请求折叠：一项 CDN 功能，可将对相同内容的多个同步请求合并到单个源请求中。

MediaTailor特定于 CDN 的术语

CDN 内容分段前缀：为清单中的内容分段生成时 AWS Elemental MediaTailor 使用 URLs 的 CDN 域名。

CDN 广告区段前缀：为清单中的广告区段生成时 MediaTailor 使用 URLs 的 CDN 域名。

有关使用配置 CDN 的更多信息 MediaTailor，请参阅设置 CDN 集成。

> **Note**
>
> 这些术语与频道组装文档中使用的术语一致。有关通道组装术语的信息，请参阅用于频道组装的 CDN 术语。

使用 CDN 设置基本 MediaTailor SSAI 以实现最佳广告投放

本节介绍 step-by-step如何使用内容分发网络 (CDN) 配置 AWS Elemental MediaTailor 动态广告插入，以优化视频获利工作流程。

有关使用动态变量的高级广告服务器配置，请参阅MediaTailor ADS 请求的动态广告变量。有关传递广告定位参数的信息，请参阅MediaTailor 清单查询参数。 CDNs

有关使用的 SSAI 的概念信息 CDNs，请参见了解 CDN 和 MediaTailor 集成的广告插入架构。

先决条件

在使用 CDN 设置广告插播之前，请确保：

- 活动 AWS Elemental MediaTailor 配置
- 提供带有适当广告标记的 HLS 或 DASH 内容的内容源服务器

  有关广告标记的信息，请参阅了解广告插入行为。
- 支持 VAST 或 VMAP 进行广告定位的广告决策服务器 (ADS)
- CDN 账户（例如 Amazon CloudFront 或其他 CDN 提供商）
- 清单操作和动态广告插入概念的基础知识

步骤 1：配置 CDN 缓存以实现最佳广告投放

正确的 CDN 缓存配置对于视频获利工作流程的最佳性能至关重要。服务器端广告插入 (SSAI) 和服务器端广告插入 (SGAI) 的缓存要求有所不同。使用以下推荐设置来确保高效交付内容和个性化广告：

SSAI CDN 缓存设置

对于服务器端广告插入工作流程，正确的缓存配置对于实现最佳性能至关重要。SSAI 需要特定的 TTL 值和缓存密钥设置，以确保在有效缓存区段的同时，不会缓存个性化清单。

有关详细的 SSAI 缓存设置，包括 TTL 值、路径模式和缓存密钥配置，请参阅 CDN 优化指南服务器端广告插入 (SSAI) 缓存中的。

SSAI 的关键缓存原则：

- 清单：将 TTL 设置为 0 秒以防止缓存个性化内容
- 分段：主动缓存（24 小时以上）以减少源站负载
- 缓存密钥：包括清单的所有查询参数，仅包含区段的 URL 路径

SGAI CDN 缓存设置

对于服务器引导的广告插入工作流程，缓存要求与 SSAI 不同，因为 SGAI 清单可以在短时间内缓存，同时仍能提供个性化的广告体验。

如需全面的 SGAI 缓存设置，包括 VOD 和实时 TTL 值，请参阅优化指南的缓存表。SGAI 的缓存效率比 SSAI 更高，同时还能保持广告个性化功能。

SGAI 缓存的主要区别：

- 清单：可以短时间缓存（VOD 缓存 5-30 分钟，直播缓存 2-10 秒）
- 区段：像 SSAI 一样积极缓存（大多数内容 24 小时以上）
- 性能优势：由于清单可缓存，因此缓存命中率比 SSAI 更好

对于 Amazon CloudFront，您可以使用具有不同 TTL 值和缓存密钥策略的缓存行为来实现这些设置。有关其他 CDNs，请参阅其实现类似缓存规则的特定文档。

第 2 步：实施混合方法（如果需要）

如果您的架构需要混合方法，在内容来源和 MediaTailor内容来源之间有一个单独的 CDN 或缓存层：

1. 在 CDN 配置中实现明确的关注点分离。
2. 配置特定的 CDN 设置以防止出现上一节中描述的技术问题。
3. 彻底测试您的配置，以验证清单个性化功能是否正常。

4. 监控性能指标，确保多变量播放列表、媒体播放列表和区段的最佳传输。 MPDs

在实施混合方法时，请考虑以下特定配置：

- 对于内容来源和 MediaTailor以下内容之间的 CDN：
  - 为清单文件配置压缩直通以保留源文件的原始压缩状态
  - 在缓存密钥中包含所有查询参数
  - 为直播内容清单设置短 TTL 值
- 对于与观众 MediaTailor 之间的 CDN：
  - 为广告区段配置更长的缓存时间
  - 为个性化清单设置 TTLs 合适的设置
  - 为内容与广告细分实施正确的来源路由

第 3 步：完成您的 CDN 设置

选择架构并了解请求流程后，请按照中的详细配置步骤完成设置设置 CDN 集成。

有关特定的 CDN 提供商，请参阅以下其他资源：

- Amazon CloudFront：CloudFront 集成有关 CloudFront特定配置步骤，请参阅
- 其他 CDNs：应用本指南中概述的一般原则，使其适应您的特定 CDN 的配置选项

步骤 4：验证您的配置

完成 CDN 设置后，请验证您的动态广告插入工作流程是否正常运行：

1. 使用示例播放器在 CDN 上测试播放
2. 确认在指定的广告断点正确插入了个性化广告
3. 检查 CDN 日志以确认请求路由正确
4. 监控缓存命中率，确保内容和广告细分的最佳性能
5. 确认广告定位参数已正确通过工作流程

有关全面的测试和验证程序，请参阅CDN 和集成的测试和 MediaTailor 验证。有关监控 SSAI 实施的详细信息，请参阅监控 CDN 和 MediaTailor 集成的运营。要优化性能，请参阅优化 CDN 和 MediaTailor 集成的性能。

将 MediaTailor SSAI 与渠道组合集成，实现货币化线性通道

本主题介绍如何将 AWS Elemental MediaTailor 服务器端广告插入与渠道组合和内容分发网络 (CDN) 集成相结合。此集成使您能够：

- 通过个性化广告创建盈利的线性渠道
- 向观看相同内容的不同观众投放定向广告
- 保持广播级质量的观看体验

将 SSAI 与渠道组装相结合的好处

将 SSAI 与渠道组合集成可带来以下几个主要好处：

线性渠道的盈利

将个性化广告插入您的线性渠道，从您的内容库中获得收入。您可以在单个线性直播中通过直播和 VOD 内容获利。

个性化广告

向观看相同频道内容的不同观众投放不同的广告。与传统的广播广告相比，这种有针对性的方法可以提高广告的相关性和潜在收入。

简化的广告插播时间管理

无需使用 SCTE-35 标记来调整内容，即可在频道组装节目中定义广告断点。这样可以更轻松地在内容的自然断点处插入广告。

广播质量的体验

MediaTailor 通过内容和广告之间的无缝切换，保持高质量的观看体验。服务器端广告插入消除了许多常见问题，例如：

- 广告过渡期间的缓冲
- 阻止获利的广告拦截器
- 播放质量不一致

可扩展的交付

当与 CDN 结合使用时，这种集成可以扩展到数百万并发观众，而不会降低性能或个性化功能。

架构概述

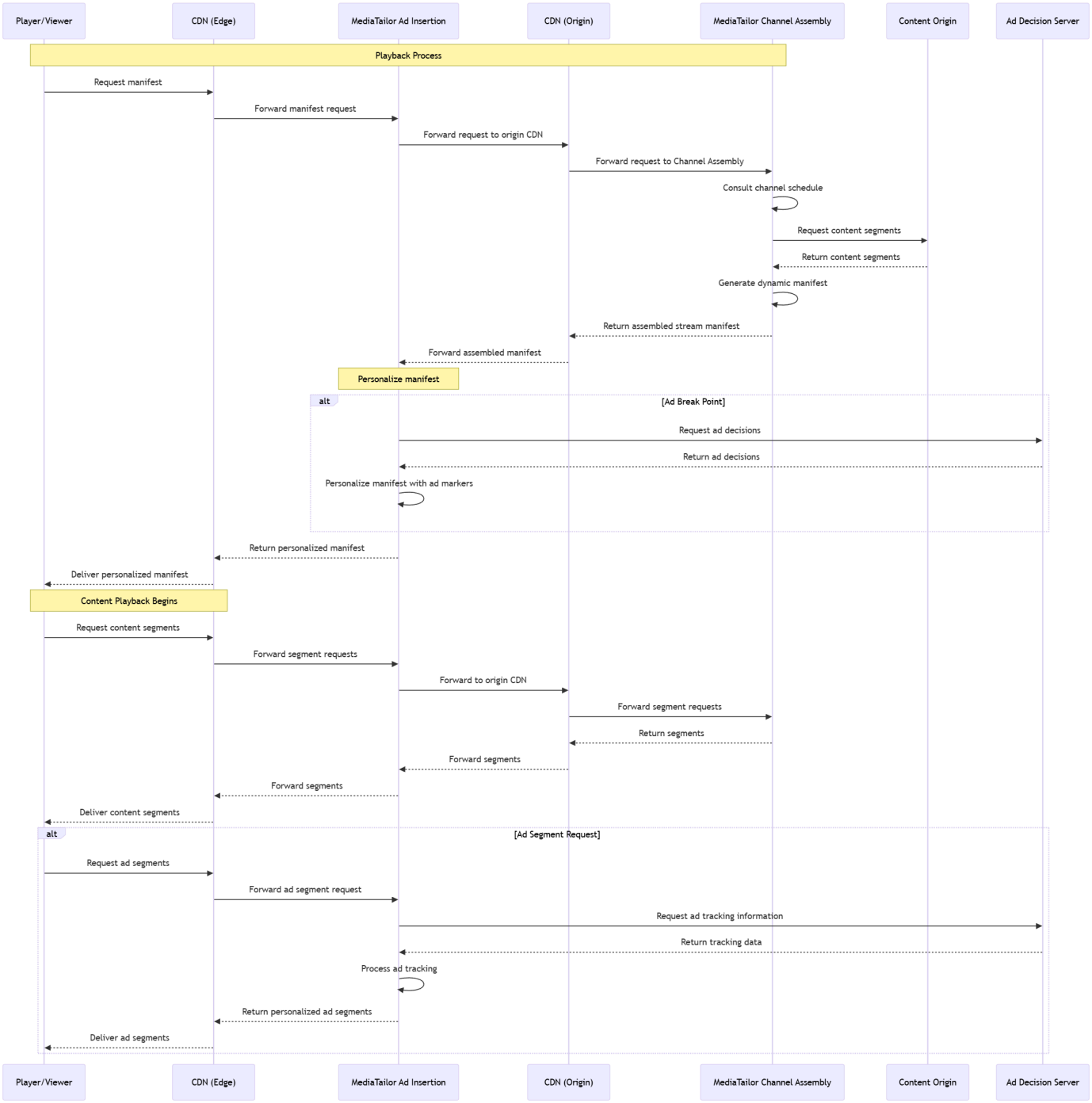
将 SSAI 与信道组装相结合的架构通常涉及以下组件：

- 频道集合：从 VOD 和直播内容创建线性频道，并在生成的清单中插入用于创建广告标记的 slate 内容
- 广告插入：识别广告中断点和 URLs 指向清单中个性化广告细分的插页
- 广告决策服务器 (ADS)：确定要为每位观众插入哪些广告
- 内容分发网络 (CDN)：向观众提供组装好的内容和广告片段
- 源服务器：存储 VOD 和直播内容片段

在此架构中：

1. 频道集合会根据你的 VOD 和直播内容创建线性频道，并在生成的清单中插入用于创建广告标记的 slate 内容
2. 当观众请求频道时，广告插入会识别插入到线性频道中的广告插播时间
3. 广告插入会调用 ADS 以接收广告列表，对其进行转码，然后将 URLs 指向转码后的广告区段插入到个性化清单中
4. CDN 向观众提供个性化直播

下图说明了此工作流程：



设置集成

要设置带有频道集合的 SSAI，请执行以下操作：

1. 将您的 Edge CDN 配置为接受观众的清单请求并将其转发到 AWS Elemental MediaTailor 广告插播中。
2. 设置 MediaTailor 广告插播以将请求转发到您的来源 CDN。
3. 配置您的源 CDN 以将请求转发给 MediaTailor 频道集合。
4. 设置 MediaTailor 渠道集合，根据当前计划生成动态清单。
5. 配置您的 Origin CDN 以将汇编好的清单转发到 MediaTailor 广告插播中。
6. 设置 MediaTailor 广告插入，以便在广告中断点向广告决策服务器请求广告决策。
7. 通过将 MediaTailor 广告标记（来自渠道组合）替换为 URLs 指向目标广告细分（来自 ADS），配置广告插播以个性化清单。
8. 设置您的边缘 CDN 以向观众提供个性化清单。
9. 配置您的 CDN 架构，以高效处理内容和广告细分请求。

在频道组装中定义广告插播时间

在频道集合中创建节目时，您可以通过多种方式定义广告插播时间：

程序过渡

在频道时间表中的节目之间插入广告。这是最简单的方法，可确保广告不会中断节目内容。

SCTE-35 记号笔

如果您的 VOD 内容包含 SCTE-35 标记，则频道组合可以保留这些标记，广告插入可以将其用作广告中断点。

基于时间的插入

在节目中的特定时间点定义广告时段。这允许您在内容的自然断点处插入广告。

有关创建带有广告插播时间的节目的详细信息，请参阅使用节目。

CDN 缓存注意事项

为了在将信道组合和 SSAI 与 CDN 结合使用时获得最佳性能：

- 配置区分信道集合和 SSAI 请求的缓存行为
- 按照中的建议为清单和区段设置相应的 TTL 值 步骤 1：配置 CDN 缓存以实现最佳广告投放

- 确保在渠道组装、广告插入和 CDN 来源之间进行正确的路由
- 监控渠道组装和广告插播组件的绩效指标

组合实现的推荐缓存设置

内容类型	TTL	缓存关键元素
渠道集结清单	0 秒	网址路径 + 查询参数
SSAI 个性化清单	0 秒	网址路径 + 所有查询参数
内容细分	24 小时以上	仅限网址路径
广告细分	24 小时以上	仅限网址路径

监控集成解决方案

为确保您的集成解决方案以最佳状态运行，请监控以下关键指标：

频道组装指标

监控清单生成时间、节目转换以及频道组装过程中的任何错误。

广告插入指标

跟踪广告填充率、广告决策服务器响应时间和广告插入错误。

CDN 指标

监控内容和广告细分的缓存命中率、原始请求量和响应延迟。

观众体验指标

跟踪重新缓冲事件、启动时间和观众参与度，尤其是在广告过渡期间。

有关监控的详细信息，请参阅监控 CDN 和 MediaTailor 集成的运营和监控 MediaTailor 频道集合 CDN 操作。

排查常见 问题

对集成解决方案的问题进行故障排除时，请考虑以下常见问题：

广告插播同步问题

如果广告未在预期的断点处显示，请确认您的频道集合计划中的广告插播定义配置正确，并且广告插入是否正确识别了这些断点。

清单配送错误

如果观众遇到播放问题，请检查 CDN 是否在频道集合和广告插入之间正确转发清单请求，以及缓存设置是否适合这些清单的动态特性。

分段路由问题

如果内容或广告区段未加载，请验证 CDN 是否正确地将区段请求路由到相应的来源，以及清单 URLs 中的区段格式是否正确。

性能下降

如果观看者遇到缓冲或高延迟，请检查 CDN 缓存命中率和原始请求量，以确定交付管道中的潜在瓶颈。

有关更多问题排查指导，请参阅使用 MediaTailor SSAI 进行故障排除， CDNs 实现不间断的广告投放。

最佳实践

要成功将 SSAI 与渠道集合集成，请遵循以下最佳实践：

- 全面测试：在部署到生产环境之前，使用各种内容类型、广告场景和观众条件对集成解决方案进行测试。
- 持续监控：设置全面的监控和警报，以快速识别和解决出现的任何问题。
- 优化缓存：根据实际使用模式和性能指标，定期检查和调整 CDN 缓存设置。
- 规模规划：设计架构以应对高峰流量负载，尤其是热门频道或活动。
- 考虑冗余：在关键组件中实现冗余，以确保线性通道的高可用性。
- 优化广告过渡：使用一致的编码配置文件和细分时长，确保内容和广告之间的平滑过渡。

相关信息

有关将 SSAI 与频道集成的更多信息，请参阅：

频道组装文档

使用 AWS Elemental MediaTailor 创建线性组装流-了解渠道组装概念

使用 CDN 进行频道组装-使用 CDN 设置频道组合

SSAI 文档

使用 CDN 进行广告插入-使用 CDN 设置广告插播

了解 CDN 和 MediaTailor 集成的广告插入架构-了解广告插入 CDN 架构

CDN 配置

设置 CDN 集成-一般 CDN 配置指南

CloudFront 集成- CloudFront 特定配置

优化 CDN 和 MediaTailor 集成的性能

通过优化内容分发网络 (CDN) 配置，最大限度地提高 AWS Elemental MediaTailor 广告插入实施的性能。这些设置可确保高效的内容交付和最佳的观看者体验。

有关特定于 SSAI 实现的详细缓存和路由优化指南，请参阅CDN 性能优化。综合优化指南提供了适用于所有 MediaTailor CDN 集成的全面缓存设置、请求路由配置和性能基准。

常见的性能挑战

使用的 SSAI 实现 CDNs 可能面临多项性能挑战：

清单操作开销

MediaTailor 执行实时清单操作，如果优化不当，可能会导致延迟。以下内容可能会引入延迟：

- 广告决策服务器 (ADS) 请求的处理时间
- 修改带有广告细分引用的清单所需的时间
- 用于个性化的额外处理

缓存效率问题

个性化清单会降低 CDN 缓存效率，因为：

- 每位观众都可能收到一份独一无二的清单
- 会话参数可能会使缓存碎片化
- 动态内容需要仔细配置缓存

Origin 负载峰值

缓存不当可能会导致源站在以下期间出现负载峰值：

- 高流量活动
- 缓存刷新
- CDN 配置变更

与广告相关的播放问题

广告插入可能会导致播放中断，例如：

- 广告过渡期间的缓冲
- 内容和广告之间的质量差异
- 无法检索广告时播放失败

有关全面的性能优化指南，包括缓存策略、请求路由、性能基准测试和高级优化技术，请参阅CDN 性能优化。综合优化指南提供了适用于所有 MediaTailor CDN 集成（包括 SSAI 实现）的详细设置和基准。

最佳实践摘要

遵循以下最佳实践，通过以下方式确保最佳 SSAI 性能： CDNs

架构最佳实践

- 根据您的规模和要求选择正确的架构模式
- 近距离部署服务以最大限度地减少延迟
- 为关键组件实现冗余

缓存最佳实践

- 对不同的内容类型使用不同的缓存策略
- 优化缓存密钥以平衡个性化和效率
- TTLs 根据内容类型和更新频率进行适当设置。有关详细的 TTL 建议，请参阅针对 CDN 和 MediaTailor 集成的缓存优化。

广告投放最佳实践

- 通过超时和后备功能优化 ADS 互动
- 准备符合内容规格的广告
- 实现高效的广告细分投放

监控最佳实践

- 监控 SSAI 实施的所有组件
- 设置性能下降警报
- 定期检查和优化您的配置

完整的优化指导

有关全面的 CDN 优化指南，包括详细的缓存策略、请求路由配置、性能基准测试和高级优化技术，请参阅CDN 性能优化。综合优化指南提供了适用于所有 MediaTailor CDN 集成（包括 SSAI 实现）的完整设置和基准。

监控 CDN 和 MediaTailor 集成的运营

AWS Elemental MediaTailor 提供强大的分析功能，这些功能与内容交付网络 (CDN) 指标相结合，可以全面了解您的 SSAI 实施情况。本主题涵盖：

有关全面的 CDN 监控指南，包括适用于所有 MediaTailor实现的基本指标、监控工具设置、警报配置和故障排除策略，请参阅内容分发网络监控。本主题重点介绍 SSAI 特定的监控要求和广告插入分析。

- SSAI 和 CDN 集成的监控策略
- 分析工具和数据收集方法
- 数据驱动的优化技术

SSAI 的关键指标包括 CDNs

要有效监控您的 SSAI 实施情况 CDNs，请跟踪以下基本指标：

广告插入指标

广告填充率：成功填充广告机会的百分比。

广告错误率：导致错误的广告请求的百分比。

广告响应时间：广告决策服务器响应广告请求所需的时间。

广告时长准确性：插入广告的实际时长与预期时长相匹配的程度。

观众体验指标

重新缓冲比率：缓冲所花费的观看时间的百分比。

启动时间：开始播放视频需要多长时间。

广告过渡流畅度：玩家在内容和广告之间过渡的无缝程度。

会话时长：观众在放弃直播之前观看了多长时间。

分析工具和集成

结合这些工具，为您的 SSAI 实施创建全面的分析解决方案：

AWS Elemental MediaTailor 服务器端指标

MediaTailor 通过 Amazon 提供内置指标 CloudWatch ，用于跟踪广告请求、响应和错误。这些指标可以在 CloudWatch 控制台中查看，也可以集成到自定义仪表板中。

关键 MediaTailor 指标包括：

- `AdDecisionServer.Ads`：广告决策服务器返回的广告数量。
- `AdDecisionServer.Duration`：广告决策服务器返回的广告总时长。
- `AdDecisionServer.Errors`：广告决策服务器返回的错误数。
- `AdDecisionServer.Latency`：广告决策服务器的响应时间。

有关 MediaTailor 指标的完整列表，请参阅 MediaTailor 使用 Amazon 进行监控 CloudWatch。

内容分发网络分析

CDN 提供商提供有关内容交付性能的详细分析。对于亚马逊 CloudFront，使用 CloudWatch 指标和亚马逊 CloudFront 访问日志来分析交付模式。

需要监控的重要 CDN 指标：

- 按内容类型划分的请求数（清单与区段）
- 不同内容类型的缓存命中率
- 观众的地理分布
- 按错误代码划分的错误率

客户端跟踪

实施客户端跟踪以收集服务器端不可用的观看者体验指标：

- 玩家事件（播放、暂停、寻找、缓冲）
- 广告浏览完成率
- 服务质量指标（分辨率变化、比特率）

- 观众参与模式

考虑使用 MediaTailor 客户端跟踪来收集和报告这些指标。

集成仪表板

创建综合仪表板，将来自多个来源的指标组合在一起：

- 使用 CloudWatch 仪表板合并 MediaTailor CloudFront 指标
- 考虑使用第三方分析平台实现更高级的可视化
- 设置跨服务关联以识别指标之间的关系

实施监测策略

按照以下步骤为您的 SSAI 实施全面的监控策略，实现 CDN：

1. 设置基本监控
   - 启用以下 CloudWatch 各项的指标 MediaTailor
   - 配置 CDN 日志记录和指标收集
   - 在视频播放器中实现客户端跟踪
2. 创建自定义控制面板
   - 构建结合关键指标的 CloudWatch 仪表板
   - 包括广告填充率、CDN 效果和观众体验的可视化效果
   - 为重要事件（配置更改、主要广播）添加注释
3. 配置警报
   - 为关键指标设置 CloudWatch 警报
   - 创建在多个相关条件下触发的复合警报
   - 配置通知渠道（电子邮件、短信、Amazon SNS）
4. 实现自动回复
   - 使用 CloudWatch 事件触发对常见问题的自动响应
   - 创建运行手册，以便在需要时进行手动干预
   - 记录不同警报场景的故障排除程序

Example创建全面的 SSAI 监控控制面板

此示例说明如何创建组合 MediaTailor 和 CloudFront指标的 CloudWatch 仪表板：

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

数据驱动的优化

使用您收集的分析数据来优化 SSAI 实施：

CDN 缓存优化

分析缓存命中率以确定改进机会：

- 根据内容类型和更新频率调整 TTL 设置。有关详细的 TTL 建议，请参阅[步骤 1：配置 CDN 缓存以实现最佳广告投放](#)。

- 优化缓存密钥设置以提高缓存效率
- 考虑为多层缓存实现原点屏蔽

广告投放优化

使用广告效果指标来改善广告投放：

- 识别并解决常见的广告插入错误
- 优化广告决策服务器响应时间
- 根据填充率分析调整广告定位参数

观众体验优化

根据客户端指标改善观众体验：

- 分析广告间歇期间的投放模式
- 优化广告过渡点以实现更流畅的播放
- 根据观众参与度数据调整广告频率和持续时间

成本优化

平衡性能和成本注意事项：

- 分析带宽使用模式以优化 CDN 成本
- 考虑对 CloudFront 分配进行价格等级调整
- 评估不同缓存策略的成本效益

最佳实践

请遵循以下最佳实践进行有效的 SSAI 监控和分析：

- 建立基准：在正常操作期间收集指标，以建立性能基准，以便在故障排除期间进行比较。
- 实施多级监控：在架构的不同级别（来源、CDN、播放器）进行监控，以全面了解性能。
- 关联服务间的指标：查找来自不同服务的指标之间的关系，以确定问题的根本原因。
- 使用异常检测：实现 CloudWatch 异常检测以自动识别指标中的异常模式。
- 定期审查和完善：安排定期审查您的监控策略，并根据不断变化的需求和新的见解进行调整。
- 记录调查结果和行动：保存优化工作及其结果的记录，以积累机构知识。

相关信息

有关使用监控和分析 SSAI 的更多信息 CDNs，请参阅：

- 优化 CDN 和 MediaTailor 集成的性能用于性能优化技术
- 使用 MediaTailor SSAI 进行故障排除， CDNs 实现不间断的广告投放用于对常见问题进行故障排除
- MediaTailor 与 Amazon CloudWatch 一起监控有关 MediaTailor指标的详细信息
- 查看 CloudFront 和边缘函数指标以获取有关 CloudFront 指标的信息

使用 MediaTailor SSAI 进行故障排除， CDNs 实现不间断的广告投放

本节提供了在内容分发网络 (CDN) 中使用 AWS Elemental MediaTailor 动态广告插入时常见问题的解决方案。这些解决方案将帮助您解决通过个性化广告实现视频盈利的问题。

有关全面的 CDN 故障排除指南，包括适用于所有 MediaTailor实现的缓存性能问题、HTTP 错误解决方案、测试程序和诊断技术，请参阅对 CDN 集成进行故障排除。本节重点介绍 SSAI 特定的故障排除要求和广告插入问题。

如果您的 CDN 和 SSAI 设置遇到问题，请检查以下常见问题：

个性化广告未出现在直播中

验证您的 ADS 是否正确响应并能与 AWS Elemental MediaTailor 之通信。检查以下潜在问题：

- 广告定位查询参数未通过您的 CDN 正确转发
- 您的内容中定义的广告断点不正确
- ADS 连接或响应问题

广告中断点出现播放错误

确保对广告细分进行正确转码，以匹配您的内容比特率和分辨率。查看以下常见问题：

- CDN 错误地路由了针对广告段的请求
- 在过渡点出现明显的操纵错误
- 内容和广告之间的编码配置文件不匹配

陈旧的清单

对于直播内容，请确认您的 CDN 缓存 TTL 设置是否合适。对于个性化清单，请使用 0 秒的 TTL。考虑为快速变化的清单实现缓存失效。有关全面的 TTL 指南，请参阅针对 CDN 和 MediaTailor 集成的缓存优化。

高延迟

检查您的 CDN 配置以获得最佳路由。确保您的 CDN 在观众附近有边缘位置，以获得最佳性能。

高级故障排除

对于更复杂的问题，请尝试以下高级故障排除技术：

广告定位行为不一致

检查玩家请求和 ADS 请求之间的查询参数是否不一致。确保所有必需的定位参数都正确通过 CDN 传递。

CDN 缓存不一致

验证缓存密钥配置以确保适当的内容区分。考虑为关键清单更新实现缓存清除。

广告跟踪失败

检查信标 URLs 是否正确转发且未被 CDN 阻止。验证客户端玩家是否可以到达跟踪端点。

性能优化

要优化动态广告插入和视频获利工作流程的表现，请执行以下操作：

- 根据内容类型和查看者模式微调 TTL 设置。有关详细的 TTL 建议，请参阅针对 CDN 和 MediaTailor 集成的缓存优化。
- 实施地理路由，最大限度地减少全球受众的延迟
- 考虑使用多重 CDNs 以获得冗余和最佳性能
- 监控缓存命中率并相应地调整配置
- 优化清单操作流程，缩短广告中断点的处理时间
- 对广告进行预转码以匹配常见的内容配置文件，实现无缝的个性化广告插入

性能基准

在优化 AWS Elemental MediaTailor 广告插入 CDN 集成时，请以以下绩效基准为目标：

缓存命中率目标

内容分段：缓存命中率大于 95%

广告细分：缓存命中率大于 90%

清单：不适用（不应缓存以进行个性化广告插入）

延迟基准测试

清单请求延迟：小于 100 毫秒 (P95)

内容分段交付：小于 50 毫秒 (P95)

广告细分投放：小于 75 毫秒 (P95)

End-to-end 启动时间：小于 2 秒

起源负载指标

每位观众的源请求数：每位观看者每分钟少于 0.1 个请求

每位观看者的源带宽：低于观看者总带宽的5%

错误率目标

明显错误：低于 0.1%

分段错误：小于 0.01%

玩家报告的缓冲次数：低于 1%

可扩展性基准

Support 在高峰事件期间支持正常流量的 10 倍而不会降级

每个通道每秒能够处理超过 1000 个请求

使用 Amazon CloudWatch 指标来跟踪这些绩效指标。有关详细的监控说明，请参阅监控 AWS Elemental MediaTailor 使用亚马逊 CloudWatch 指标。

相关信息

有关使用插入广告的更多信息 CDNs，请参阅：

广告插入文档

MediaTailor 广告插入入门-了解广告插入概念

设置-开始使用广告插入

内容分发网络集成

设置 CDN 集成-一般 CDN 配置指南

CloudFront 集成- CloudFront 特定配置

渠道组装集成

使用 CDN 进行频道组装-通过以下方式了解频道组装 CDNs

实现广告插入-通过频道集合实现广告插入

监控和优化

监控 CDN 和 MediaTailor 集成的运营-全面的监控和分析

优化 CDN 和 MediaTailor 集成的性能-性能优化指南

监控 AWS Elemental MediaTailor 使用亚马逊 CloudWatch 指标-的 CloudWatch 指标 MediaTailor

使用频道集合和 CDN 构建 MediaTailor 线性频道

本节提供了有关将 AWS Elemental MediaTailor 渠道组合与内容分发网络 (CDN) 集成的全面指导。请按照以下步骤设置、配置和优化您的频道组合 CDN 集成。

您还可以将渠道组装与服务器端广告插入 (SSAI) 相结合，通过个性化广告创建获利的线性渠道。这种强大的集成使您能够向观看相同频道内容的不同观众投放定向广告，从而在保持广播质量的观看体验的同时增加收入机会。有关使用的 SSAI 的信息 CDNs，请参阅使用 CDN 进行广告插入。

在本主题中，我们使用清单一词来统称多变体播放列表、媒体播放列表和。 MPDs

有关 MediaTailor 通道组装的更多信息，请参阅使用 AWS Elemental MediaTailor 创建线性组装流。

你需要什么

在使用 CDN 设置 MediaTailor 频道集合之前，请收集以下所需资源：

AWS 账户和权限

具有创建和管理 MediaTailor资源的适当权限的 AWS 账户

CloudFront （如果使用）和相关服务的 IAM 权限 MediaTailor

有关详细的权限要求，请参阅中的安全性 AWS Elemental MediaTailor。

必需的服务

运行 MediaTailor 频道汇编频道（不只是 SSAI 配置）

内容分发网络 (CDN) 账户（Amazon CloudFront 或第三方 CDN）

您的 VOD 内容的原始存储（Amazon S3 或其他源服务器） MediaPackage

内容要求

VOD 源已正确编码并以 HLS 或 DASH 格式打包。有关使用来源位置和 VOD 来源的信息，请参阅使用来源位置。

片段持续时间一致的内容（建议最短：1 秒）

广告插播时段的广告板内容（如果实现了广告插入）。有关配置 slate 的信息，请参阅MediaTailor slate 广告插入。

开始前的准备工作

> **Important**
>
> 此工作流程需要一个正在运行的 MediaTailor 渠道汇编通道。仅有 MediaTailor SSAI 配置不足以实现此集成。在继续进行 CDN 集成之前，您必须配置并运行活动信道汇编频道。

在使用 CDN 实现 MediaTailor 频道组合之前，请完成以下设置任务：

1. 配置 CDN 和源服务器之间的网络连接 MediaTailor
2. 设置 HTTPS 以实现安全的内容传输
3. 为您的 CDN 域名配置 DNS 设置

知识先决条件

要成功实施此解决方案，您应该：

- 了解流媒体协议 (HLS/DASH)
- CDN 配置原理的基础知识
- 熟悉 MediaTailor 频道组装概念

有关基本 MediaTailor 设置，请参阅设置和MediaTailor 频道汇编入门。有关使用来源位置和 VOD 来源的信息，请参阅使用来源位置。有关配置 slate 的信息，请参阅MediaTailor slate 广告插入。

CDN 集成的好处

将渠道组合与 CDN 集成可带来这些主要好处。

改善了观众体验

正确配置 CDNs 可减少线性通道的缓冲、启动时间和播放错误。这可以提高观众的参与度和满意度。

降低成本

高效的缓存策略可最大限度地减少源请求。这降低了数据传输成本和源服务器负载，这对于大容量线性通道尤其重要。

可扩展性

经过优化的 CDN 配置可在不降低性能的情况下处理热门活动期间的流量峰值，从而确保即使在观看高峰时段，您的线性频道也能保持可用。

全球影响力

正确配置 CDNs 向世界各地的观众提供低延迟的内容，无论他们身在何处，从而扩大您的潜在受众。

无缝节目过渡

经过优化的CDN配置可确保线性频道中节目之间的平稳过渡，从而创造广播质量的观看体验。

主题

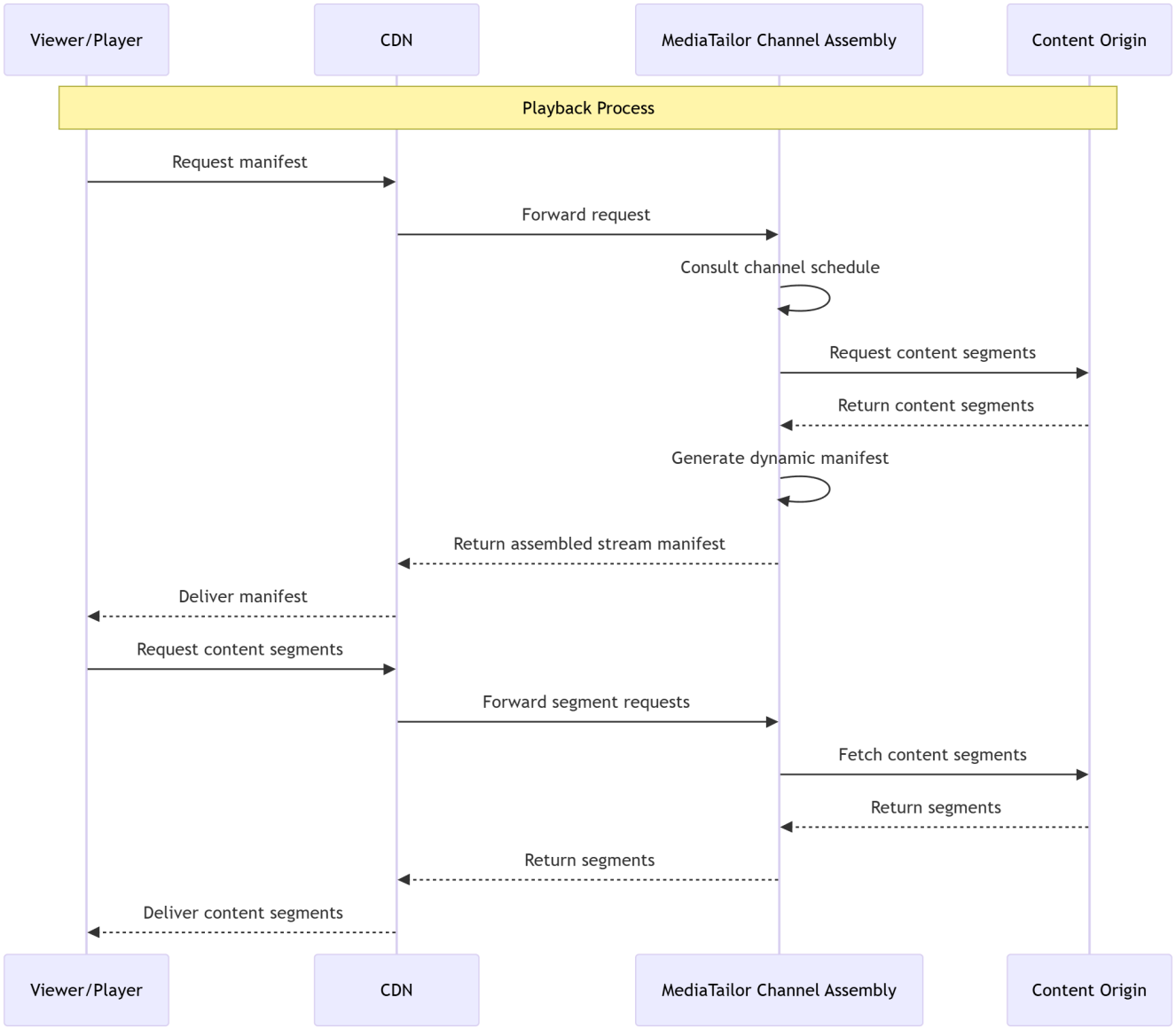
- 了解 MediaTailor 频道组装 CDN 架构
- 使用 CDN 设置基本 MediaTailor 频道组合
- URLs 为 MediaTailor 频道集合 CDN 配置基础
- 通过渠道集合实现 MediaTailor 广告插入
- 为 MediaTailor 频道集合配置时移视图
- 监控 MediaTailor 频道集合 CDN 操作
- 完成频道组装 CDN 优化

了解 MediaTailor 频道组装 CDN 架构

AWS Elemental MediaTailor 频道组合与内容交付网络 (CDNs) 集成，以提供具有更高性能和全球覆盖范围的线性流媒体频道。推荐的架构将 CDN 置于观众和频道集合之间，频道集合直接从您的来源访问内容。本主题说明了核心架构组件以及它们如何协同工作以交付您的内容。

1. 观众从 CDN 请求清单
2. CDN 将请求转发给频道集合
3. 频道集合汇编来自 VOD 来源的清单
4. 频道集会将清单返回到 CDN，CDN 将其转发给观众
5. 观众通过 CDN 请求区段
6. CDN 将分段请求路由到相应的来源

这种架构可确保最佳性能，同时保持使用 CDN 的安全性和灵活性优势。



用于频道组装的 CDN 术语

了解这些关键术语将有助于您实现渠道组装 CDN 集成并对其进行故障排除：

原始 CDN 和边缘 CDN

Origin CDN：位于 MediaTailor 和您的内容来源之间的 CDN。它会缓存内容片段，以减少源服务器的负载。在多 CDN 架构中，这是第一个直接与源连接的 CDN 层。

边缘 CDN：位于观众和之间的 CDN。 MediaTailor它向观众提供个性化的清单和内容。在多 CDN 架构中，这是最外层的 CDN 层，直接与观众交互。

CDN 配置条款

缓存行为：确定 CDN 如何处理不同类型的请求（包括缓存持续时间和源路由）的规则。

TTL（存活时间）：内容在需要从来源刷新之前在 CDN 缓存中保持有效的持续时间。有关详细的 TTL 建议，请参阅针对 CDN 和 MediaTailor 集成的缓存优化。

缓存密钥：CDN 用于存储和检索缓存内容的唯一标识符，通常包括 URL 路径、查询参数和标头。

Orig@@ in shield：CDN 边缘站点和源服务器之间的中间缓存层，可减少对源站的请求数量。

请求折叠：一项 CDN 功能，可将对相同内容的多个同步请求合并到单个源请求中。

MediaTailor特定于 CDN 的术语

CDN 内容分段前缀：为清单中的内容分段生成时 MediaTailor 使用 URLs 的 CDN 域名。

CDN 广告区段前缀：为清单中的广告区段生成时 MediaTailor 使用 URLs 的 CDN 域名。

有关使用配置 CDN 的更多信息 MediaTailor，请参阅设置 CDN 集成。

使用 CDN 设置基本 MediaTailor 频道组合

AWS Elemental MediaTailor 频道组合使您能够配置与内容分发网络 (CDN) 的基本集成，以便向观众高效地交付线性流媒体频道。请按照以下步骤设置频道集合和 CDN 之间的集成。

1. 将您的 CDN 配置为接受观众的清单并将其转发给 MediaTailor频道集合。
2. 设置 MediaTailor 频道集合以访问您的频道时间表并确定当前节目。
3. 配置 MediaTailor 频道集合，以根据计划向源服务器请求内容片段。
4. 确保您的内容来源可以将请求的区段交付给 MediaTailor 渠道组合。
5. 设置 MediaTailor 渠道集合，根据当前计划生成动态清单。
6. 配置您的 CDN，将组装好的多变体播放列表、媒体播放列表和观众分发 MPDs 给观众。
7. 使用适当的缓存设置，将 CDN 设置为处理来自观众的细分请求。
8. 配置您的 CDN 以将缓存失误转发给 MediaTailor 频道集合。
9. 设置 MediaTailor 频道集合，以便从您的内容来源检索请求的片段。
10. 配置您的 CDN 以将内容分段提供给观看者进行播放。

URLs 为 MediaTailor 频道集合 CDN 配置基础

AWS Elemental MediaTailor 频道组合需要适当的基本 URL 配置，以确保通过内容分发网络 (CDN) 的内容路由能够正常运行。在频道集合中配置基本 URL 设置，以便成功向观众交付内容。

内容分段 URL 配置

在您的频道集合频道配置中，将 “基本 URL” 设置为 CDN 域。这样可以确保汇编后的清单 URLs 中的所有区段都指向您的 CDN，而不是直接指向您的源服务器。

例如，如果您的源内容位于，`http://origin.example.com/content/`而您的 CDN 域名为`https://cdn.example.com/`，则将基本 URL 设置为。`https://cdn.example.com/content/`

访问限制配置

为了增强安全性，请将 CDN 配置为限制对源服务器的直接访问：

1. 在 CDN 中设置源站访问控制。
2. 将您的源服务器配置为仅接受来自 CDN 的请求。
3. 如有必要，可使用签名 URLs 或 Cookie 进行查看者身份验证。

对于亚马逊 CloudFront，您可以使用源站访问控制 (OAC) 来保护对您的来源的访问。有关保护 CloudFront 集成安全的更多信息，请参阅CloudFront 集成。

通过渠道集合实现 MediaTailor 广告插入

中的渠道组合与服务器端广告插入 (SSAI) 和内容交付网络 (CDNs) 无缝 AWS Elemental MediaTailor 集成，通过个性化广告创建获利的线性渠道。

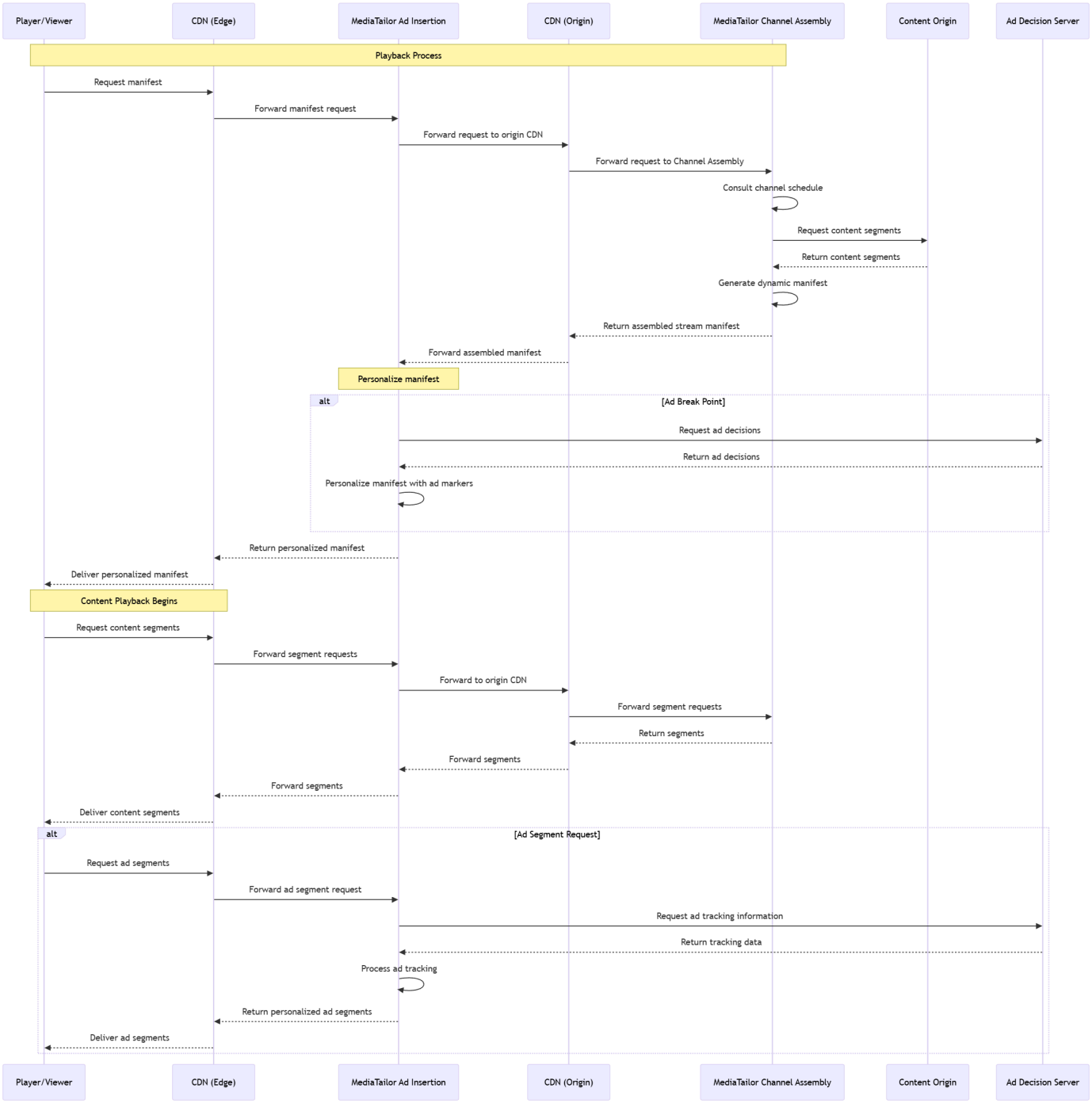
当您将频道组合与 SSAI 相结合时，您可以构建线性渠道，在保持广播质量体验的同时向观众投放个性化广告。此集成让您可以执行以下操作：

- 通过内容获利-通过线性渠道中的定向广告创造收入
- Personalize 体验——根据观看同一频道的观众的个人资料向他们投放不同的广告
- 保持质量-确保内容和广告之间的无缝切换，以实现广播级质量的观看
- 高效扩展-通过 CDN 交付支持数百万并发观众

有关使用的 SSAI 的详细信息 CDNs，请参阅使用 CDN 进行广告插入。

1. 将您的 Edge CDN 配置为接受观众的清单请求并将其转发到 MediaTailor 广告插播中。
2. 设置 MediaTailor 广告插播以将请求转发到您的来源 CDN。
3. 配置您的源 CDN 以将请求转发给 MediaTailor 频道集合。
4. 设置 MediaTailor 渠道集合，根据当前计划生成动态清单。
5. 配置您的 Origin CDN 以将汇编好的清单转发到 MediaTailor 广告插播中。
6. 设置 MediaTailor 广告插入，以便在广告中断点向广告决策服务器请求广告决策。
7. 配置 MediaTailor 广告插入，使用广告标记对清单进行个性化设置。
8. 设置您的边缘 CDN 以向观众提供个性化清单。
9. 配置您的 CDN 架构，以高效处理内容和广告细分请求。

下图说明了这种组合的工作流程：



为了在组合通道组装和 SSAI 时获得最佳性能：

- 配置区分信道集合和 SSAI 请求的缓存行为
- 按照中的建议为清单和区段设置相应的 TTL 值 [针对 CDN 和 MediaTailor 集成的缓存优化]

- 确保在渠道组装、广告插入和 CDN 来源之间进行正确的路由
- 监控渠道组装和广告插播组件的绩效指标

有关使用配置 SSAI 的详细信息 CDNs，请参阅：

- 了解 CDN 和 MediaTailor 集成的广告插入架构-了解 SSAI 架构和概念
- 使用 CDN 设置基本 MediaTailor SSAI 以实现最佳广告投放- Step-by-step SSAI 配置说明
- 使用 MediaTailor SSAI 进行故障排除， CDNs 实现不间断的广告投放-解决常见的 SSAI 集成问题

为 MediaTailor 频道集合配置时移视图

AWS Elemental MediaTailor 频道组合支持时移观看功能，这些功能可为线性通道启用类似 DVR 的功能，例如暂停、倒带和重新开始。通过将您的内容分发网络 (CDN) 配置为支持时移观看来启用这些功能，这样观众就可以控制自己的观看体验。

了解时移视图

时移观看可为线性通道提供类似 DVR 的功能，包括：

- 重新开始：观众可以从头开始观看节目，即使他们在节目开始后加入
- 暂停和恢复：观众可以暂停内容并稍后继续观看
- 快@@ 退和快进：观众可以在可用内容中向后和向前导航
- 延迟观看：观众可以观看该频道日程安排中较早播出的内容

时移观看的工作原理是向频道的播放 URL 添加一个`start`参数。该参数指定相对于当前时间的偏移量（以秒为单位）：

- 负值表示过去的时间（例如`start=-3600`，表示 “从 1 小时前开始”）
- 正值表示未来的某个时间（例如，`start=3600`表示 “从未来 1 小时开始”）

带有时移参数的网址示例：

```
https://example-cdn.com/out/v1/channel-name/index.m3u8?start=-3600
```

延时分辨率

要支持使用 CDN 进行时移观看，请执行以下操作：

1. 配置您的 CDN 以将`start`查询参数转发给频道集合。
2. 设置在缓存密钥中包含`start`参数的缓存行为。
3. 对于带有时移参数的清单，请使用短 TTL 或不使用缓存。

这样可以确保每位观众都能收到与其请求的时间位置相对应的正确清单。

CDN 对时移的要求

您的 CDN 必须满足以下要求才能通过频道集合支持时移观看：

- 将所有查询参数转发给频道组件。
- 在缓存密钥中包含该`start`参数。
- Support 支持对时移清单进行适当的缓存失效。
- 根据查询参数处理不同的清单响应。

监控 MediaTailor 频道集合 CDN 操作

AWS Elemental MediaTailor 与内容分发网络 (CDN) 集成后，渠道组合需要进行有效的监控，以确保可靠的内容交付。为您的渠道整合和 CDN 集成实施监控策略，以帮助确保可靠的内容交付和快速解决问题。

有关全面的 CDN 监控指南，包括适用于所有 MediaTailor实现的基本指标、监控工具设置、警报配置和故障排除策略，请参阅内容分发网络监控。本节重点介绍特定于频道组装的监控要求。

对您的渠道组合和 CDN 集成实施特定的监控：

- 在渠道汇编中跟踪清单生成指标。
- 监控时移查看请求及其对 CDN 缓存命中率的影响。
- 针对清单请求中的异常模式配置警报。
- 对内容来源中的区段可用性实施跟踪。

对于 Amazon CloudFront，创建一个将 CDN 指标与 MediaTailor 指标集成的控制面板，使用控制CloudWatch面板可视化您的整个交付渠道。

如果您还在渠道组合中使用 SSAI，请参阅，了解监控 CDN 的运行情况针对广告插入的其他监控建议。

完成频道组装 CDN 优化

有关特定于渠道汇编实现的详细 CDN 优化指南，包括缓存策略、请求路由配置、性能基准测试和高级优化技术，请参阅CDN 性能优化。

综合优化指南提供：

- 特定于频道集合的缓存设置以及推荐的 TTL 值
- 舱单和区段交付的请求路径优化
- 所有 MediaTailor 实施的性能基准和目标
- 高级优化技术，包括 Origin Shield 和压缩
- 渠道组装与 SSAI 的组合工作流程指南

MediaTailor 与 CDN MediaPackage 集成

AWS Elemental MediaTailor 与集成 AWS Elemental MediaPackage ，通过内容分发网络 (CDN) 投放个性化视频广告。 MediaPackage 是一项 just-in-time视频打包和制作服务，用于准备和保护您的视频内容，以便通过互联网传送。它可以将你的直播或点播视频内容打包成HLS和DASH等流媒体格式，为各种设备上的观众做好准备。

当你 MediaPackage 与 MediaTailor 和 CDN 结合使用时，你就创建了一个完整的流媒体工作流程，可以大规模投放个性化广告。CDN 在全球范围内分发您的内容，从而减少延迟并改善观众体验，同时在您的直播中 MediaTailor 插入有针对性的广告。

本主题重点介绍让 MediaTailor MediaPackage、和 CDN 协同工作的基本集成步骤。有关高级配置选项、故障排除和监控指南，请参阅后续步骤。

了解 MediaPackage 和 CDN 工作流程

在配置集成之前，请务必了解 MediaPackage MediaTailor、和 CDN 是如何协同工作的：

1. 内容准备： MediaPackage 接收您的直播或点播视频内容，并将其打包成流媒体格式（HLS 或 DASH 清单和片段）。
2. 广告插入： MediaTailor 向观众请求清单 MediaPackage，插入个性化广告，并将修改后的清单提供给观众。

3. 全球分发：您的 CDN 会缓存内容区段（来自 MediaPackage）和广告细分（来自），并将其分发给全球 MediaTailor观众。
4. 观看者播放：视频播放器通过 CDN 请求清单，CDN 在 MediaTailor（对于清单）和 MediaPackage（对于内容分段）之间适当地路由请求。

这种架构具有以下几个优点：

- 可扩展性：CDN 可在不影响源服务器的情况下处理高观看者负载
- 性能：内容从离观众最近的边缘位置传送
- 成本效益：通过缓存降低带宽成本
- 可靠性：多个边缘位置提供冗余

先决条件

在开始之前，请确保已配置以下组件：

1. MediaPackage 端点：用于接收和打包您的视频内容的已配置 MediaPackage 端点。有关设置说明，请参阅 MediaPackage 用户指南 MediaPackage中的入门。
2. MediaTailor 配置：使用您的 MediaPackage 终端节点作为内容来源的 MediaTailor 配置。有关设置说明，请参阅集成内容源以进行 MediaTailor 广告插入。
3. CDN 分发：配置为使用流媒体的 CDN 分发（例如 CloudFront）。有关设置说明，请参阅 CloudFront 开发者指南中的创建发行版。
4. 广告决策服务器：已配置的广告决策服务器，可返回 VAST 或 VMAP 响应以进行广告插入。

步骤 1：配置基本的 CDN 设置

正确的 CDN 配置对于成功 MediaPackage 集成至关重要。设置不正确可能会导致播放失败、缓存性能不佳和成本增加。如果没有正确的缓存策略和查询参数转发，您的 CDN 可能无法正确交付清单，或者可能完全绕过缓存，从而导致源服务器负载过高和观看者体验降低。

配置基本缓存设置

配置基本缓存至关重要，因为 MediaPackage 使用特定的缓存控制标头来优化内容交付。如果没有适当的缓存设置，您的 CDN 可能会忽略这些标头，从而导致不必要的源站请求和延迟增加。请按照以下步骤确保最佳的缓存行为：

要配置与之配合使用的基本缓存，请执行 MediaPackage以下操作：

1. 在 CloudFront 控制台中打开您的 CloudFront 分发设置。
2. 为您的 MediaPackage 源选择或创建缓存策略。
3. 启用 “Origin Cache-Control 标头” 选项。
4. MediaPackage 允许通过其缓存控制标头来控制缓存行为。

此基本配置允许 MediaPackage 自动为不同的内容类型设置适当的缓存持续时间。要使用特定 TTL 值和性能调整实现高级缓存优化，请先完成此基本设置，然后继续。优化 CDN 缓存 MediaTailor 和 MediaPackage 内容交付

配置基本查询参数

查询参数配置对于 MediaPackage 功能至关重要。您的 CDN 必须转发特定的查询参数才能启用时移观看和低延迟直播等功能。不正确的查询参数设置可能会使这些功能无法运行，我担心会降低缓存效率。按照以下步骤配置查询参数转发：

为确保您的 CDN 将所需的查询参数转发至： MediaPackage

1. 在 CloudFront 分发设置中，为清单请求选择或创建缓存策略。
2. 在 “缓存密钥设置” 下，选择 “包括指定的查询字符串”。
3. 添加以下基本查询参数：
   - `start`和 `end`-用于时移查看功能。将这些参数传递 MediaPackage 到以定义特定的内容窗口，以便从头到尾查看和追赶查看
   - `_HLS_msn`和 `_HLS_part`-用于支持 LL-HLS 播放请求
   - `m`-用于捕获端点 ti/compame 的修改内容。 MediaPackage 响应始终包含用于捕获端点修改时间的`?m=###`标签。如果已经使用此标签的不同值缓存了内容，则会 CloudFront 请求新的清单，而不是提供缓存的版本
   - `aws.manifestfilter`-用于清单筛选功能。如果您使用的是清单筛选，则必须包含此参数才能将分配配置为将`aws.manifestfilter`查询字符串转发到 MediaPackage源，这是清单筛选功能起作用所必需的
4. 仅包括 MediaPackage 使用的查询字符串。包含不必要的查询字符串会为相同的内容创建多个缓存变体，从而降低缓存效率。

这些参数使您的 CDN 具有基本 MediaPackage 功能。如果您需要针对不同的订阅等级或设备类型实现内容筛选，请先完成此基本查询参数设置，然后继续使用 MediaTailor、 MediaPackage和 CDN 设置清单筛选。

有关如何 MediaTailor 将查询参数传递给`start`和`end`传递到 MediaPackage 时移查看的信息，请参阅MediaTailor 原点的查询参数处理中的。MediaTailor 清单查询参数

配置 LL-HLS 的响应超时

超时配置对于低延迟 HLS 至关重要，因为 LL-HLS 使用 “阻止请求” 机制，CDN 会等待新的内容分段。如果超时时间过短，请求就会失败，然后 MediaPackage 才能用新的片段进行响应，从而导致播放中断和观众体验不佳。配置适当的超时时间以确保流畅播放 LL-HLS：

如果您使用的是低延迟 HLS，请配置 CDN 超时设置：

1. 在您的 CDN 设置中，找到源站超时配置。
2. 将响应超时值设置为零件持续时间的至少三倍。
3. 例如，如果您的零件持续时间为 0.3 秒，请将超时设置为至少 0.9 秒。

这样可以确保 CDN 在使用阻止请求机制时等待足够长的时间 MediaPackage 才能做出响应。

第 2 步：验证您的集成

测试您的集成对于确保所有组件在观众遇到问题之前都能正常协同工作至关重要。集成失败可能会导致播放中断、广告丢失或效果不佳。此验证过程可帮助您在受控环境中识别和解决问题。

配置 CDN 设置后，通过测试从内容请求到广告插入的完整工作流程，验证您的集成是否正常运行。

步骤 2.1：测试基本播放

基本播放测试可验证您的 CDN 是否正确处理清单请求并将其转发给。 MediaTailor此测试有助于识别缓存策略、查询参数转发和清单处理方面的配置问题。请按照以下步骤测试基本清单交付：

通过您的 CDN 请求清单，测试您的基本集成是否正常运行：

1. 使用网络浏览器或 curl 通过您的 CDN 请求清单网址。
2. 验证清单是否成功加载并且包含内容和广告细分。
3. 检查清单 URLs 中的内容区段是否指向你的 CDN 域。

4. 确认广告区段 URLs 也指向您的 CDN 域名。

如果清单正确加载并包含预期内容 URLs，则说明您的基本集成正在运行。有关全面的测试方法和高级验证程序，请参阅CDN 和集成的测试和 MediaTailor 验证。要设置对集成的性能和运行状况的全面监控，请参阅监控 CDN 和 MediaTailor 集成的性能 MediaPackage。

步骤 2.2：测试视频播放

视频播放测试可确保您的完整集成正常运行 end-to-end，包括广告插入和通过 CDN 交付内容。此测试可验证内容细分和广告区段是否已正确缓存和投放，以及观看者体验是否符合您的质量标准。请按照以下步骤测试完整的播放功能：

测试插入广告后视频播放是否能正常播放：

1. 使用视频播放器（例如 Video.js 或 HLS.js）通过 CDN 播放您的内容。
2. 确认视频播放流畅无缓冲问题。
3. 确认广告在播放期间按预期的时间插入。
4. 检查内容和广告区段是否都从您的 CDN 加载（而不是直接来自来源）。

如果广告可以顺畅播放，则说明您的集成运行正常。有关全面的测试方法和高级验证程序，请参阅CDN 和集成的测试和 MediaTailor 验证。如果您遇到任何播放问题、缓冲或广告插入问题，请参阅故障排除 MediaPackage、CDN 和集成 MediaTailor。

后续步骤

完成基本集成后，您可以实现高级功能和优化：

高级 CDN 优化

有关缓存优化、TTL 配置和性能调整的详细信息，请参阅优化 CDN 缓存 MediaTailor 和 MediaPackage 内容交付。

清单筛选

要对分层服务、设备优化或访问控制实施内容筛选，请参阅使用 MediaTailor、 MediaPackage和 CDN 设置清单筛选。

问题排查

如果您在集成时遇到问题，请参阅故障排除 MediaPackage、CDN 和集成 MediaTailor。

性能监控

要设置全面监控并了解关键性能指标，请参阅监控 CDN 和 MediaTailor 集成的性能 MediaPackage。

使用 MediaTailor、 MediaPackage和 CDN 设置清单筛选

AWS Elemental MediaTailor 使用清单筛选功能来自定义通过内容分发网络 (CDN) 交付给不同观众的清单中包含哪些音频和视频流。 AWS Elemental MediaPackage 这对于实施分层服务产品、特定于设备的优化或内容访问控制特别有用。

本主题专门介绍实现清单筛选功能。在实施清单筛选之前，您必须完成基本的内容分发网络集成设置。如果您尚未设置基本 MediaPackage 和内容分发网络集成，请从开始MediaTailor 与 CDN MediaPackage 集成。

清单筛选功能

在实现清单筛选之前，请先了解使用此功能可以完成什么：

核心过滤功能

清单筛选提供了几项关键功能，可帮助您控制内容交付：

- 限制观看者访问优质内容（例如 4K HEVC）
- 使用适当的直播来定位特定的设备类型
- 根据音频采样率、语言或视频编解码器筛选内容
- 为不同的订阅者提供不同的质量等级

常见使用案例

这些用例演示了清单筛选如何满足特定的业务需求：

订阅套餐

为基本订阅者提供分辨率较低的直播，同时为高级订阅者提供4K内容的访问权限

示例：基本等级限制为 720p，高级等级最高为 4K

设备优化

根据设备功能自动提供适当的直播

示例：移动设备获得更低的比特率，智能设备 TVs 获得更高的质量

带宽管理

在使用高峰期限制直播质量以管理网络成本

示例：在高流量事件期间降低最大比特率

区域内容

根据观众的位置提供不同的音频语言或内容变体

示例：自动筛选本地语言音轨

有关清单筛选概念的更多信息，请参阅 AWS Elemental MediaPackage 用户指南中的清单筛选。

配置您的 CDN 以进行清单筛选

清单筛选的 CDN 配置是必不可少的，因为您的 CDN 必须将`aws.manifestfilter`查询参数转发到才能 MediaPackage 进行筛选。如果没有正确的查询字符串转发，CDN 将删除过滤器参数，所有查看者都将收到未经过滤的清单，无论其订阅等级或设备功能如何。此配置可确保您的过滤逻辑达到预期 MediaPackage 并按预期运行。

要通过 CDN 启用清单筛选，您需要配置查询字符串转发：

1. 在您的 CloudFront 分配中，创建或编辑清单请求的缓存行为。
2. 对于缓存策略，请创建新策略或编辑现有策略。
3. 在 “缓存密钥设置” 下，选择 “包括指定的查询字符串”。
4. `aws.manifestfilter`添加到允许的查询字符串列表中。
5. 如果您还使用其他 MediaPackage 功能，请添加其查询参数：
   - `start`和 `end`-用于时移观看
   - `time_delay`-用于延时功能
   - `_HLS_msn`还有 `_HLS_part`-对于 LL-HLS

有关创建分配的更多信息，请参阅 Amazon CloudFront 开发者指南中的创建分配。

实现客户端筛选

在客户端实现中，您可以定义视频播放器和应用程序如何请求经过筛选的内容。此配置根据每位观看者的订阅级别、设备功能或其他标准来确定他们接收的内容。适当的实施可确保观看者只收到他们本应有权访问的内容，同时保持最佳的 CDN 缓存效率。

要在视频播放器和应用程序中实现清单过滤，请执行以下操作：

筛选的工作原理

筛选过程的工作原理如下：

1. 您的视频播放器或应用程序请求包含过滤器参数的清单 URL
2. CDN 将请求（包括查询参数）转发到 MediaTailor
3. MediaTailor 在请求来源清单 MediaPackage 时将过滤器参数传递给
4. MediaPackage 应用筛选条件并返回仅包含符合您条件的变体的自定义清单
5. MediaTailor 处理过滤后的清单以进行广告插入，然后将其返回给玩家

用于过滤的 URL 格式

了解正确的 URL 格式对于成功实施过滤至关重要。不正确的 URL 格式会导致过滤参数被忽略或导致 HTTP 错误。URL 结构必须包含过滤器参数作为查询字符串，您的 CDN 会将其转发到该字符串。MediaPackage请按照以下步骤实现正确的 URL 格式：

要在视频播放器中实现清单过滤，请执行以下操作：

1. 修改玩家的清单请求 URLs 以包含相应的过滤器参数。
2. 在查询参数中使用以下 URL 格式：

```
https://CloudFront-Domain/v1/master/MediaTailor-Config/index.m3u8?
aws.manifestfilter=video_codec:h264;audio_language:en-US
```

3. 当您的玩家请求此网址时， MediaTailor 会将这些参数传递给 MediaPackage，从而生成经过过滤的清单。

常见的过滤场景

使用以下示例来实现常见的过滤场景：

特定于设备的内容交付

要根据设备功能进行筛选，请将此参数添加到清单请求中：

```
aws.manifestfilter=video_codec:h264;audio_sample_rate:0-44100
```

此示例将内容限制为采样率高达 44.1 kHz 的 H.264 视频和音频，适用于移动设备。

付费内容限制

要限制对高比特率内容的访问，请在清单请求中添加以下参数：

```
aws.manifestfilter=video_bitrate:0-9000000
```

此示例将视频比特率限制在 9 Mbps 或更低，适用于基本订阅级别。

语言选择

要筛选特定的音频语言，请将此参数添加到清单请求中：

```
aws.manifestfilter=audio_language:fr,en-US,de
```

此示例仅包括法语、美国英语和德语音轨。

分辨率定位

要筛选特定的视频分辨率，请将此参数添加到您的清单请求中：

```
aws.manifestfilter=video_height:240-360,720-1080
```

此示例包括高度介于 240-360 像素和 720-1080 像素之间的视频流，不包括中档分辨率。

基于编解码器的过滤

要筛选特定的视频编解码器，请将此参数添加到清单请求中：

```
aws.manifestfilter=video_codec:h264,h265
```

此示例仅包含 H.264 和 H.265 视频流，不包括其他编解码器。

特殊注意事项和限制

要避免在实现清单筛选时出现常见问题，请执行以下操作：

技术限制

- 对于 TS 清单，请使用音频再现组以避免移除与过滤掉的音频流多路复用的视频流
- 在 TS 和 CMAF 清单中，音频采样率和视频比特率不容易在清单中显示以供验证
- 附加到媒体播放列表或片段的请求参数将导致 HTTP 400 错误

错误条件

- 如果筛选结果清单为空（没有符合筛选条件的流），则 MediaPackage 将返回 HTTP 400 错误
- 过滤器配置（端点过滤器 + 查询参数）冲突会导致 HTTP 404 错误
- 过滤器语法无效或不支持的过滤器类型会导致 HTTP 400 错误

性能注意事项

- 每个独特的过滤器组合都会创建一个单独的缓存条目，这可能会降低缓存效率
- 具有许多标准的复杂筛选器可能会影响清单生成性能
- 考虑在静态过滤场景中使用端点级过滤器来提高缓存性能

测试您的过滤实现

测试清单筛选实现对于确保查看者根据访问级别和设备功能接收正确的内容至关重要。过滤失败可能会导致观看者收到不正确的质量等级、不支持的格式或他们不应该访问的内容。全面的测试有助于在这些问题影响您的观众之前识别和解决这些问题。

要验证您的清单筛选是否正常运行，请执行以下操作：

1. 请求具有不同筛选参数的清单并验证结果
2. 检查筛选后的清单是否仅包含预期的直播流
3. 测试边缘案例（空结果、过滤器无效）以确保正确处理错误
4. 验证您的 CDN 是否正确转发过滤器参数
5. 使用不同的设备和播放器进行测试以确保兼容性

有关筛选问题的疑难解答，请参阅排除 MediaPackage CDN 集成问题。

如果您遇到 HTTP 400 错误、空清单或过滤参数无法按预期工作，请参阅故障排除 MediaPackage、CDN 和集成 MediaTailor以获取具体的清单筛选疑难解答指南。

优化 CDN 缓存 MediaTailor 和 MediaPackage 内容交付

AWS Elemental MediaTailor 当 AWS Elemental MediaPackage 用作内容来源时，需要适当的内容分发网络 (CDN) 缓存配置才能获得最佳性能。 MediaPackage 提供特定的缓存控制标头，告诉您的内容分发网络将不同类型的内容缓存多长时间。遵循这些建议可以确保流畅的播放和高效的内容交付。

本主题专门介绍如何优化缓存行为以最大限度地提高性能和降低成本。在实施高级缓存优化之前，请确保您已完成基本的内容交付网络集成设置。如果您尚未设置基本集成，请从开始MediaTailor 与 CDN MediaPackage 集成。

MediaPackage 缓存控制标头

MediaPackage 为不同的内容类型设置特定的 TTL 值以优化缓存行为：

多变体播放列表（HLS 和 LL-HLS）

TTL：媒体片段时长的一半

原因：这些播放列表会随着新片段的推出而变化，因此需要经常更新

媒体播放列表（常规 HLS）

TTL：媒体片段时长的一半

原因：与多变体播放列表类似，这些播放列表会随着内容的进展而更新

媒体播放列表 (LL-HLS)

TTL：1 秒

原因：低延迟直播需要非常频繁的更新

TS 媒体片段和初始化片段

TTL：1209600 秒（14 天）

原因：媒体区段创建后不会更改，因此可以长时间缓存

CMAF 媒体片段和初始化片段

TTL：1209600 秒（14 天）

原因：就像 TS 区段一样，这些区段一旦创建就不可变

有关所有 MediaTailor 工作流程的全面 TTL 建议以及其他缓存优化策略，请参阅针对 CDN 和 MediaTailor 集成的缓存优化。

配置 CDN 缓存策略

正确的缓存策略配置对于实现最佳性能和成本效益至关重要。不同类型的内容（清单、区段、初始化文件）有不同的缓存要求。使用单独的缓存行为可以优化每种内容类型的缓存，从而提高缓存命中率并减少源负载。如果没有适当的缓存策略，您可能会遇到不必要的原始请求、成本增加和播放性能不佳的情况。

要正确使用缓存 MediaPackage 控制标头并优化缓存，请执行以下操作：

1. 在 CloudFront 控制台中打开您的 CloudFront 分发设置。
2. 为不同的内容类型创建单独的缓存行为：
   - 清单请求（*.m3u8、*.mpd）
   - 媒体片段（*.ts、*.mp4、*.m4s）
   - 初始化片段
3. 对于每种缓存行为，请使用以下设置创建或选择缓存策略：
   - 启用 “Origin 缓存控制标头” 选项
   - 设置 “原始请求策略” 以转发必要的标头
   - 根据内容类型配置查询字符串转发

清单缓存行为

对于清单请求（*.m3u8、*.mpd）：

- 路径模式：*.m3u8 和 *.mpd
- 缓存策略：Honor Origin 缓存控制标头
- 查询字符串：转发特定参数（请参阅优化查询字符串的转发）
- 标题：转发所有标题（有关最低要求，请参阅MediaTailor CDN 集成所需的标头）

媒体片段缓存行为

对于媒体片段（*.ts、*.mp4、*.m4s）：

- 路径模式：*.ts、*.mp4、*.m4s
- 缓存策略：Honor Origin 缓存控制标头（14 天 TTL）
- 查询字符串：无（区段不使用查询参数）
- 压缩：启用压缩可提高交付性能

优化查询字符串的转发

查询字符串优化对于缓存效率至关重要，因为不必要的查询参数会为相同的内容创建多个缓存变体。每个唯一的查询参数组合都会创建一个单独的缓存条目，从而降低缓存命中率并增加源请求。通过仅转发 MediaPackage 实际使用的查询字符串，可以在保持全部功能的同时最大限度地提高缓存效率。

将 CDN 配置为仅转发 MediaPackage 使用的查询字符串，从而提高缓存效率：

基本查询字符串

- `start`和 `end`-用于时移查看窗口
- `time_delay`-用于对清单内容应用延迟
- `_HLS_msn`、`_HLS_m`、和 `_HLS_part`-适用于 LL-HLS 播放请求

特定功能的查询字符串

- `aws.manifestfilter`-用于清单筛选

> **Important**
>
> 请勿在缓存密钥中包含任何其他查询字符串。 MediaPackage 忽略无法识别的参数，包括这些参数会产生不必要的缓存变体，从而降低缓存效率。

性能优化技术

这些优化是在您的 CDN（例如 CloudFront）上配置的，而不是在或中 MediaPackage 配置的。MediaTailor实施以下额外优化以最大限度地提高缓存性能：

起源盾牌

Origin shield 在您的 CDN 边缘站点和 MediaPackage 端点之间提供了额外的缓存层。这样可以减少到达 MediaPackage终端节点的请求数量，从而提高性能并降低成本，尤其是在流量高峰期间或缓存命中率低于最佳状态时。Origin shield 对于直播特别有用，因为在直播中，多个边缘站点可能会同时请求相同的内容。

启用 Origin shield 以减少 MediaPackage 终端的负载：

1. 在你的 CloudFront 分发中，为你的来源启用 Orig MediaPackage in Shield。
2. 选择靠近您的 MediaPackage 终端的起源盾牌区域。

3. 这会创建额外的缓存层，从而将请求减少到 MediaPackage。

压缩配置

为基于文本的响应启用压缩：

- 为清单文件（*.m3u8、*.mpd）启用压缩
- 不要压缩媒体片段（已压缩）
- 确保所有标题都转发到 MediaPackage （有关最低要求，请参阅MediaTailor CDN 集成所需的标头）

监控缓存性能

跟踪以下关键指标以确保最佳缓存性能：

缓存命中率

目标：媒体细分为 90% 或更高，清单为 70% 或更高

低比率可能表示 TTL 设置不正确或查询参数不必要

源站请求量

监控到达 MediaPackage 终端节点的请求

高音量可能表示存在缓存问题

缓存密钥变体

查看缓存密钥模式以识别不必要的变体

变体太多会降低缓存效率

实施这些缓存优化后，设置监控以跟踪其有效性。有关监控缓存命中率、源请求模式和其他关键性能指标的指导，请参阅监控 CDN 和 MediaTailor 集成的性能 MediaPackage。如果您发现缓存性能不佳或源请求意外，故障排除 MediaPackage、CDN 和集成 MediaTailor请参见以了解故障排除步骤。

监控 CDN 和 MediaTailor 集成的性能 MediaPackage

AWS Elemental MediaTailor 需要进行有效的监控，以保持您的 AWS Elemental MediaPackage 和内容分发网络 (CDN) 集成的最佳性能。本主题提供有关要跟踪的关键指标、要使用的监控工具以及如何设置警报以进行主动问题检测的指导。

在设置监控之前，请确保您的基本集成正常运行。如果您尚未完成基本的内容分发网络集成设置，请从开始MediaTailor 与 CDN MediaPackage 集成。如果您需要对通过监控发现的问题进行故障排除，请参阅CDN 集成疑难解答...

关键绩效指标

监控以下基本指标，确保您 MediaPackage 和 CDN 集成的最佳性能：

CDN 性能指标

如需全面的 CDN 性能指标，包括缓存命中率目标、源站请求量监控和响应时间基准，请参阅 CDN 优化指南CDN 和 MediaTailor 集成的性能基准中的。

针对 CDN 指标的 EMP 特定关键注意事项：

EMP 缓存控制标头

验证内容：确保您的 CDN 支持 EMP 的缓存控制标头以实现最佳 TTL 行为

预期行为：根据 EMP 的标头，不同的内容类型应具有不同的缓存持续时间

有关 EMP 缓存优化的详细指南，请参阅优化 CDN 缓存 MediaTailor 和 MediaPackage 内容交付。

查询参数的影响

要监控的内容：跟踪特定于 EMP 的查询参数如何影响缓存效率

优化目标：确保缓存密钥中仅包含必要的 EMP 查询参数

CDN 响应时间

监控内容：跟踪不同内容类型（清单与区段）的响应时间。

目标值：

- 缓存内容：小于 100 毫秒
- 源站请求：小于 500 毫秒

MediaPackage 性能指标

错误率

监控内容：监控 CDN 和 MediaPackage 终端节点的 HTTP 错误率。请特别注意 4xx 错误，这可能表明存在配置问题。

关键错误代码：

- 400 错误：通常与清单筛选问题有关
- 404 错误：可能表示存在路由或配置问题
- 504 错误：超时问题，尤其是 LL-HLS

请求音量和图案

监控内容：跟踪 MediaPackage 端点的请求模式，以确定使用趋势和容量需求。

值得关注的模式：

- 高峰使用时间
- 请求的地理分布
- 内容类型分发（直播与点播）

延迟指标

End-to-end 延迟

监控内容：对于 LL-HLS 实现，监控从内容摄取到观看者播放的 end-to-end延迟。高延迟可能表明 CDN 配置存在问题。

目标值：

- LL-HLS：延迟小于 3 秒 glass-to-glass
- 常规 HLS：小于 30 秒

清单生成时间

要监控的内容：生成清单所 MediaPackage花费的时间，尤其是在应用筛选的情况下。

目标值：清单生成小于 200 毫秒

监控工具和设置

设置全面的监控工具对于保持最佳性能和在问题影响观看者之前快速识别问题至关重要。如果没有适当的监控，性能下降、缓存效率低下或集成问题可能会被忽视，直到观众体验到较差的播放质量。正确的监控设置可让您全面了解您 MediaPackage 和 CDN 集成的各个方面。

使用以下 AWS 服务和工具来监控您 MediaPackage 和 CDN 的集成：

Amazon CloudWatch

Amaz CloudWatch on 通过收集 MediaPackage 和存储来自这两项服务的指标，为监控您和 CDN 的集成奠定了基础。正确的 CloudWatch 配置可确保您拥有识别性能趋势、解决问题和优化集成所需的数据。如果没有 CloudWatch 指标，您将无法了解系统性能，并且可能要等到问题变得严重后才能发现问题。

设置 CloudWatch 监控以收集全面的指标：

1. MediaPackage 指标：为 MediaPackage 终端节点启用 CloudWatch 指标，以跟踪请求量、错误率和响应时间。
2. CDN 指标：配置 CloudWatch 为收集 CloudFront 指标，包括缓存命中率、源请求计数和错误率。
3. 自定义指标：为特定业务创建自定义指标， KPIs 例如观众参与度或内容受欢迎程度。

CloudWatch 仪表板

创建全面的仪表板以可视化您的指标：

1. 概述控制面板：显示整体系统运行状况的高级指标
2. CDN 性能控制面板：详细的 CDN 指标，包括缓存性能和地理分布
3. MediaPackage 性能控制面板： MediaPackage特定指标，包括请求模式和错误率
4. 延迟控制面板：不同内容类型和区域的 End-to-end延迟指标

日志分析

设置日志分析以进行详细的故障排除：

1. CDN 访问日志：启用和分析 CDN 访问日志，以了解请求模式并识别问题
2. MediaPackage CloudWatch 日志：监控 MediaPackage 日志中是否存在错误和性能问题
3. 日志聚合：使用 Amazon L CloudWatch ogs Insights 或第三方工具分析日志模式

设置警报和通知

警报配置对于主动检测和解决问题至关重要。如果没有适当的警报，问题可能会被忽视，直到它们严重影响观看者体验或导致服务中断。配置良好的警报可帮助您在问题影响观众之前识别和解决问题，并确保您的团队收到需要立即关注的关键问题的通知。

配置主动警报，以便在问题影响查看者之前将其识别出来：

关键警报

为关键问题设置即时警报：

- 错误率高：当 4xx 或 5xx 错误率在 5 分钟内超过 5% 时发出警报
- 缓存命中率下降：当清单的缓存命中率低于 70% 或区段的缓存命中率低于 85% 时发出警报
- 高延迟：延迟超过目标阈值时 end-to-end发出警报
- 源站请求峰值：当源请求比基准增加超过 50% 时发出警报

警告警报

为可能表明正在出现问题的趋势设置警告提醒：

- 性能逐渐下降：在 30 分钟内响应时间增加 20% 时发出警报
- 缓存效率趋势：当缓存命中率显示随时间推移而下降的趋势时发出警报
- 异常流量模式：针对请求量或地理分布的意外变化发出警报

使用监控数据进行优化

利用监控数据持续提高性能：

定期绩效评估

1. 每周回顾：分析每周绩效趋势并确定优化机会
2. 每月容量规划：使用流量模式规划容量需求和 CDN 优化
3. 季度架构审查：评估整体架构效率并确定需要改进的领域

常见的优化操作

根据监控数据，考虑以下优化操作：

- 缓存策略调整：根据实际内容更新模式修改 TTL 值。有关详细的 TTL 优化指南，请参阅针对 CDN 和 MediaTailor 集成的缓存优化。
- 地理优化：在流量大的地区添加 CDN 边缘站点
- 查询参数优化：移除片段缓存中不必要的查询参数
- Origin shield 配置：在起源请求量高的区域实施原点盾牌

有关特定于的详细监控指南 MediaPackage，请参阅 MediaPackage 用户指南 MediaPackage中的监控。

故障排除 MediaPackage、CDN 和集成 MediaTailor

AWS Elemental MediaTailor AWS Elemental MediaPackage 与内容分发网络 (CDN) 的集成可能会遇到影响播放、缓存或其他集成功能的常见问题。遇到播放问题、缓存问题或其他与集成相关的错误时，请使用本指南。

有关全面的 CDN 故障排除指南，包括适用于所有 MediaTailor 实现的通用缓存性能问题、HTTP 错误解决方案、测试程序和诊断技术，请参阅对 CDN 集成进行故障排除。本节重点介绍 MediaPackage 特定的故障排除要求。

在进行故障排除之前，请确保您已正确完成基本的集成设置。如果您尚未设置集成或需要查看设置步骤，请参阅MediaTailor 与 CDN MediaPackage 集成。有关在解决问题后优化缓存性能的指导，请参阅CDN 缓存。

清单筛选错误

根据记录的错误情况， MediaPackage 清单筛选功能存在问题：

清单筛选时出现 HTTP 400 错误

症状：带`aws.manifestfilter`参数的请求返回 HTTP 400 错误请求

经过验证的原因（来自 AWS 文档）：

- 筛选条件导致清单为空（没有符合过滤条件的直播流）
- 过滤器参数名称或值无效
- 格式错误的查询字符串语法
- 重复或重复的过滤器参数
- 筛选参数字符串超过 1024 个字符
- 应用于媒体播放列表或片段的查询参数（不支持）

解决方案

1. 查看您的过滤器参数，确保它们与可用的内容流相匹配。如果筛选结果没有匹配流，则 MediaPackage返回 HTTP 400。
2. 根据支持的参数名称和值格式验证过滤器语法。
3. 检查查询字符串中是否有重复的参数。

4. 确保过滤器参数仅适用于多变体播放列表，而不适用于媒体播放列表或片段。
5. 验证您的参数字符串总数是否低于 1024 个字符。

参考：AWS Elemental MediaPackage 清单筛选错误情况

清单筛选不起作用（HTTP 200 但未应用筛选）

症状：请求返回 HTTP 200 但清单包含所有流，而不是过滤后的子集

可能的原因：

- CDN 未将`aws.manifestfilter`查询参数转发到 MediaPackage
- 在可用流中找不到过滤器参数（返回未经过滤的带有 HTTP 200 的清单）

解决方案

1. 验证您的 CDN 缓存策略是否包含`aws.manifestfilter`在转发的查询字符串列表中。
2. 直接针对 MediaPackage 端点测试过滤器参数（绕过 CDN），以验证它们是否按预期工作。
3. 检查过滤器值是否与内容流的实际特征相匹配。

参考：AWS Elemental MediaPackage 清单筛选错误情况

诊断程序

系统的诊断程序可帮助您快速、高效地确定集成问题的根本原因。遵循结构化方法可以防止将时间浪费在错误的假设上，并确保您解决的是实际问题而不是症状。这些基于证据的诊断步骤旨在隔离问题并指导您找到适当的解决方案。

按照以下循证诊断步骤识别问题：

分析缓存性能

缓存性能分析对于 EMP 集成至关重要，因为较差的缓存效率会导致源负载增加、成本上涨和潜在的播放问题。

有关全面的缓存性能故障排除，包括缓存命中率分析、缓存密钥优化和系统诊断步骤，请参阅 CDN 主故障排除指南CDN 缓存性能问题中的。

EMP 特定的缓存注意事项：

- EMP 缓存控制标头：验证你的 CDN 是否支持 EMP 的缓存控制标头，而不是覆盖它们
- EMP 查询参数：确保缓存密钥中仅包含必要的 EMP 查询参数
- EMP TTL 行为：确认不同的 EMP 内容类型具有适当的缓存持续时间

有关优化 EMP 缓存策略和 TTL 设置的详细指南，请参阅。优化 CDN 缓存 MediaTailor 和 MediaPackage 内容交付

验证清单筛选配置

清单筛选验证至关重要，因为筛选问题可能会导致查看者收到不正确的内容、不支持的格式或他们不应该访问的内容。系统测试有助于确定问题是否与 CDN 配置、过滤器参数语法或内容可用性有关。

系统地测试清单筛选功能：

1. 直接针对 MediaPackage 端点测试过滤器参数（绕过 CDN），以验证它们是否正常工作。
2. 比较已筛选和未经过滤的清单，以确认包含/排除预期的直播流。
3. 验证您的 CDN 缓存策略是否转发了`aws.manifestfilter`查询参数。
4. 检查 HTTP 400 错误，并将其与记录的错误条件进行匹配。

如果您在解决问题后需要实现或修改清单筛选，请参阅使用 MediaTailor、 MediaPackage和 CDN 设置清单筛选以获取完整的设置指南。

验证查询参数配置

确保您的 CDN 仅转发所需的查询参数：

1. 查看您的 CDN 缓存策略以确认其仅包含 AWS 推荐的参数：
   - `aws.manifestfilter`-用于清单筛选
   - `aws.manifestsettings`-用于时移观看
   - `_HLS_msn`还有 `_HLS_part`-用于 LL-HLS 支持
2. 从缓存密钥中移除任何其他查询参数，因为 MediaPackage 忽略它们会降低缓存效率。

参考：使用 AWS Elemental MediaPackage 和 CDNs

错误代码参考

有关记录在案的错误情况及其原因的参考资料：

HTTP 400 错误请求（清单筛选）

记录在案的原因：

- 应用过滤器会导致清单为空
- 参数名称或值无效

- 格式错误的查询字符串语法
- 重复的过滤器参数
- 参数字符串超过 1024 个字符
- 查询媒体播放列表或片段的参数

参考：MediaPackage 清单筛选错误情况

HTTP 200 OK（未应用任何筛选）

记录在案的原因：

- 在可用直播中找不到过滤器参数（返回未经过滤的清单）
- 筛选后仅显示字幕流（返回未经过滤的清单）

参考：MediaPackage 清单筛选错误情况

其他故障排除资源

对于本主题中未涉及的问题，请查阅以下官方 AWS 资源：

- 从中预览清单 AWS Elemental MediaPackage-使用清单预览来解决内容打包问题
- 提高 CloudFront 缓存命中率-CDN 缓存优化综合指南
- 清单筛选- MediaPackage 筛选功能完整指南

AWS Elemental MediaTailor 与 Amazon 集成 CloudFront

AWS Elemental MediaTailor 与 Amazon 集成 CloudFront ，以提高内容交付性能和可靠性。CloudFront 是一个内容分发网络 (CDN)，它通过名为边缘位置的全球数据中心网络分发您的内容。当观众从中请求您的内容时 MediaTailor，会将请求 CloudFront 路由到最近的边缘站点。这种方法可以减少延迟并提高观众的性能。

MediaTailor 与集成 CloudFront 可带来以下几个主要好处：

- 减少了观众访问个性化内容的延迟
- 提高了可扩展性，便于处理庞大的受众规模
- 通过冗余传送路径提高可靠性
- 通过高效的缓存策略优化成本
- 高级功能，例如具有媒体质量感知弹性 (MQAR) 的多区域故障转移

有关 CloudFront 功能的全面信息，请参阅《CloudFront 开发人员指南》。有关 CloudFront 定价的信息，请参阅CloudFront 定价。

为简洁起见，我们有时会使用 “清单” 来统称多变体播放列表、媒体播放列表和。 MPDs

主题

- 使用设置基本 CloudFront 集成 MediaTailor
- 利用 CloudFront 功能优化 MediaTailor 性能
- MediaTailor 使用 MQAR 实现多区域弹性
- 监控您的和 MediaTailor 集成 CloudFront 并排除故障

使用设置基本 CloudFront 集成 MediaTailor

AWS Elemental MediaTailor 与 Amazon 的集成 CloudFront 可提高您的观众的内容交付性能。本主题将指导您为设置基本 CloudFront 发行版 MediaTailor。通过这种集成，您的观众可以通过 CloudFront 网络访问个性化内容。您还将学习如何为不同的内容类型配置适当的缓存。

有关通过传递 CloudFront 用于授权和路由的查询参数的信息，请参见MediaTailor 清单查询参数。有关使用动态变量的高级路由，请参阅MediaTailor 多个内容源的域变量。

先决条件

在 CloudFront 使用进行配置之前 MediaTailor，请确保您具备以下条件：

- 具有创建和管理 CloudFront分配权限的活跃 AWS 账户
- 已配置的 MediaTailor 播放配置（请参阅使用 AWS Elemental MediaTailor 插入广告）
- 您的内容源服务器已正确设置并可以访问
- 对视频流媒体概念的基本理解 (HLS/DASH)

配置 CloudFront 分发

按照以下步骤为以下各项创建和配置 CloudFront 分发 MediaTailor：

为创建 CloudFront 分配 MediaTailor

1. 登录 AWS 管理控制台并在 https://console.aws.amazon.com/cloudfront/v3/ hom CloudFront e 中打开控制台。
2. 选择 Create Distribution（创建分配）。

3. 对于 Origin 域名，请输入您的 MediaTailor 终端节点 URL（例如`a1b2c3d4.mediatailor.us-west-2.amazonaws.com`）。
4. 对于协议，请选择仅限 HTTPS。
5. 在“名称”中，输入一个可以帮助您识别此来源的名称（例如，`mediatailor-origin`）。
6. 配置默认的缓存行为设置：

   a. 对于路径模式，使用默认值 (*)。
   b. 对于“自动压缩对象”，选择“是”。
   c. 对于查看器协议策略，请选择将 HTTP 重定向到 HTTPS。
   d. 对于“允许的 HTTP 方法”，选择 GET、HEAD。
   e. 在缓存策略中，选择CachingDisabled。
   f. 在 O rigin 请求策略中，选择AllViewer转发默认行为的所有标头。

   > **Note**
   >
   > 默认行为 AllViewer 用于安全处理任何与特定路径模式不匹配的内容。清单和区段的特定缓存行为将使用适当的策略单独配置。

7. 配置分发设置：

   a. 对于价格等级，请选择与您的受众位置最匹配的选项。
   b. 对于 AWS WAF Web ACL，请选择现有的 Web ACL 或保留为“请勿启用安全保护”。
   c. 对于默认根对象，请留空。
   d. 对于标准日志记录，请选择开启以启用日志记录。
8. 选择 Create Distribution（创建分配）。

配置缓存行为

创建发行版后，您需要配置其他缓存行为以适当地处理不同类型的内容。本节介绍的基本缓存行为设置 CloudFront。

有关全面的缓存优化，包括高级 TTL 设置、缓存密钥配置和性能调整，请参阅 CDN 优化指南针对 CDN 和 MediaTailor 集成的缓存优化中的。

配置清单缓存行为

不要缓存清单，因为清单包含个性化内容。按照以下步骤配置缓存行为：

配置清单缓存行为

1. 在 CloudFront 控制台中，选择您的发行版。
2. 选择行为选项卡。
3. 选择创建行为。
4. 在路径模式中，输入`*.m3u8`以匹配 HLS 多变体播放列表和媒体播放列表。
5. 在 O rigin 中，选择您的 MediaTailor 起运地。
6. 在缓存策略中，选择CachingDisabled。
7. 在 O rigin 请求策略中，选择AllViewer转发动态内容所需的所有标题。
8. 选择创建。
9. 使用路径模式对 DASH 清单重复这些步骤`*.mpd`进行匹配 MPDs。

此配置可确保每位观众都能收到包含其特定广告内容的个性化清单。CDN 不会缓存这些清单，因此每个请求都会直接发送到。 MediaTailor

配置分段缓存行为

为广告细分和内容区段配置单独的缓存行为，以优化效果并确保正确处理 CORS。

配置广告区段缓存行为

通过`/tm/*`路径模式投放的广告区段需要特定的配置才能正确处理 CORS。按照以下步骤进行操作：

配置广告区段缓存行为

1. 在 CloudFront 控制台中，选择您的发行版。
2. 选择行为选项卡。
3. 选择创建行为。
4. 在 “路径模式” 中`/tm/*`，输入匹配投放的广告区段 MediaTailor。
5. 对于 Origin，选择您的 MediaTailor 分段来源（使用`segments.mediatailor.region.amazonaws.com`主机名）。
6. 在缓存策略中，选择CachingOptimized。
7. 在 O rigin 请求策略中，选择无。
8. 对于响应标头策略，请选择 CORS-with-preflight-and-SecurityHeadersPolicy 以确保响应中包含正确的 CORS 标头。

9. 选择创建。

配置内容分段缓存行为

内容分段可以使用标准缓存策略来获得最佳性能。为不同的区段格式配置单独的行为：

配置内容分段缓存行为

1. 在 CloudFront 控制台中，选择您的发行版。
2. 选择行为选项卡。
3. 选择创建行为。
4. 在路径模式中，输入`*.ts`以匹配 HLS 内容分段。
5. 在 O rigin 中，选择您的内容来源。
6. 在缓存策略中，选择CachingOptimized。
7. 在 O rigin 请求策略中，选择无。
8. 对于响应标头策略，请选择 CORS-with-preflight-and-SecurityHeadersPolicy 以确保对所有内容类型的 CORS 处理保持一致。
9. 选择创建。
10. 使用适当的路径模式对其他内容段格式重复以下步骤：
    - `*.mp4`用于 MP4 细分
    - `*.m4s`适用于 DASH 区段
    - `*.cmfv`并`*.cmfa`适用于 CMAF 细分市场

此配置可确保通过适当的 CORS 处理来适当地缓存广告细分和内容区段。广告细分使用带有 CORS 保护的 MediaTailor 区段来源，而内容细分则使用您的内容来源并经过优化的缓存策略。

更新 MediaTailor 配置

设置好 CloudFront 发行版后，请更新您的 MediaTailor 配置以使用该 CloudFront 域名：

更新您的 MediaTailor 配置

1. 打开 MediaTailor 控制台。
2. 选择要更新的配置。

3. 在 CDN 配置部分，在 CDN 内容 CloudFront分段前缀字段中输入您的分发域名（例如d1234abcdef.cloudfront.net）。
4. 保存更改。

使用此配置， MediaTailor 生成 URLs 指向您的 CloudFront分发而不是直接指向原点的清单。

测试您的集成

配置 CloudFront 发行版并更新 MediaTailor 配置后，请测试集成：

测试您的 CloudFront 和 MediaTailor 集成

1. 通过您的 CloudFront 发行版请求清单（例如，https://d1234abcdef.cloudfront.net/v1/master/12345/my-config/index.m3u8）。
2. 验证清单中是否 URLs 包含指向您的 CloudFront域名的区段。
3. 通过视频播放器播放内容，并验证内容和广告的播放是否正确。
4. 检查 CloudFront 日志，确保请求路由正确。

示例配置

以下是 MediaTailor 具有适当缓存行为的 CloudFront 分发配置示例：

Example CloudFront 分发配置示例

```
{
  "DefaultCacheBehavior": {
    "TargetOriginId": "mediatailor-origin",
    "ViewerProtocolPolicy": "redirect-to-https",
    "AllowedMethods": {
      "Quantity": 2,
      "Items": ["GET", "HEAD"]
    },
    "CachePolicyId": "4135ea2d-6df8-44a3-9df3-4b5a84be39ad",
    "OriginRequestPolicyId": "59781a5b-3903-41f3-afcb-af62929ccde1",
    "Comment": "Default behavior with CachingDisabled and AllViewer"
  },
  "CacheBehaviors": [
    {
```

```
      "PathPattern": "*.m3u8",
      "TargetOriginId": "mediatailor-origin",
      "ViewerProtocolPolicy": "redirect-to-https",
      "CachePolicyId": "4135ea2d-6df8-44a3-9df3-4b5a84be39ad",
      "OriginRequestPolicyId": "59781a5b-3903-41f3-afcb-af62929ccde1",
      "Comment": "Manifest behavior with CachingDisabled and AllViewer"
    },
    {
      "PathPattern": "*.ts",
      "TargetOriginId": "mediatailor-origin",
      "ViewerProtocolPolicy": "redirect-to-https",
      "CachePolicyId": "658327ea-f89d-4fab-a63d-7e88639e58f6",
      "OriginRequestPolicyId": "88a5eaf4-2fd4-4709-b370-b4c650ea3fcf",
      "Comment": "Segment behavior with CachingOptimized and HostHeaderOnly"
    }
  ]
}
```

此示例显示：

- 默认行为：使用`CachingDisabled`和`AllViewer`来安全处理任何与特定路径模式不匹配的内容
- 清单行为 (*.m3u8)：使用和用于动态内容 `CachingDisabled` `AllViewer`
- 区段行为 (*.ts)：用途和 `CachingOptimized` `CORS-with-preflight-and-SecurityHeadersPolicy`

后续步骤

设置基本 CloudFront 集成后 MediaTailor，请考虑以下步骤：

- 使用其他 CloudFront 功能优化性能（参见利用 CloudFront 功能优化 MediaTailor 性能）
- 使用 MQAR 实现多区域弹性（参见）MediaTailor 使用 MQAR 实现多区域弹性
- 设置监控和故障排除（请参阅监控您的和 MediaTailor 集成 CloudFront 并排除故障）

利用 CloudFront 功能优化 MediaTailor 性能

AWS Elemental MediaTailor 除了基本配置之外， CloudFront 还可以通过其他功能来增强 Amazon 的性能。使用设置基本 CloudFront 配置后 MediaTailor，您可以实现其他功能以增强性能、可靠性和自定义选项。这些优化有助于提供更好的观看体验。它们还可以降低成本和原产地负荷。

有关使用动态路由和参数处理进行性能优化的信息，请参见MediaTailor ADS 请求的动态广告变量。有关查询参数优化策略，请参阅MediaTailor 清单查询参数。

使用 Origin Shield 减少原点负荷

Origin Shield 在 CloudFront 边缘站点和您的源服务器之间添加了一个缓存层。此功能对于直播和热门的 VOD 内容非常有用。当许多观众同时请求相同的内容时，这会有所帮助。

通过整合来自多个边缘站点的请求，Origin Shield 可以减少内容来源的负担 MediaTailor 。

为你的起源启用 Orig MediaTailor in Shield

1. 打开 CloudFront 控制台并导航到您的发行版。
2. 选择指向您的 MediaTailor 播放配置的原点。
3. 在 Origin Shield 下，选择 “是”。
4. 从下拉菜单中，选择离您的 MediaTailor终端节点最近的 AWS 区域。
5. 保存更改。

对于高流量事件，Origin Shield 可显著减少源站的请求负载并提高可靠性。有关详细说明，请参阅 CloudFront 开发者指南中的启用 Origin Shield。

使用函数自定义内容交 CloudFront 付

CloudFront 函数允许您在边缘运行轻量级 JavaScript 代码来修改查看者的请求和响应。您可以使用这些函数进行简单的自定义，例如网址修改、标头操作或基本身份验证。对于 MediaTailor 工作流程，Functions 可以帮助完成不需要复杂处理的任务。

CloudFront 函数提供了一种在边缘自定义内容交付的轻量级方法。以下是如何为您的 MediaTailor 集成实现它们：

要实现以下各项的 CloudFront 函数 MediaTailor

1. 在 CloudFront 控制台中，导航到函数。
2. 创建一个新函数并选择适当的用途：

   - 网址操纵-在多变播放列表、媒体播放列表和 MPD 请求到达 URLs 之前对其进行修改 MediaTailor
   - 标头操作-添加或修改请求标头
   - 简单身份验证-用于验证令牌或查询参数

3. 编写你的 JavaScript 函数代码。
4. 使用示例 MediaTailor 请求测试您的函数。
5. 发布该函数并将其与分配的缓存行为相关联。

Example URL 标准化示例 CloudFront 函数

```
function handler(event) {
    var request = event.request;
    var uri = request.uri;

    // Normalize URLs to lowercase to improve cache hit ratio
    if (uri.includes('.m3u8') || uri.includes('.mpd')) {
        request.uri = uri.toLowerCase();
    }

    return request;
}
```

有关更多信息和代码示例，请参阅 CloudFront 开发者指南中的使用 CloudFront 函数在边缘进行自定义。

使用 Lambda 实现高级自定义 @Edge

当您需要比 CloudFront 函数所能提供的更复杂的处理能力时，可以使用 Lambda @Edge。此服务允许您在 CloudFront 边缘位置运行 Node.js 或 Python 函数。

Lambda @Edge 函数可以执行复杂的操作，例如复杂的身份验证、更大的响应修改或第三方 API 集成。

要进行更复杂的自定义，请将 Lambda @Edge 函数与 MediaTailor 你的 CloudFront和集成一起使用：

要使用 Lambda @Edge 来实现 MediaTailor

1. 在美国东部（弗吉尼亚州北部）区域创建 Lambda 函数。
2. 为以下用例之一编写函数代码：

    - 网址操纵-在多变播放列表、媒体播放列表和 MPD 请求到达 URLs 之前对其进行修改 MediaTailor

- A/B 测试-将用户路由到不同的广告决策服务器
- 请求身份验证-添加身份验证标头
- 修改响应标头-添加 CORS 标头

3. 发布您的函数版本并创建函数别名。
4. 在适当的触发点（查看者请求或查看者响应）将函数与您的 CloudFront 分布相关联。

> **Note**
>
> 使用 Lambda @Edge 时 MediaTailor，如果您计划使用媒体质量感知弹性 (MQAR) 功能，请避免使用面向源的触发器（源请求和源站响应），因为这些功能目前不兼容。

有关更多信息和代码示例，请参阅 CloudFront 开发者指南中的使用 Lambda @Edge 在边缘进行自定义。

其他性能优化技巧

考虑以下额外优化以进一步提高性能：

优化缓存命中率

在 CloudFront 指标中监控你的缓存命中率，并寻找改善它的机会：

- 标准化 URL 模式以提高缓存密钥的一致性
- 使用查询字符串白名单在缓存密钥中仅包含必需的参数
- 考虑使用 CloudFront 函数实现 URL 标准化

减少延迟

实施以下技术以最大限度地减少延迟：

- 为基于文本的响应启用 Brotli 压缩
- 使用 HTTP/2 或 HTTP/3 来提高连接效率
- 考虑启用对现代网络的 IPv6 支持

成本优化

在性能与成本效益之间取得平衡：

- 使用 Origin Shield 减少冗余的源站请求

- 考虑根据受众地理位置选择价格等级
- 对区段实施主动缓存以减少源站流量

后续步骤

使用 CloudFront 功能优化性能后，请考虑以下步骤：

- 使用 MQAR 实现多区域弹性（参见）MediaTailor 使用 MQAR 实现多区域弹性
- 设置监控和故障排除（请参阅监控您的和 MediaTailor 集成 CloudFront 并排除故障）

MediaTailor 使用 MQAR 实现多区域弹性

AWS Elemental MediaTailor 通过媒体 Quality-Aware 弹性 (MQAR) 增强了多区域弹性，这是一项先进的亚马逊 CloudFront 功能，有助于 MediaTailor提供最佳的流媒体体验。当您在不同 AWS 地区有多个起源时，它会自动选择质量得分最高的原产地。此功能对于需要不间断服务的直播特别有价值。

MQAR 是如何运作的

MQAR CloudFront 允许自动选择质量分数最高的原产地。这样可以确保观众获得最佳的流媒体体验。正确配置后，MQAR 可提供以下好处：

- 自动选择最高质量的原产地
- 停机期间 AWS 区域间的无缝故障转移
- 以最少的干扰改善了观众体验
- Real-time 质量监测和适应

第 1 步：验证 MQAR 要求

在实施 MQAR 之前，请验证您的基础设施是否满足这些要求。MQAR 的工作原理是比较来自多个来源的质量分数，因此您需要在多个 AWS 区域提供相同的内容。

- 编码器使用周期锁定的 CMAF 采集流集将对齐的采集流发送到所有 MediaPackage 频道
- 不同 AWS 区域中的两个相同 MediaPackage 频道，源端点相同
- MediaPackage 频道的 CMAF 推流（默认启用）
- CloudFront 配置为支持 MQAR 的发行版
- MediaTailor 每个 MediaPackage 端点的配置

第 2 步：为 MQAR 配置编码器

您的编码器需要在所有区域生成一致、同步的输出，以便 MQAR 有效运行。这种一致性 CloudFront 允许在原产地之间进行准确的质量比较。

使用以下设置配置您的 MediaLive 输出：

- 确保 CMAF 输出组中的所有视频帧率一致（全部为小数帧速率或全部整数帧率）。
- 避免在编码会话中在小数帧速率和整数帧速率之间过渡。
- 配置输出分段序列号，使其在编码会话中永远不会倒退。
- 在所有区域使用相同的编码器输出名称。

有关配置 MQAR MediaLive 的更多信息，请参阅 MediaLive 用户指南中的使用 MQCS。

第 3 步：为 MQA MediaPackage R 进行配置

使用以下配置设置您的 MediaPackage 频道和终端节点：

为 MQA MediaPackage R 进行配置

1. 在每个 AWS 区域创建相同的频道和终端节点配置。
2. 使用 CMAF 作为通道输入类型。
3. 对于主 MediaPackage 源，使用以下设置启用强制端点错误配置：
    - 陈旧的多变体播放列表、媒体播放列表或 MPD
    - 多变量播放列表、媒体播放列表或 MPD 不完整
    - Slate 输入
4. 对于备份 MediaPackage 源，不要启用这些错误配置，以最大限度地提高成功故障转移的机会。

有关配置 MQAR MediaPackage 的更多信息，请参阅 MediaPackage 用户指南 AWS Elemental MediaPackage中的使用来利用媒体质量分数。

步骤 4：为 MQA CloudFront R 进行配置

在 CloudFront 配置中，您启用 MQAR 并定义它如何在您的来源之间进行选择。在启用媒体质量得分选项的情况下创建来源组。

为 MQA CloudFront R 进行配置

1. 创建源指向您的 MediaTailor终端节点的 CloudFront 分配。
2. 创建包含这些来源的原始组。
3. 在原始组设置中，启用 “媒体质量得分” 选项。
4. 将故障转移标准配置为包含 404 未找到响应代码。您可以选择添加其他 4xx/5xx 代码。
5. 为每个通道的路径模式创建单独的缓存行为。这样可以防止在同一个来源组提供多个频道时混用分数。

> **Note**
>
> 在与您的分配缓存行为相关的面向源的触发器（源请求和源响应）中使用 Lambda @Edge 函数时，MQAR 不可用。

有关配置 MQAR CloudFront 的更多信息，请参阅开发人员指南中的媒体质量感知弹性。 CloudFront

第 5 步：为 MQA MediaTailor R 进行配置

要完成 MQAR 设置，请在每个区域进行配置，使其 MediaTailor 与您的多区域架构配合使用。这样可以确保无论 CloudFront 选择哪个来源，都能始终如一地插入广告。

为 MQA MediaTailor R 进行配置

1. 在每个区域创建相同的 MediaTailor 配置，指向相应的 MediaPackage 终端节点。
2. 配置 CDN 内容分段前缀以使用您的 CloudFront 分发域。
3. 确保各地区的广告决策服务器配置相同。

这种设置可确保无论根据质量得分 CloudFront 选择哪个来源， MediaTailor 都能持续不断地对广告进行个性化设置。

第 6 步：测试您的 MQAR 配置

设置 MQAR 后，测试您的配置以确保其按预期运行：

测试您的 MQAR 配置

1. 通过您的 CloudFront 分发请求内容。

2. 监控实时日志，CloudFront 以验证是否根据质量分数选择了来源。
3. 模拟主 AWS 区域的故障以测试故障转移行为。
4. 验证广告插入在故障转移期间是否能正常工作。

后续步骤

在实现 MQAR 之后，请考虑以下后续步骤：

- 为您的 MQAR 配置设置监控和故障排除（参见监控您的和 MediaTailor 集成 CloudFront 并排除故障）
- 使用实现自动部署 AWS CloudFormation （参见使用自动化 MediaTailor 和 CDN CloudFormation）

监控您的和 MediaTailor 集成 CloudFront 并排除故障

AWS Elemental MediaTailor 与亚马逊的集成 CloudFront 需要持续的监控和故障排除以保持最佳性能。在实施 CloudFront 和 MediaTailor 集成后，监控性能并准备对任何问题进行故障排除。内容分发网络 (CDN) 提供的工具可帮助您了解发行版的表现并识别潜在问题。

为您的集成设置监控

实施以下监控策略来跟踪您的 CloudFront 和 MediaTailor集成的性能：

为您的 CloudFront 和 MediaTailor 集成设置监控

1. 启用 CloudFront 标准日志：
    a. 在 CloudFront 控制台中，选择您的发行版。
    b. 选择日志选项卡。
    c. 在 “标准日志” 下，选择 “编辑” 。
    d. 选择开启并配置 Amazon S3 存储桶用于日志存储。
    e. 选择保存更改。
2. 设置 CloudFront 实时日志：
    a. 在 CloudFront 控制台中，选择您的发行版。
    b. 选择日志选项卡。
    c. 在 “Real-time 日志” 下，选择 “编辑” 。

d. 选择开启并配置 Amazon Kinesis 数据流或 Amazon Data Firehose 传输流。

e. 在日志配置中包含以下字段：

- `time-to-first-byte`-响应时间
- `sc-status`-HTTP 状态码
- `c-ip`-查看者 IP 地址
- `cs-uri-stem`-请求 URI 路径
- `cs-user-agent`-用户代理
- `x-edge-result-type`-结果类型（命中、未命中、错误）

f. 选择保存更改。

3. 创建 CloudWatch 仪表板：

a. 在 CloudWatch 控制台中，选择控制面板。

b. 请选择创建控制面板。

c. 为这些 CloudFront 指标添加小工具：

- 请求
- BytesDownloaded
- 4xx ErrorRate
- 5xx ErrorRate
- TotalErrorRate
- CacheHitRate

4. 设置 CloudWatch 警报：

a. 在 CloudWatch 控制台中，选择警报。

b. 选择创建警报。

c. 针对以下情况创建警报：

- 5xx ErrorRate > 1% 持续 5 分钟
- 4xx 在 5 分钟内ErrorRate 大于 5%
- CacheHitRate 30 分钟内小于 80%

监控 MQAR 性能

使用 MQAR 时，监控可帮助您了解如何在来源之间进行 CloudFront 选择以及质量分数是否符合预期。 Real-time 日志显示了这些决策的发生情况。

监控 MQAR 表现

1. 为您的 CloudFront 分发启用实时日志。
2. 在日志配置中包含以下字段：
   - `r-host`-所选来源的主机名
   - `sr-reason`-选择原产地的原因
   - `x-edge-mqcs`-媒体质量信心分数
3. 在亚马逊 Kinesis Data Streams 或 Amazon Data Firehose 中设置日志目的地。
4. 根据这些指标创建仪表板或警报，以监控质量分数和原产地选择决策。

Example MQAR CloudWatch 监控的示例仪表板

使用以下指标创建 CloudWatch 仪表板：

- 按地区划分的原产地选择次数
- 一段时间内的平均质量分数
- 故障转移事件
- 按来源划分的 4xx 和 5xx 错误率

有关设置实时日志的更多信息，请参阅 CloudFront 开发者指南中的Real-time日志。

排查常见 问题

即使经过仔细的规划 CloudFront ，您的和 MediaTailor 整合也可能会出现问题。了解常见问题和解决方案有助于您快速解决问题并最大限度地减少对浏览者的影响。

如果您在 CloudFront 和 MediaTailor 集成方面遇到问题，请查看以下常见问题和解决方案：

清单缓存问题

症状：清单过时或广告未更新

解决方案：确认您使用的是多变量播放列表、媒体播放列表和 MPD 路径的`CachingDisabled`缓存策略。检查您的配置是否正确转发了查询参数。

CORS 错误

症状：浏览器控制台在访问内容时显示 CORS 错误

解决方案：使用相应的 CORS 标头配置响应标头策略，并将其与您的缓存行为相关联。

MQAR 不起作用

症状：原产地选择忽略质量分数

解决方案：检查您是否在来源组设置中启用了媒体质量得分选项。确认你没有使用面向 Origin 的 Lambda @Edge 触发器。

广告插入失败

症状：广告无法正确插入

解决方案：验证您的 MediaTailor 配置指向您的 CloudFront 分发内容段前缀。检查您的设置是否转发了所有必需的标题。

对于更复杂的问题，您可以使用以下故障排除方法：

1. 检查 CloudFront 分发日志中是否存在错误模式
2. 使用浏览器开发者工具检查网络请求
3. 将 MediaTailor 直接的清单内容与通过交付的内容进行比较 CloudFront
4. 使用支持详细记录的简单播放器进行测试

如需更多故障排除帮助，请参阅 MediaTailor 用户指南中的故障排除部分。

故障排除工作流程

遵循以下系统方法来解决与您的 CloudFront 和 MediaTailor集成有关的问题：

对 MediaTailor 集成问题 CloudFront 进行故障排除

1. 确定具体问题：
    a. 确定问题是影响所有观众还是仅影响部分观众

b. 确定哪些内容类型受到影响（清单、区段或两者）
c. 记下任何错误消息或症状

2. 检查 CloudFront 日志：
   a. 查找错误状态码（4xx 或 5xx）
   b. 检查缓存 hit/miss 模式
   c. 验证请求是否已路由到正确的来源
3. 验证配置：
   a. 检查缓存行为以确定路径模式是否正确
   b. 验证缓存策略是否已正确应用
   c. 确认原始请求策略正在转发必要的标头
4. 测试直接访问：
   a. 尝试直接访问来自的内容 MediaTailor （绕过 CloudFront）
   b. 比较直接访问和 CloudFront 访问之间的响应
5. 实施解决方案：
   a. 根据您的发现应用相应的修复方法
   b. 测试以验证问题是否已解决
   c. 记录问题和解决方案以备将来参考

后续步骤

为您的 CloudFront 和 MediaTailor 集成设置监控和故障排除后，请考虑以下后续步骤：

- 使用实现自动部署 AWS CloudFormation （参见使用自动化 MediaTailor 和 CDN CloudFormation）
- 为常见操作场景和故障排除程序创建运行手册
- 为常见问题设置自动补救措施

为 MediaTailor 广告投放设置第三方 CDNs

CDNs 像 Akamai 和 Fastly 这样的第三方可以显著提高 AWS Elemental MediaTailor 广告投放的性能和可扩展性，同时降低带宽成本。但是，个性化广告的 CDN 配置需要与标准视频交付不同的特定设置。

有关通过第三方传递查询参数的信息 CDNs，请参阅MediaTailor 清单查询参数。有关使用动态变量的高级路由配置，请参阅MediaTailor ADS 请求的动态广告变量。

本指南将引导您完成设置第三方 CDN 以实现最佳运行的完整过程。 MediaTailor您将学习如何配置两个基本行为：

- Manifest bypass：通过防止清单缓存，确保每位观看者都能收到个性化的广告插入
- 分段缓存：通过高效缓存视频内容，最大限度地提高性能并降低成本

配置过程通常需要 30-60 分钟，包括提供商选择、设置、验证和优化。完成后，您将拥有一个 CDN 配置，该配置可以高效投放个性化广告，同时保持最佳的观众体验。

> **Note**
>
> 本指南重点介绍 Akamai 和 Fastly 的配置。有关 CloudFront 设置说明，请参阅CloudFront 集成。

先决条件

在使用设置第三方 CDN 之前 MediaTailor，请确保：

- 包含您的内容来源和广告决策服务器的有效 MediaTailor 配置
- 访问您的 CDN 的配置接口
- 内容中使用的文件扩展名列表（.m3u8、.mpd、.ts 等）
- 您的 CDN 提供商的文档供参考

有关 CloudFront 安装说明（而不是第三方） CDNs，请参阅CloudFront 集成。

术语

要了解 CDN 配置要求，您需要了解以下清单类型：

- HLS 表现为：
  - 多变播放列表：包含媒体播放列表链接的顶级清单
  - 媒体播放列表：包含内容分段链接的二级清单
- DASH 表现为：

- MPD（媒体演示文稿描述）：DASH 清单的标准术语

在讨论常见配置要求时，本指南将所有清单文件（多变体播放列表、媒体播放列表和 MPDs）统称为清单。

有关适用于所有提供商的常规 CDN 配置原则，请参阅通过以下方式设置 CDN 集成 MediaTailor。

有关 CDN 优化指南，请参阅CDN 和 MediaTailor 集成的性能优化指南。

配置 CDN 缓存规则

CDN 缓存配置对于 MediaTailor 广告投放至关重要，因为它决定了您的内容如何吸引观众。适当的配置可确保每个查看者的清单保持个性化，同时有效缓存区段，从而减少源负载并提高性能。

每个 CDN 提供商此配置通常需要 15-30 分钟，并且需要两种不同的行为：

- 清单处理：防止缓存以确保每位观看者都能收到个性化的广告插页
- 分段缓存：最大限度地提高视频内容的缓存效率以提高交付性能

请按照以下步骤配置 CDN 的缓存规则，以实现最佳广告投放。

从以下选项卡中选择您的 CDN 提供商以获取具体说明：

Akamai

在您的 Akamai 属性中配置以下两种行为：

- 防止缓存的清单处理
- 分段缓存以获得最佳性能

配置清单交付

配置您的 Akamai CDN 以避免缓存清单，以便每位查看者都能收到个性化广告。

清单文件包含为每位观看者 MediaTailor生成的个性化广告插入点。缓存这些文件会导致所有观众看到相同的广告，从而违背个性化广告的目的。

请按照以下步骤处理清单请求（以.m3u8、.mpd 或.smil 结尾的文件）：

1. 创建一种行为来匹配清单文件扩展名（.m3u8、.mpd、.smil）

2. 将缓存选项设置为无存储
3. 配置缓存密钥以包含所有查询参数
4. 启用转发主机标头以实现正确的源路由
5. 为所有标头配置标头转发。有关最低要求，请参阅MediaTailor CDN 集成所需的标头。

配置分段传送

将您的 Akamai CDN 配置为缓存视频片段，以最大限度地提高 CDN 效率并减少源负载。

视频片段是可以安全缓存的实际内容文件，因为它们不包含个性化信息。适当的片段缓存可以降低带宽成本，并提高观众的播放性能。

按照以下步骤处理分段请求（以.ts、.mp4、.m4s 等结尾的文件）：

1. 创建行为以匹配分段文件扩展名（.ts、.mp4、.m4s）
2. 将 Honor Origin 缓存控制设置为是
3. 配置缺少原始标头时的默认 time-to-live (TTL) 设置：
   - 默认 TTL：86400 秒（24 小时）
   - 最大 TTL：604800 秒（7 天）

> **Note**
>
> 配置这些行为后，在 Akamai 控制中心激活您的属性更改。
> 这些更改将在激活后生效。

Fastly

在你的 Fastly 服务中创建以下两个配置：

- 防止缓存的清单处理
- 分段缓存以获得最佳性能

配置清单交付

将您的 Fastly CDN 配置为绕过清单文件的缓存，以便每位观看者都能收到个性化的广告内容。

每个请求都必须到达 MediaTailor 清单文件，以确保广告的个性化效果正确。绕过这些文件的缓存可确保正确处理每个查看者的独特定位参数。

请按照以下步骤处理清单请求：

1. 创建请求条件以识别清单路径
2. 将这些请求的缓存条件设置为 “不缓存”
3. 配置转发设置以包含所有查询参数
4. `User-Agent`添加到您的标题转发配置中

配置分段传送

将 Fastly CDN 配置为缓存视频片段，以提高交付性能并减少源站流量。

区段缓存对于具有成本效益的交付和最佳的观看者体验至关重要。这些文件对所有查看者来说都是相同的，并且可以从 CDN 缓存中受益匪浅。

对于区段请求，请按照以下步骤操作：

1. 创建请求条件以识别区段路径
2. 将缓存设置设置为尊重源缓存标头
3. 将缺少原始标头时的默认 time-to-live (TTL) 配置为 86400 秒（24 小时）

> **Note**
>
> 进行这些更改后，激活新版本的 Fastly 服务。
> 该配置在激活后生效。

验证您的 CDN 配置

在您引导生产流量通过之前，验证可确保您的 CDN 配置正常运行。这些测试证实了广告个性化功能正常，并且缓存可以提供预期的性能优势。

完成验证通常需要 10-15 分钟，如果可能，应在多个地理位置进行。

设置 CDN 后，请执行以下检查以验证其是否正常运行：

1. 测试清单个性化：
   a. 使用不同的广告参数请求相同的内容网址
   b. 验证每个请求是否返回不同的广告插页
2. 测试段缓存：
   a. 检查 CDN 指标以了解分段缓存命中率（应大于 90%）
   b. 监控源站流量以确认其低于直接配送
3. 测试播放性能：
   a. 通过 CDN 从不同位置播放内容
   b. 验证播放流畅无缓冲问题

有关全面的测试方法和高级验证程序，请参阅CDN 和集成的测试和 MediaTailor 验证。

优化 CDN 性能

验证基本配置后，实施以下优化以最大限度地提高性能并最大限度地降低成本：

监控关键绩效指标

跟踪以下指标以确保最佳性能：

缓存命中率

目标：视频片段大于 90%

影响：更高的比率可以减少原点负载并改善观众体验

监控：每天查看 CDN 提供商的分析仪表板

源站响应时间

目标：清单请求不到 200 毫秒

影响：更快的清单交付速度缩短了查看者的启动时间

监视器：设置响应时间超过 500 毫秒的警报

错误率

目标：所有请求均低于 0.1%

影响：错误率高表明存在配置问题或源站问题

监控：设置错误率超过 1% 的警报

微调缓存行为

根据您的内容特征和观看者模式调整以下设置：

区段 TTL 优化

直播内容：使用较短的 TTL（1-4 小时）来确保及时更新

VOD 内容：使用更长的 TTL（24-48 小时）以最大限度地提高缓存效率

广告细分：对于经常更新的广告内容，可以考虑缩短 TTL（30 分钟到 2 小时）

有关所有 MediaTailor 工作流程的全面的 TTL 建议和缓存策略，请参阅针对 CDN 和 MediaTailor 集成的缓存优化。

地理优化

多区域来源：根据查看者位置配置来源选择

边缘位置：在观众高度集中的区域启用其他边缘位置

故障转移：配置备份源以实现高可用性

优化成本

实施以下策略以降低 CDN 成本，同时保持性能：

- 压缩：为清单文件启用 gzip 压缩以减少带宽使用量
- 清除策略：实现选择性缓存清除，而不是清除全部缓存
- 流量分析：每月查看流量模式以确定优化机会
- 层级选择：根据您的性能要求使用适当的 CDN 服务层

解决第三方 CDN 问题

CDN 配置问题通常表现为广告个性化问题或性能下降。使用这种系统的方法来识别和解决影响 MediaTailor 广告投放的最常见问题。

通过检查具体症状并应用相应的解决方案，大多数故障排除可以在 15-30 分钟内完成。

如果观众在广告投放或播放质量方面遇到问题，请使用本指南来识别和解决常见的 CDN 配置问题：

Akamai：缓存的清单

症状：即使您配置了不同的定位参数，观众也会看到相同的广告。

解决方案：验证您是否对清单路径应用了 No Store 缓存选项。

此外，请确认您在缓存密钥中包含了查询参数。

Fastly：缓存密钥不正确

症状：观众体验到的广告个性化不一致。

观众可能还会看到面向其他观众的广告。

解决方案：确认您已将转发设置配置为在缓存密钥中包含所有查询参数。

一般：源站流量大

症状：您的源服务器出现异常高的流量

解决方案：验证分段缓存设置和 time-to-live (TTL) 值。

检查 CDN 指标中的缓存命中率。

常规：播放错误

症状：观看者遇到缓冲或播放失败

解决方案：检查 CDN 路由规则和源站运行状况。

确认所有必需的标头都已正确转发。

> **Note**
>
> 如果这些解决方案无法解决您的问题，请查看 CDN 提供商的文档。您也可以联系他们的支持团队以获取其他故障排除步骤。

有关一般的 CDN 故障排除指南，请参阅对 CDN 集成 MediaTailor 和 CDN 集成问题进行故障排除。

CDN 和 MediaTailor 集成的性能优化指南

AWS Elemental MediaTailor 通过系统的内容交付网络 (CDN) 优化，可以最大限度地提高性能。无论您是在实施服务器端广告插入 (SSAI)、渠道组装还是组合工作流程，优化原则和绩效目标都保持一致。本指南提供了适用于所有 MediaTailor 实现的全面优化技术和基准。

有关使用动态变量和配置别名的高级路由优化，请参阅MediaTailor ADS 请求的动态广告变量。有关查询参数优化策略，请参阅MediaTailor 清单查询参数。

优化工作流程概述：

1. 配置缓存-设置适当的 TTL 值和缓存行为
2. 优化路由-配置请求路由和源站策略
3. 衡量绩效-根据既定基准进行跟踪
4. 应用高级技术-实现其他优化功能

主题

- 针对 CDN 和 MediaTailor 集成的缓存优化
- 针对 CDN 和 MediaTailor 集成的请求路由优化
- CDN 和 MediaTailor 集成的性能基准
- 用于 CDN 和 MediaTailor 集成的高级优化技术

针对 CDN 和 MediaTailor 集成的缓存优化

AWS Elemental MediaTailor 缓存要求因工作流程类型和内容格式而异。正确的缓存配置对于实现最佳性能、成本效益和观看者体验至关重要。以下各节为不同的 MediaTailor实现提供了详细的缓存指导。

服务器端广告插入 (SSAI) 缓存

对于服务器端广告插入工作流程，无法缓存个性化清单，但应积极缓存内容和广告细分：

SSAI CDN 缓存设置

内容类型	TTL	路径模式	缓存关键元素
多变体播放列表	0 秒	/v1/master/ *	网址路径 + 所有查询参数

内容类型	TTL	路径模式	缓存关键元素
媒体播放列表	0 秒	/v1/manifest/ *	网址路径 + 所有查询参数
DASH MPDs	0 秒	/v1/dash/ *	网址路径 + 所有查询参数
内容细分	24 小时以上	特定于内容的路径	仅限网址路径
广告细分	24 小时以上	/v1/segment/ *	仅限网址路径

- 将个性化清单的 TTL 设置为 0 秒，以确保观众收到广告内容 up-to-date
- 为内容和广告区段配置更长的 TTL 值，以最大限度地提高缓存效率
- 如果您支持定向广告，请设置缓存行为，在缓存密钥中包含个性化参数
- 在 CDN 级别实现请求折叠以高效处理并发请求

推荐的 TTL 配置设置

为了获得最佳 SSAI 性能，请使用以下特定的 TTL 设置配置 CDN 缓存策略：

SSAI TTL 配置设置

内容类型	TTL 设置	推荐值
广告细分	最小 TTL	1 秒
广告细分	最大 TTL	86400 秒（24 小时）
广告细分	默认 TTL	86400 秒（24 小时）
内容细分	最小 TTL	1 秒
内容细分	最大 TTL	86400 秒（24 小时）
内容细分	默认 TTL	86400 秒（24 小时）

这些设置可确保：

- 最小 TTL 为 1 秒：允许在需要时快速使缓存失效，同时防止过多的源请求
- 最大 TTL 为 24 小时：在缓存效率和内容新鲜度要求之间取得平衡
- 默认 TTL 为 24 小时：为没有明确缓存控制标头的区段提供最佳缓存

服务器引导式广告插入 (SGAI) 缓存

服务器引导式广告插入 (SGAI) 通过使用可预测网址模式的可缓存媒体清单，实现高效的 CDN 缓存。本节重点介绍 CDN 特定的配置要求，以实现最佳 SGAI 性能。

SGAI 的 CDN 缓存配置

使用以下特定于 SGAI 的缓存行为来配置 CDN：

SGAI CDN 缓存设置

内容类型	TTL	路径模式	缓存关键元素
SGAI 多变体播放列表（请勿缓存）	0 秒（不缓存）	/v1/master/ *	URL 路径 + 选定的查询参数
SGAI 媒体播放列表	1-4 秒（半段长度）	/v1/i-media/ *	URL 路径 + 选定的查询参数
资产列表响应（请勿缓存）	0 秒（不缓存）	/v1/插页式广告/ *	网址路径 + 所有查询参数
广告细分	24 小时以上	广告专用路径	仅限网址路径

缓存行为配置

为 SGAI 内容设置专用缓存行为：

- SGAI 清单行为-为 TTL 为 1-4 秒的`/v1/i-media/*`路径创建缓存行为
- 资产列表行为-为 TTL 为 0 秒 TTL 的`/v1/interstitials/*`路径创建缓存行为
- 查询参数处理-在缓存密钥中仅包含基本的定位参数，以最大限度地提高缓存效率
- Origin 请求标头-转发广告定位所需的标头，同时保持可缓存性

频道集合缓存

对于频道汇编工作流程，清单可以短时间缓存，而片段则应积极缓存：

频道集合 CDN 缓存设置

内容类型	VOD TTL	直播 TTL	路径模式	缓存关键元素
多变体播放列表	5-30 分钟	5-10 秒	频道特定路径	网址路径 + 所有查询参数
媒体播放列表	5-30 分钟	2-5 秒	频道特定路径	网址路径 + 所有查询参数
短跑 MPDs	5-30 分钟	5-10 秒	频道特定路径	网址路径 + 所有查询参数
内容细分	24 小时以上	5-15 分钟	特定于内容的路径	仅限网址路径
广告细分	24 小时以上	24 小时以上	广告专用路径	仅限网址路径

- 为清单设置短 TTL 值，以确保观众能收到节目 up-to-date
- 为内容分段配置更长的 TTL 值以最大限度地提高缓存效率
- 如果您支持时移查看，请设置在缓存键中包含时移参数的缓存行为
- 在缓存密钥中加入查询参数以正确处理时移查看请求

有关详细的 TTL 配置设置和最佳实践，请参阅the section called “缓存优化”。

SSAI 和频道集合缓存的组合

在实现信道组合和 SSAI 时，请确保这两个服务的缓存策略保持一致，以避免冲突并优化性能：

组合工作流缓存设置比较

内容类型	频道组装	SSAI	综合推荐
VOD 清单	5-30 分钟	0 秒	（使用单独的配置）

内容类型	频道组装	SSAI	综合推荐
现场清单	2-10 秒	0 秒	（使用单独的配置）
SGAI VOD 清单	5-30 分钟	5-30 分钟	5-30 分钟（可缓存清单）
SGAI Live 清单	2-4 秒	2-4 秒	2-4 秒（可缓存清单）
内容细分	24 小时以上	24 小时以上	24 小时以上（一致）
广告细分	24 小时以上	24 小时以上	24 小时以上（一致）

此配置可最大限度地提高缓存效率，同时确保观众收到个性化广告插入的 up-to-date清单。

针对 CDN 和 MediaTailor 集成的请求路由优化

为所有 AWS Elemental MediaTailor CDN 集成实施以下路由优化：

- 为清单和区段请求创建单独的缓存行为
- 配置源站请求策略以控制标头转发
- 设置正确的错误处理和故障转移机制
- 如果您的 CDN 中可用，请实施原点盾牌以减少源站负载
- 在 CDN 级别实现请求折叠以高效处理并发请求

CDN 和 MediaTailor 集成的性能基准

在优化 AWS Elemental MediaTailor CDN 集成时，请以这些性能基准为目标。这些目标适用于所有 MediaTailor 实现，包括 SSAI、渠道组装和组合工作流程：

缓存命中率目标

内容分段：缓存命中率大于 95%

广告细分：缓存命中率大于 90%

清单：不适用于 SSAI（不应缓存以进行个性化广告插入）；85% 以上适用于渠道组装

延迟基准测试

清单请求延迟：小于 100 毫秒 (P95)

内容分段交付：小于 50 毫秒 (P95)

广告细分投放：小于 75 毫秒 (P95)

End-to-end 启动时间：小于 2 秒

起源负载指标

每位观众的源请求数：每位观看者每分钟少于 0.1 个请求

每位观看者的源带宽：低于观看者总带宽的5%

错误率目标

明显错误：低于 0.1%

分段错误：小于 0.01%

玩家报告的缓冲次数：低于 1%

可扩展性基准

Support 在高峰事件期间支持正常流量的 10 倍而不会降级

每个通道每秒能够处理超过 1000 个请求

使用 Amazon CloudWatch 指标来跟踪这些绩效指标。有关详细的监控说明，请参阅设置监控工具。

用于 CDN 和 MediaTailor 集成的高级优化技术

在实现了基本的缓存和路由优化之后，可以考虑使用以下高级技术来进一步增强性能：

Origin Shield 实现

Origin Shield 在 CDN 边缘站点和您的源服务器之间添加了一个缓存层，从而减少了源站负载并提高了性能：

- 为高流量内容和直播启用 Origin Shield
- 选择你所在 MediaTailor 地区附近的 Origin Shield 地点
- 监控 Origin Shield 缓存命中率并根据需要进行调整
- 考虑在多个 Origin Shield 地点进行全球部署

内容压缩优化

优化内容压缩以减少带宽并提高交付速度：

- 为清单文件启用 gzip 压缩
- 为清单压缩配置接受编码标头转发 MediaTailor
- 在支持的地方使用 Brotli 压缩以节省更多带宽
- 避免压缩已经压缩的视频片段

区域优化策略

通过区域策略优化全球受众的表现：

- 在多个区域为全球受众部署 MediaTailor 配置
- 使用地理路由将观众引导到最近的区域 MediaTailor
- 配置区域故障转移以实现高可用性
- 分别监控区域绩效指标

CDN 和 MediaTailor 集成的架构注意事项

在架构中正确定位内容分发网络 (CDN)，以确保最佳性能和可靠性。 AWS Elemental MediaTailor推荐的架构将 CDN 放置在查看者之间 MediaTailor，而不是放在您的来源 MediaTailor 之间。

有关详细的架构图和工作流程说明，请参阅以下主题。

- 使用 CDN 进行广告插入获取广告插入架构图和详细的工作流程
- 了解 CDN 架构用于通道组装架构图和工作流程

在架构中正确放置 CDN：

1. 将你的 CDN 放在玩家和之间 MediaTailor （而不是介于 MediaTailor 和你的起源之间）。

   这种架构允许您的 CDN 缓存广告细分和内容区段。同时， MediaTailor 可以为每位观众生成个性化清单。
2. 为不同的请求类型创建单独的缓存行为：
   - 清单请求（无缓存）

- 内容分段（较长的 TTL）
- 广告细分（较长的 TTL）

3. 配置正确的错误处理：
   - 设置负缓存（暂时存储错误响应），以避免在服务中断期间因重复请求而使源站不堪重负。负缓存意味着 CDN 将临时存储错误响应（例如 404 或 500 错误），以防止重复请求不存在或暂时不可用的内容。
   - 配置适当的错误响应代码和重试行为
4. 实现中间缓存（原点盾牌）：

   Origin shield 是一项在 CDN 边缘站点和您的源服务器之间创建额外缓存层的功能。这减少了到达您的源服务器的冗余请求的数量。
   - 在边缘位置和原点之间配置中间缓存层
   - 减少缓存未命中时向源站发出的冗余请求的数量
   - 提高整个 CDN 基础设施的缓存命中率

高级 CDN 功能适用于 MediaTailor

实现基本内容分发网络 (CDN) 配置后，探索这些高级功能以进一步增强 AWS Elemental MediaTailor 流媒体平台的性能和可靠性。

媒体质量感知路由 (MQAR)

MQAR 是 Amazon CloudFront 的一项功能，可根据实时网络性能指标自动选择最高质量的内容来源。MQAR 不是使用固定源服务器，而是根据延迟和吞吐量等因素将请求动态路由到性能最好的源。这有助于确保即使在网络波动期间，观众也能获得尽可能高质量的直播。

如果您使用 Amazon CloudFront，请实施 MQAR 以根据实时指标自动选择最高质量的内容来源。有关详细信息，请参阅CloudFront 集成 CloudFront 集成部分中的。

清单筛选

Manifest 筛选允许您自定义 MediaTailor 交付给观众的清单中包含哪些格式副本（相同内容的不同质量版本）。筛选功能可删除不适合某些设备或网络条件的格式副本，从而帮助优化带宽使用率。例如，您可以删除移动设备或低带宽连接的 4K 格式副本。

有关使用实现清单筛选的详细信息 AWS Elemental MediaPackage，请参阅MediaPackage 内容分发网络集成。

多 CDN 策略

多 CDN 策略同时使用多个 CDN 提供商来提高可靠性和性能。如果一个 CDN 遇到问题，流量可以自动转移到另一个提供商。这种方法对于可靠性至关重要的备受瞩目的现场活动特别有价值。

有关使用实现多 CDN 策略的信息 MediaTailor，请参阅。规划 CDN 集成

监控 MediaTailor CDN 的运营和性能

对您的 AWS Elemental MediaTailor 和内容分发网络 (CDN) 的集成进行有效监控，可确保可靠的内容交付、最佳性能和快速的问题检测。这种监控方法适用于所有 MediaTailor 实现，包括服务器端广告插入 (SSAI)、渠道组装和组合工作流程。

监控您的 CDN 集成使您能够：

- 在问题影响观众之前检测并解决问题
- 跟踪关键绩效指标并保持服务质量
- 保持最佳缓存性能并减少源负载
- 确保广告插入成功率符合业务要求

要对监控数据中可能出现的参数相关问题进行故障排除，请参阅。MediaTailor 参数疑难解答指南有关监控查询参数使用情况的信息，请参见MediaTailor 清单查询参数。

基本 CDN 性能指标

跟踪以下核心指标，以评估您的 CDN MediaTailor 实施的有效性：

缓存命中率

它衡量的是什么：来自 CDN 缓存的请求与来自来源的请求的百分比。

目标值：

- 内容分段：95% 或更高的缓存命中率
- 广告细分：90% 或更高的缓存命中率
- 清单：因实现而异（不应缓存个性化清单）

为何重要：较高的缓存命中率可减少源站负载、缩短响应时间并降低带宽成本。

原始请求量

它衡量什么：到达您的 MediaTailor 源服务器的请求数量。

目标模式：应保持低位和稳定，偶尔会出现缓存失误或新内容的峰值。

为何重要：源请求量过高表明缓存效率低下，可能会影响 MediaTailor性能。

响应延迟

它衡量什么：CDN 响应观众请求的速度。

目标值：

- 缓存内容：小于 100 毫秒 (P95)
- 源站请求：小于 500 毫秒 (P95)
- 清单请求：小于 100 毫秒 (P95)
- 分段请求：小于 50 毫秒 (P95)

为何重要：低延迟可确保流畅的播放和良好的观众体验。

错误率

它衡量什么：导致 HTTP 错误的请求所占的百分比。

目标值：

- 4xx 错误：少于请求总数的 0.1%
- 5xx 错误：少于请求总数的 0.01%
- 源站错误：低于 0.05% 的源站请求

为何重要：错误率高表示配置问题或服务问题会影响观看者体验。

MediaTailor 特定指标

将这些 MediaTailor 指标与 CDN 指标一起监控，以获得全面的可见性：

广告填充率

关键指标：`Avail.FillRate`和 `AdDecisionServer.FillRate`

目标值：两个指标均高于 90%

为何重要：直接影响广告收入和观众体验

清单生成性能

关键指标：GetManifest.Latency和 GetManifest.Errors

目标值：延迟低于 200 毫秒，错误率低于 1%

为何重要：影响播放启动时间和可靠性

广告决策服务器运行状况

关键指标：AdDecisionServer.LatencyAdDecisionServer.Errors、和 AdDecisionServer.Timeouts

目标值：延迟低于 1000 毫秒、错误率低于 5%、最小超时时间

为何重要：广告效果直接影响广告插入成功率

设置监控工具

配置以下工具以有效监控您 MediaTailor 和 CDN 的集成：

亚马逊 CloudWatch 集成

Amazon 为监控您 MediaTailor 和 CDN 的集成 CloudWatch 提供了基础：

MediaTailor 指标

MediaTailor 自动向 CloudWatch 其发布跟踪请求、响应和错误的指标。关键指标包括：

- RequestCount: 对的请求总数 MediaTailor
- ResponseTime: MediaTailor 响应延迟
- 4xxErrorCount和5xxErrorCount：错误跟踪

有关 MediaTailor 指标的完整列表，请参阅监控 AWS Elemental MediaTailor 使用亚马逊 CloudWatch 指标。

CDN 指标

启用 CDN 指标收集， CloudWatch 包括：

- 不同内容类型的缓存命中率
- Origin 请求计数和响应时间
- 按状态码划分的错误率

对于 CloudFront 特定指标，请在分发设置中启用详细监控。

控制面板配置

创建仪表板，让您可以了解您 MediaTailor 和 CDN 的表现：

1. 创建统一的仪表板：将指标 MediaTailor和 CDN 指标合并到一个 CloudWatch 仪表板中，以实现全面的可见性。
2. 按工作流程整理：按实现类型（SSAI、渠道集合或组合工作流程）对指标进行分组。
3. 包括关键绩效指标：
   - 一段时间内的缓存命中率趋势
   - 响应延迟百分位数（P50、P95、P99）
   - 错误率趋势和峰值

有关详细的仪表板创建指南，请参阅CloudWatch仪表板。

警报配置

配置警报以在问题影响查看者之前对其进行检测：

关键警报

为严重问题配置即时警报：

- 错误率高：在 5 分钟内 5xx 错误超过 0.1% 的请求时发出警报
- 源服务器问题：源站响应时间超过 2 秒时发出警报
- 缓存命中率下降：当清单的缓存命中率低于 70% 或区段的缓存命中率低于 85% 时发出警报

警告警报

配置性能下降的预警警报：

- 延迟增加：当 P95 响应时间超过 200 毫秒时发出警报
- 缓存效率下降：当分段的缓存命中率降至 90% 以下时发出警报

实施清单

使用此清单来确保全面的监控覆盖范围：

1. 指标收集：
   - ✓ 中启用了 MediaTailor 指标 CloudWatch
   - ✓ 已启用 CDN 详细监控
2. 仪表板设置：
   - ✓ 创建了统一的 CloudWatch 仪表板
   - ✓ 在适当的时间范围内可视化关键指标
   - ✓ 与相关团队共享仪表板
3. 警报配置：
   - ✓ 配置有即时通知功能的严重警报
   - ✓ 设置警告警报以便及早发现
   - ✓ 警报升级程序已记录在案
4. 操作程序：
   - ✓ 记录事故响应程序
   - ✓ 制定了定期审查时间表
   - ✓ 团队训练已完成

相关主题

有关特定监控场景的更多指导：

- 性能优化：有关基于监控数据的详细优化技术，请参阅CDN 性能优化。
- 故障排除：有关使用监控数据的详细故障排除程序，请参阅特定于工作流程的故障排除文档。
- 日志分析：有关全面的日志分析和监控，请参阅CDN 集成日志分析和错误代码参考 MediaTailor。

CDN 和集成的测试和 MediaTailor 验证

AWS Elemental MediaTailor 内容分发网络 (CDN) 集成需要进行全面测试，以确保可靠的广告投放和最佳性能。系统测试有助于在集成问题影响观看者之前将其识别出来。使用这种全面的方法来验证您的内容交付网络以及跨不同场景、配置和负载条件的 MediaTailor 集成。

相关话题：

- 有关包括监控和日志记录在内的操作设置，请参见 内容分发网络监控
- 要在出现问题时进行故障排除，请参阅 对 CDN 集成进行故障排除

- 有关性能优化指南，请参阅 CDN 和 MediaTailor 集成的性能优化指南

主题

- 测试 CDN 和集成的先决条件和 MediaTailor 设置
- CDN 和 MediaTailor 集成的系统测试方法
- CDN 和集成的部署前测试清单 MediaTailor
- CDN 和集成的测试工具和实用程序参考 MediaTailor

测试 CDN 和集成的先决条件和 MediaTailor 设置

AWS Elemental MediaTailor 内容分发网络 (CDN) 集成测试需要适当的准备和资源分配。在开始系统测试之前，请确保您拥有必要的资源和工具。

所需资源：

- 测试配置为镜像生产设置的 CDN 分发
- 使用已知内容和广告来源测试 MediaTailor 配置
- 测试具有可预测特征的内容（时长、格式、广告插播标记）
- 测试广告决策服务器或模拟 ADS 响应
- 多种测试设备和玩家类型

测试工具：

- `curl`用于 HTTP 请求测试
- `ffprobe`用于 HLS 清单验证
- `mp4box`用于 DASH 清单验证
- 用于网络分析的浏览器开发者工具
- 用于 end-to-end测试的视频播放器

CDN 和 MediaTailor 集成的系统测试方法

AWS Elemental MediaTailor 内容交付网络 (CDN) 集成测试应遵循系统的分阶段方法，以确保全面覆盖。按照这种结构化方法全面测试您的内容交付网络和 MediaTailor 集成。每个阶段都建立在前一个阶段的基础上，以隔离潜在的问题。

有关系统测试方法的更多指导，请参阅 Well-Architecte AWS d Framework 中的可靠性测试。

第 1 阶段：测试直接 MediaTailor 连接

首先在没有 CDN 参与的情况下测试 MediaTailor 功能以建立基准。

1. 直接测试向 MediaTailor 端点发送的清单请求：
   - 测试 HLS 多变体播放列表请求：`curl -v "https://your-emt-endpoint.mediatailor.region.amazonaws.com/v1/master/hls/config-name/master.m3u8"`
   - 测试 DASH MPD 请求：`curl -v "https://your-emt-endpoint.mediatailor.region.amazonaws.com/v1/dash/config-name/manifest.mpd"`
   - 验证清单响应是否包含预期的广告插播标记
   - 检查该分段 URLs 点以纠正原点
2. 验证广告插入是否正常：
   - 使用不同的广告定位参数进行测试
   - 验证广告区段已正确转码且可用
   - 查看广告中断时间和持续时间
   - 测试广告不可用时的回退行为
3. 衡量基准性能：
   - 记录清单请求响应时间
   - 衡量广告决策服务器的响应时间
   - 测试会话的创建和管理

成功标准：所有直接 MediaTailor 请求都会返回 HTTP 200 响应，其清单格式正确，其中包含预期的广告内容。

第 2 阶段：测试基本 CDN 集成

将 CDN 添加到请求路径并测试基本功能。

1. 通过 CDN 测试清单请求：
   - 使用简单的路由规则配置 CDN
   - 通过 CDN 端点测试清单请求

- 验证 CDN 是否正确地将请求转发到 MediaTailor
- 检查清单响应是否未被缓存 (TTL = 0)

2. 测试段路由：
   - 验证内容分段到源服务器的路由
   - 验证广告细分到 MediaTailor 广告存储的路径
   - 测试区段缓存行为
3. 比较 CDN 与直接性能：
   - 测量响应时间差异
   - 检查回复中是否存在任何内容差异
   - 验证错误处理是否正常

成功标准：CDN 应成功代理发往 MediaTailor 和源服务器的请求，同时对性能的影响最小。

第 3 阶段：测试查询参数转发

添加查询参数转发并测试广告个性化。

1. 在 CDN 上配置查询参数转发：
   - 启用将所有查询参数转发到 MediaTailor
   - 测试会话初始化（会话 ID 由 MediaTailor 在第一次请求时自动生成）
   - 使用自定义定位参数进行测试
2. 测试广告个性化：
   - 验证不同的参数会产生不同的广告响应
   - 测试参数编码和特殊字符
   - 检查参数是否正确传递给 ADS
3. 验证会话管理：
   - 测试会话的创建和保持
   - 验证请求间的会话 ID 一致性
   - 测试会话过期处理

成功标准：广告内容因查询参数而异，并且会话管理得当。

第 4 阶段：测试标头转发

在 CDN 中添加标头转发并测试特定于设备的定位。

1. 为所有标头配置标头转发。有关最低要求，请参阅MediaTailor CDN 集成所需的标头。
2. 测试设备定位：
   - 使用不同的用户代理字符串（手机、台式机、电视）进行测试
   - 验证特定设备的广告响应
   - 使用不同的 IP 地址测试地理定位
3. 验证 CORS 的处理方式：
   - 测试 CORS 标头的网络播放器兼容性
   - 验证印前检查选项请求是否正常运行
   - 从不同的领域进行测试

成功标准：设备和地理位置定位应能正常运行，网络玩家不应遇到 CORS 错误。

第 5 阶段：全面的场景测试

在多个场景中进行测试，以确保稳健运行。

1. 使用不同的玩家类型进行测试：
   - 网络浏览器中的 HLS.js 玩家
   - Video.js 支持 HLS 和 DASH 的玩家
   - 移动设备上的原生玩家
   - 智能电视和机顶盒播放器
2. 在不同的设备和平台上进行测试：
   - 移动设备（iOS、安卓）
   - 桌面浏览器（Chrome、火狐浏览器、Safari、Edge）
   - 智能 TVs 和流媒体设备
   - 不同的操作系统版本
3. 测试不同的内容类型：
   - 直播内容
   - 视频点播 (VOD) 内容
   - 不同的视频格式和比特率

- 具有不同广告插播模式的内容

4. 测试广告定位场景：
   - 不同的受众特征定位参数
   - 跨不同地区的地理定位
   - 基于时间的定位（一天中的不同时间）
   - 自定义业务逻辑参数

成功标准：所有玩家和设备组合都应在适当的广告定位下正常运行。

第 6 阶段：负载和性能测试

在实际负载条件下验证性能。

> **Important**
>
> 在进行@@ 负载测试之前，请联系 AWS S upport：在进行负载和性能测试之前，请创建 AWS 支持票证，将您计划的测试通知 MediaTailor 服务团队。这可确保：
>
> - 该服务已准备好满足您的预期负载水平
> - 在测试时段内有适当的容量可用
> - 您的测试不会被误认为是生产事件
> - 您将获得有关测试最佳实践和限制的指导
>
> 在您的支持请求中包括：预期的并发用户数、测试持续时间、地理区域以及您计划测试的任何特定场景。

1. 测试并发用户场景：
   - 模拟多个并发观看者
   - 测试 CDN 扩展和缓存性能
   - 监控负载下的源服务器性能
   - 验证是否 MediaTailor 可以处理并发会话
2. 衡量绩效指标：
   - 监控负载下的响应时间

- 检查缓存命中率是否符合预期（热门内容的命中率> 80%）
- 测量不同场景下第一帧的时间
- 跟踪负载峰值期间的错误率

3. 测试故障转移场景：
   - 在 ADS 不可用时测试行为
   - 测试源服务器故障转移
   - 验证错误处理和恢复
   - 测试 CDN 边缘位置故障转移

成功标准：系统应在预期负载下保持可接受的性能，并在故障期间正常降级。在执行之前，请务必联系 AWS Supp ort，他们会批准您的负载测试计划。

CDN 和集成的部署前测试清单 MediaTailor

AWS Elemental MediaTailor 在生产部署之前，内容分发网络 (CDN) 集成必须通过全面测试。在将配置更改部署到生产环境之前，请使用此清单。

基本功能：

- □ 清单请求返回 HTTP 200 响应
- □ 内容分段正确加载
- □ 广告区段正确加载
- □ 广告插播时间在预期的时间出现
- □ 播放内容和广告之间平滑过渡

配置验证：

- □ 查询参数已正确转发
- □ 必填标题已正确转发
- □ 清单缓存已禁用 (TTL = 0)
- □ 区段缓存配置得当
- □ CORS 标题是为网络播放器配置的

跨平台测试：

- □ 已在移动设备上测试
- □ 已在桌面浏览器上测试
- □ 针对不同的玩家类型进行了测试
- □ 测试了 HLS 和 DASH 格式

性能验证：

- □ AWS 为负载测试批准而创建的 Support 票证
- □ 响应时间达到绩效目标
- □ 缓存命中率是可以接受的
- □ 错误率在可接受的范围内
- □ 已配置监控和警报

CDN 和集成的测试工具和实用程序参考 MediaTailor

AWS Elemental MediaTailor 内容分发网络 (CDN) 集成测试需要各种工具来进行全面的验证和调试。内容分发网络和 MediaTailor 集成测试中常用工具的参考指南。

curl-HTTP 请求测试

测试清单请求：`curl -v "https://your-cdn-domain.com/v1/master/hls/config/master.m3u8"`

使用标题进行测试：`curl -H "User-Agent: TestAgent/1.0" "https://your-cdn-domain.com/..."`

使用参数进行测试：`curl "https://your-cdn-domain.com/...?aws.sessionId=test123"`

ffprobe-HLS 清单验证

验证 HLS 语法：`ffprobe -v quiet -print_format json -show_format "https://your-cdn-domain.com/master.m3u8"`

查看区段信息：`ffprobe -v quiet -show_entries packet=pts_time "segment.ts"`

mp4box-DASH 清单验证

验证 DASH MPD：`mp4box -info "https://your-cdn-domain.com/manifest.mpd"`

检查分段时间：`mp4box -info segment.m4s`

浏览器开发者工具

在 “网络” 选项卡中监控网络请求

在 “控制台” 选项卡中检查 CORS 错误

检查 request/response 标题

分析时间和性能指标

其他资源：

- 故障排除 CloudFront-全面的 CDN 故障排除指南
- 提高 CloudFront 缓存命中率-性能优化指南
- 监控 AWS 资源-性能监控最佳实践

对 CDN 集成 MediaTailor 和 CDN 集成问题进行故障排除

这份全面的故障排除指南涵盖了所有 AWS Elemental MediaTailor 实现中常见的内容分发网络 (CDN) 集成问题，包括服务器端广告插入 (SSAI)、渠道组装和集成。 AWS Elemental MediaPackage 当你的 CDN 和 MediaTailor 集成遇到问题时，请使用这种系统的诊断方法快速确定根本原因并实施经过验证的解决方案。

无论您的具体工作流程如何，本指南都适用于所有 MediaTailor CDN 集成。有关特定服务或工作流程的问题，请参阅本指南末尾提及的相关疑难解答部分。

开始之前：准备好以下物品以进行高效的故障排除：

- 演示问题的回放 URLs 示例
- 问题发生时段的 CDN 访问日志
- MediaTailor 配置名称和 AWS 区域
- 玩家类型和版本（例如，HLS.js 1.4.0、Video.js 8.0）
- 出现问题的设备和浏览器信息

相关话题：

- 有关操作设置和故障排除准备工作，请参阅 对 CDN 集成进行故障排除

- 有关日志分析和错误代码参考，请参见 CDN 集成日志分析参考
- 有关升级和获取更多帮助，请参阅 获取 CDN 集成支持

MediaTailor 和 CDN 集成的诊断清单

AWS Elemental MediaTailor 内容分发网络 (CDN) 集成问题可能以多种方式表现出来。使用此清单快速确定您遇到的问题类型：

1. 该问题是否影响了所有观众或特定观众？
   - 所有观众 → 可能有 CDN 或 MediaTailor 配置问题
   - 特定观众 → 可能存在个性化或定位问题
2. 清单是否正确加载？
   - 否 → CDN 路由或 MediaTailor 连接问题
   - 是的，但是内容不对 → 缓存或个性化问题
3. 区段加载是否正确？
   - 内容分段失败 → Origin 连接问题
   - 广告区段失败 → 广告投放或转码问题

4. 广告的插入是否正确？
   - 未显示任何广告 → 检查 ADS 连接和配置
   - 出现错误的广告 → 检查广告定位参数和个性化设置
   - 广告无法播放 → 查看广告转码和区段可用性
5. 播放是否流畅且不间断？
   - 缓冲问题 → 检查 CDN 缓存性能和源站响应时间
   - 播放错误 → 检查清单语法和片段可用性
   - 广告过渡问题 → 检查广告中断时间和区段对齐情况
6. 是否有特定的错误代码或消息？
   - HTTP 4xx 错误 → 检查 CDN 的路由和配置
   - HTTP 5xx 错误 → 检查源服务器和 MediaTailor 服务运行状况
   - 玩家特定的错误 → 检查清单格式和玩家兼容性

基于您的诊断的后续步骤：

CDN 配置问题

有关 CDN 路由和缓存故障排除的详细信息，请参阅对 CDN 集成 MediaTailor 和 CDN 集成问题进行故障排除。

清单和播放问题

有关清单验证和播放疑难解答，请参阅CDN 集成测试程序。

广告插入和定位问题

有关特定于广告的故障排除（包括 ADS 连接和广告投放），请参阅工作流程特定的疑难解答文档。

性能和监控问题

有关性能分析和监控设置，请参见监控 MediaTailor CDN 的运营和性能。

日志分析和错误代码

有关详细的日志分析和错误代码参考，请参阅CDN 集成日志分析和错误代码参考 MediaTailor。

测试和验证

有关全面的测试程序，请参阅CDN 和集成的测试和 MediaTailor 验证。

如果您需要即时帮助或无法使用链接的资源解决问题，获取 CDN 和集成的支持和 MediaTailor 故障排除帮助请参阅，了解上报程序。

CDN 集成测试程序

在将 MediaTailor CDN 集成部署到生产环境之前，必须进行适当的测试。这些测试程序有助于识别不同设备和平台之间的配置问题、性能问题和兼容性问题。

基本集成验证

执行以下基本测试以验证您的 CDN 集成是否正常运行：

1. 测试清单交付：
   - 通过您的 CDN 请求清单并验证它是否返回了有效的响应
   - 验证清单是否包含预期的内容和广告插入点
   - 检查清单是否 URLs 使用你的 CDN 域名，而不是来源

- 使用 HLS 或 DASH 验证工具验证清单语法

2. 验证网址重写：
   - 检查清单 URLs 中的内容段是否指向你的 CDN 域名
   - 验证广告区段 URLs 指向你的 CDN 域名
   - 确保所有亲属 URLs 都得到正确解析
3. 测试内容播放：
   - 通过视频播放器播放内容并验证播放流畅
   - 验证内容和广告都能畅通无阻地播放
   - 检查内容和广告之间的过渡是否正确
   - 测试搜索和擦洗功能
4. 验证 CDN 路由：
   - 监控 CDN 访问日志，确保请求路由正确
   - 验证缓存 hit/miss 模式是否符合预期
   - 检查是否只有在缓存未命中时才会发出源请求

高级测试程序

执行以下额外测试以进行全面验证：

1. 跨平台兼容性测试：
   - 在多台设备（台式机、手机、平板电脑、智能电视）上进行测试
   - 验证不同浏览器的兼容性
   - 使用各种视频播放器（HLS.js、Video.js、原生播放器）进行测试
   - 在不同的操作系统上进行验证
2. 性能测试：
   - 衡量清单请求的响应时间（目标：缓存时间小于 100 毫秒）
   - 测试不同比特率下的片段加载性能
   - 验证启动时间是否符合性能目标
   - 在各种网络条件下进行测试
3. 广告跟踪验证：
   - 验证广告跟踪信标是否正确触发
   - 检查广告分析数据的准确性

- 测试印象和完成跟踪
- 验证点击功能

4. 错误状态测试：
   - 测试源暂时不可用时的行为
   - 验证是否对格式错误的请求进行优雅处理
   - 测试 CDN 故障转移场景
   - 验证错误消息的清晰度和实用性

创建测试环境

设置一个反映您的生产配置的测试环境以进行全面验证：

1. 设置单独的 CDN 发行版进行测试：
   - 创建与生产版本具有相同缓存行为的测试 CDN 分发
   - 配置反映您的生产设置的测试来源
   - 使用单独的域名以避免与生产流量发生冲突
2. 创建测试 MediaTailor 配置：
   - 使用与制作相同的设置来设置测试回放配置
   - 配置测试广告决策服务器端点
   - 使用与您的制作广告格式相匹配的测试广告内容
3. 实施系统的测试流程：
   - 为配置更改创建测试清单
   - 为您的团队记录测试程序
   - 尽可能设置自动测试

在多个场景中进行测试

在不同的场景和条件下验证您的集成，以确保全面覆盖：

1. 使用多种玩家类型和设备进行测试：
   - 使用不同的视频播放器（网络、手机、联网电视）进行测试
   - 在不同的操作系统和浏览器上进行验证
   - 测试各种网络条件和连接速度

2. 创建自动测试脚本：
   - 自动验证清单请求
   - 创建脚本以测试广告插入场景
   - 针对高流量场景实施性能测试
3. 验证广告定位和个性化：
   - 使用不同的用户资料和定位参数进行测试
   - 验证广告决策服务器集成
   - 在广告不可用时测试备用场景

测试工具和技术

使用以下工具和技术进行有效的测试：

浏览器开发者工具

使用 “网络” 选项卡检查 HTTP 请求和响应

监控控制台 JavaScript 是否存在错误和警告

验证响应标头和缓存行为

查看时序信息以进行性能分析

命令行测试

使用 curl 来测试特定的标题 URLs 并检查标题：

```
curl -I "https://your-cdn-domain.com/path/to/manifest.m3u8"
```

使用 wget 进行下载测试和时序分析

使用像 httpie 这样的工具进行更具可读性的 HTTP 测试

视频播放器测试

使用多个玩家实现进行测试

使用玩家调试模式检查内部行为

监控玩家事件和错误回调

验证自适应比特率切换行为

CDN 分析和监控

在测试期间监控实时 CDN 指标

查看访问日志以了解请求模式

如果有 CDN 特定的测试工具，请使用

为测试验证设置临时警报

有关其他全面的测试方法和系统验证方法，请参阅CDN 和集成的测试和 MediaTailor 验证。

对 CDN 清单 404 错误进行故障排除 MediaTailor

AWS Elemental MediaTailor 内容分发网络 (CDN) 清单 404 错误是一个常见的集成问题，会导致无法开始播放。本节提供清单交付失败的 step-by-step疑难解答。

多变体播放列表、媒体播放列表或 MPD 请求返回 404 错误

快速修复（先试）：

1. 验证您的 URL 中的 MediaTailor 配置名称是否完全匹配（区分大小写）
2. 在 MediaTailor 没有 CDN 的情况下直接测试清单网址：`curl -v "https://your-emt-endpoint.mediatailor.region.amazonaws.com/v1/master/hls/config-name/master.m3u8"`
3. 如果直接测试有效，请检查清单请求的 CDN 路由规则

如果快速修复不起作用：

症状：玩家无法开始播放，清单请求在 CDN 日志中返回 HTTP 404 错误。

错误消息示例：

- 浏览器控制台：`"Failed to load resource: the server responded with a status of 404 (Not Found)"`
- 玩家错误：`"MANIFEST_LOAD_ERROR"`或 `"NETWORK_ERROR"`
- CDN 日志：`GET /v1/master/hls/example-config/master.m3u8 404`

解决方法：

确认您的 CDN 路由规则配置正确，可以将多变体播放列表、媒体播放列表和 MPD 请求转发到。MediaTailor

检查 MediaTailor 配置是否存在且设置正确。

确保您的 CDN 行为模式与预期的清单请求路径相匹配（例如`*.m3u8`，`*.mpd`）。

诊断 CDN 清单的交付问题和错误 MediaTailor

AWS Elemental MediaTailor 内容分发网络 (CDN) 清单投放问题可能会妨碍广告的正确插入和播放。如果观众在多变播放列表、媒体播放列表中收到不正确或不一致的广告，或者 MPDs：

1. 检查缓存的清单：
   - 验证所有多变体播放列表、媒体播放列表和 MPD 路径的 TTL 设置均设置为 0
   - 确认您的 CDN 没有缓存多变体播放列表、媒体播放列表，或者没有缓存 TTL MPDs 设置
   - 查看 CDN 日志以了解缓存状态 — 清单请求应显示`Miss`或`RefreshHit`，不是 `Hit`
2. 验证 CDN 路由配置：
   - 确认清单请求正在路由到 MediaTailor 终端节点，而不是缓存或从源站提供服务
   - 检查 CDN 行为模式是否正确匹配清单路径（*.m3u8、*.mpd）
   - 验证查询参数是否被转发到以 MediaTailor 进行个性化设置
   - URLs 直接针对清单进行测试 MediaTailor 以隔离 CDN 和服务问题
3. 检查标题转发配置：
   - 验证是否正在转发所需的标头（请参阅MediaTailor CDN 集成所需的标头）
   - 确认已转发 User-Agent 标头以用于特定设备的广告定位
   - 检查 X-Forwarded-For标题是否已转发以进行地理定位
   - 确保已转发 “接受编码” 标头以支持压缩
4. 验证清单内容和结构：
   - 检查清单中是否包含预期的广告插入标记（HLS 为 EXT-X-CUE-OUT/IN）
   - 验证清单 URLs 中的区段使用您的 CDN 域名，而不是源域名
   - 确认广告区段已正确插入且可访问
   - 使用验证工具测试清单语法（HLS 为 ffprobe，DASH 为 mp4box）
5. 测试不同的场景：
   - 使用不同的会话进行测试 IDs 以验证个性化设置是否有效

- 从不同的地理位置进行测试以验证地理位置定位
- 使用不同的 User-Agent 字符串进行测试以验证设备定位
- 比较带有 CDN 和不使用 CDN 的清单响应，以找出差异

其他疑难解答资源：

- 有关 CDN 缓存配置的详细信息，请参阅 针对 CDN 和 MediaTailor 集成的缓存优化
- 有关全面的 CDN 路由设置，请参阅 为以下各项设置 CDN 路由行为 MediaTailor
- 有关标头转发的要求，请参阅 MediaTailor CDN 集成所需的标头
- 有关日志分析和错误代码，请参见 CDN 集成日志分析和错误代码参考 MediaTailor
- 有关测试程序和验证的信息，请参见 CDN 和集成的测试和 MediaTailor 验证

成功标准：解决后，玩家应正常开始播放，广告应按预期显示。清单请求应在 CDN 日志中返回 HTTP 200 状态码，并且清单应包含经过适当个性化的广告内容。

解决 CDN 分段交付和加载问题 MediaTailor

AWS Elemental MediaTailor 内容分发网络 (CDN) 分段交付问题可能会导致缓冲和播放中断。如果玩家无法加载分段或体验缓冲：

1. 查看 CDN 路由规则：
   - 验证内容分段是否被路由到正确的来源
   - 确认广告区段已路由到正确的 MediaTailor 广告存储位置
   - 检查分段文件扩展名是否与您的 CDN 行为模式相匹配
   - 验证清单 URLs 中的分段是否使用正确的 CDN 域
   - 有关设置路由和行为路径模式的详细说明，请参阅 为以下各项设置 CDN 路由行为 MediaTailor
2. 验证 CORS 配置：
   - 对于网络玩家，请确保您的 CDN 已通过或正确设置 CORS 标头
   - 使用浏览器开发者工具进行测试以识别与 CORS 相关的错误
   - 验证印前检查选项请求是否得到正确处理
3. 测试区段的可访问性和性能：
   - URLs 直接测试各个区段以验证其是否可访问
   - 检查分段响应时间并确定性能瓶颈

- 验证分段文件大小是否适合您的带宽目标
- 从不同的地理位置加载测试区段

4. 验证分段的 CDN 缓存行为：
   - 验证内容分段是否具有适当的 TTL 设置（通常比清单长）
   - 检查是否根据个性化要求适当缓存了广告区段
   - 监控内容和广告细分的缓存命中率
   - 确保缓存密钥不包含会降低缓存效率的不必要参数
5. 检查源服务器的连接和运行状况：
   - 验证源服务器对分段请求的响应是否正确
   - 检查负载下的源服务器容量和响应时间
   - 验证源服务器是否有预期的分段文件可用
   - 如果配置了多个源站，则测试源故障转移场景
6. 解决特定于广告细分的问题：
   - 验证广告区段是否已正确转码且可用于 MediaTailor
   - 检查广告区段 URLs 是否在清单中正确生成
   - 使用不同的广告定位参数测试广告细分加载情况
   - 监控可能导致区段不可用的广告转码延迟
7. 验证玩家的兼容性和行为：
   - 使用不同的玩家类型和版本加载测试片段
   - 检查玩家缓冲区设置和分段请求模式
   - 验证玩家对失败的分段请求的错误处理
   - 测试自适应比特率切换和分段选择逻辑

其他疑难解答资源：

- 有关 CDN 路由和行为配置，请参阅 为以下各项设置 CDN 路由行为 MediaTailor
- 有关 CDN 缓存优化的信息，请参阅 针对 CDN 和 MediaTailor 集成的缓存优化
- 有关 CORS 配置指南，请参阅 CDN 集成安全最佳实践 MediaTailor
- 有关性能监控和分析的信息，请参见 监控 MediaTailor CDN 的运营和性能
- 有关全面的测试程序，请参阅 CDN 和集成的测试和 MediaTailor 验证
- 有关日志分析和错误诊断，请参见 CDN 集成日志分析和错误代码参考 MediaTailor

成功标准：解决后，玩家应顺利加载片段，而不会出现缓冲中断。细分请求应返回带有适当响应时间的 HTTP 200 状态码，并且内容和广告区段都应可访问并正确缓存。

修复 CDN 会话管理和跟踪问题 MediaTailor

AWS Elemental MediaTailor 内容分发网络 (CDN) 会话管理对于适当的广告个性化和跟踪至关重要。如果您遇到与会话相关的错误或请求中的行为不一致：

1. 检查会话 ID 的一致性：
   - 确认您的播放器在单个播放会话的所有请求中都保持相同的会话 ID
   - 检查 CDN 日志以确认会话 IDs 已正确转发
   - 确保 IDs 在查询参数中对会话进行了正确的 URL 编码
   - 使用 CloudWatch 日志验证请求间的会话 ID 一致性（参见下面的验证步骤）
2. 验证会话初始化：
   - 确认第一个清单请求成功创建了会话
   - 检查会话参数转发是否正确（例如，`aws.sessionId`）
   - 使用调试日志验证会话初始化（参见下面的调试日志设置）
3. 启用调试日志以进行详细的会话故障排除：
   - 对于服务器端报告：`?aws.logMode=DEBUG`添加到您的播放请求中：

     ```
     GET <mediatailorURL>/v1/master/<hashed-account-id>/<origin-id>/<asset-id>?
     aws.logMode=DEBUG
     ```

   - 对于客户端报告：包含`"logMode": "DEBUG"`在会话初始化请求正文中
   - 重要：该`DEBUG`值区分大小写
   - 最多允许 10 个活动调试会话同时进行
4. 使用 CloudWatch 日志查询来验证会话行为：
   - 验证调试会话是否处于活动状态：

     ```
     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
     ```

   - 查看会话的所有活动：

```
fields @timestamp, @message, eventType, mediaTailorPath
| filter sessionId = "your-session-id-here"
| sort @timestamp asc
```

- 检查会话的清单生成：

```
fields @timestamp, responseBody, @message
| filter mediaTailorPath like "/v1/master/" and eventType = "GENERATED_MANIFEST"
 and sessionId = "your-session-id-here"
```

5. 通过 CDN 转发测试会话参数：
   - 直接针对会话参数测试清单请求 MediaTailor（绕过 CDN）
   - 比较使用 CDN 和不使用 CDN 的会话行为，以确定转发问题
   - 验证 CDN 查询参数转发配置包括会话相关参数
   - 检查 CDN 是否未缓存本应特定于会话的响应

常见的会话错误消息：

- `ConflictException`(HTTP 409)-同一会话的多个同步播放列表请求。解决方案：确保您的玩家按照 HLS 规范逐个请求播放列表
- `NotFoundException`(HTTP 404)-会话不可用或配置不存在。解决方案：检查您的配置有效性并重新初始化会话
- `BadRequestException`(HTTP 400)-会话 ID 无效或请求格式不正确。解决方案：验证请求格式和会话 ID 的有效性

其他疑难解答资源：

- 有关完整的调试日志记录设置和字段参考，请参阅 正在生成 AWS Elemental MediaTailor 调试日志
- 有关 CloudWatch 日志查询示例和日志分析，请参阅 编写 AWS Elemental MediaTailor 直接登录到亚马逊 CloudWatch 日志
- 有关 CDN 查询参数转发配置，请参阅 为以下各项设置 CDN 路由行为 MediaTailor
- 有关全面的错误代码参考，请参阅 从播放问题进行故障排除 MediaTailor

成功标准：解决后，会话应正确初始化， IDs 在请求之间保持会话的一致性，调试日志应显示正确`SESSION_INITIALIZED`的事件和清单生成而不会出现错误。

解决 CDN 广告中断时间和同步问题 MediaTailor

AWS Elemental MediaTailor 内容分发网络 (CDN) 广告中断时间必须与内容标记精确同步。如果广告的展示时间不正确或广告中断时间不一致：

1. 验证内容中的广告插播标记：
   - 检查 SCTE-35 标记是否正确放置在您的原始内容中
   - 验证广告时长是否与实际广告内容长度相符
   - 确认广告中断时间与内容界限一致
   - 验证来源清单中的 SCTE-35 标记格式和计时准确性
   - 使用不同的内容类型（直播与视频点播）测试广告插播标记
2. 检查 CDN 缓存对时间的影响：
   - 确保清单 TTL 设置为 0 以防止计时偏差
   - 确认没有缓存对时间敏感的参数
   - 检查内容源和 CDN 之间是否存在时钟同步问题 MediaTailor
   - 监控长时间直播中的时序偏差
   - 验证 CDN 边缘服务器与 NTP 的时间同步
3. 验证 SCTE-35 标记的实现：
   - 验证 EXT-X-DATERANGE标签是否包含正确的 SCTE35出局和持续时间规格
   - 使用明确的提示计时 SCTE35时，请检查是否有配对的出局和 SCTE35输入标记
   - 验证 START-DATE 时间戳是否与实际内容时间一致
   - 测试不同的 SCTE-35 标记格式（基于持续时间的标记与配对标记）
4. 测试不同场景下的广告中断时间：
   - 将广告中断时间与直接 MediaTailor 请求与 CDN 请求进行比较
   - 测试不同 CDN 边缘位置的计时一致性
   - 根据不同的玩家类型和缓冲行为验证广告中断时间
   - 监控流量高峰时段的计时精度
5. 使用日志和监控调试计时问题：
   - 启用调试日志记录以跟踪广告插播时间处理时间
   - 监控广告插入时序模式的 CloudWatch 指标
   - 查看 CDN 日志，了解与计时相关的请求模式
   - 使用播放器调试工具从客户角度验证广告中断时间

预期的时序容差：

- 广告中断时间应与内容中的 SCTE-35 标记保持一致
- 广告时长应与您的广告决策服务器响应中指定的持续时间一致
- 内容源 MediaTailor、和 CDN 之间的时钟同步应在 1 秒以内
- SCTE-35 标记计时应精确到实际内容计时的 100 毫秒以内

其他疑难解答资源：

- 有关 SCTE-35 标记格式和实现的信息，请参阅 集成内容源以进行 MediaTailor 广告插入
- 有关调试日志记录设置和时序分析，请参见 正在生成 AWS Elemental MediaTailor 调试日志
- 有关 CDN 缓存配置和时序影响，请参阅 针对 CDN 和 MediaTailor 集成的缓存优化
- 有关包括时序验证在内的全面测试程序，请参阅 CDN 和集成的测试和 MediaTailor 验证
- 有关监控广告投放时间和效果的信息，请参阅 监控 MediaTailor CDN 的运营和性能

成功标准：解决后，广告插播时间应精确按照 SCTE-35 标记指定的时间显示，所有 CDN 边缘位置和玩家类型的时机保持一致。调试日志应显示准确的广告插播时间处理时间，而不会出现偏差或同步错误。

优化 CDN 性能并解决延迟问题 MediaTailor

AWS Elemental MediaTailor 内容分发网络 (CDN) 集成性能直接影响观众体验和广告投放质量。如果您遇到响应时间慢或性能下降的情况：

绩效衡量技术

在对性能问题进行故障排除之前，请确定基线测量和持续监控：

1. 衡量关键绩效指标：
   - 响应时间：清单请求应在 200 毫秒内完成，分段请求应在 100 毫秒内完成
   - 缓存命中率：内容区段 > 95%，广告区段 > 90%
   - 原始请求量：缓存优化后，应少于总请求量的 5%
   - 第一帧播放时间：初始播放应在 2-3 秒内开始
2. 使用性能衡量工具：
   - CDN 分析仪表板：监控缓存性能、响应时间和错误率

- CloudWatch 指标：跟踪 MediaTailor服务指标，包括 GetManifest .Latency
- 浏览器开发者工具：衡量客户端性能和网络时序
- 命令行工具：使用带有计时选项的 curl 来衡量特定请求

3. 实施持续监控：
   - 为响应时间下降设置自动性能警报
   - 监控不同地理区域的绩效
   - 在交通高峰时段追踪表现
   - 比较配置更改前后的性能指标

绩效衡量资源：

- 有关全面的性能监控设置，请参见 监控 MediaTailor CDN 的运营和性能
- 有关性能测试程序，请参见 CDN 和集成的测试和 MediaTailor 验证
- 有关 CloudWatch 指标和监控，请参阅 监控 AWS Elemental MediaTailor 使用亚马逊 CloudWatch 指标

CDN 缓存性能问题

缓存性能问题是最常见的 CDN 集成问题之一。这些问题会影响所有 MediaTailor 实现，并可能显著影响观众体验和成本。

缓存命中率低

症状：源站请求量高、延迟增加、带宽成本增加、观看者体验不佳

目标值：

- 内容分段：95% 或更高的缓存命中率
- 广告细分：90% 或更高的缓存命中率
- 清单：因实现而异（不应缓存个性化清单）

常见原因：

- 不同内容类型的 TTL 设置不正确
- 缓存密钥配置包括不必要的查询参数
- 来自来源的缓存控制标头配置不正确
- 频繁的缓存失效或清除

- 地理分布问题（内容未在边缘位置缓存）

解决方案

1. 查看并优化 TTL 设置：
   - 内容分段：将 TTL 设置为与片段持续时间或更长时间相匹配
   - 广告细分：将可重复使用的广告的 TTL 设置为 24 小时或更长时间
   - 静态资产：将 TTL 设置为 24 小时或更长时间

   有关全面的 TTL 建议和缓存优化策略，请参阅针对 CDN 和 MediaTailor 集成的缓存优化。
2. 优化缓存密钥配置：
   - 从缓存密钥中移除不必要的查询参数
   - 确保仅包含影响内容的参数
   - 标准化参数顺序和区分大小写
3. 验证源缓存控制标头是否设置正确
4. 为高流量实现实现实现实现原点屏蔽（或等效的 CDN 功能）。Origin shield 功能在主要版本中可用， CDNs 但可能有不同的名称（例如 O CloudFront rigin Shield、Fastly Shield、Cloudflare Argo 分层缓存）。如果您的 CDN 不提供此功能，则可以在联系 AWS S uppor MediaTailor t 时将其启用。
5. 查看缓存失效策略并减少不必要的清除

验证步骤：

1. 使用 CDN 分析仪表板监控缓存命中率
2. URLs 使用 curl 进行特定测试以验证缓存标头
3. 比较更改前后的源站请求量

源站请求量高

症状：到达 MediaTailor 源站的请求数量出乎意料地高、源服务器负载增加、成本更高

预期模式：当缓存命中率处于最佳状态时，原始请求应少于查看者请求总数的 5%

常见原因：

- 由于 TTL 值较低，缓存丢失
- 缓存密钥碎片（唯一缓存密钥过多）
- 没有缓存内容的区域的地理流量激增
- 频繁的缓存失效

解决方案

1. 分析请求模式以确定缓存丢失的原因
2. 根据内容类型和更新频率优化 TTL 设置
3. 为新内容实施缓存预热策略
4. 考虑 Origin shield 的实现（适用于不同 CDNs 名称的主打——详情Origin Shield 实现请参阅）

警报阈值：当源请求超过请求总数的 10% 或比基线增加 50% 时设置警报

常见的 HTTP 错误解决方法

CDN 集成中的 HTTP 错误通常表示配置问题或服务问题。这些错误模式在所有 MediaTailor实现中都是一致的。

404 未找到错误

症状：清单或分段请求返回 HTTP 404，玩家无法加载内容，玩家日志中返回“MANIFEST_LOAD_ERROR”

常见原因：

- CDN 源配置不正确（ MediaTailor 端点 URL 错误）
- 缓存行为路径模式缺失或不正确
- CDN 配置中的网址重写问题
- MediaTailor 配置名称或播放端点错误
- 直播内容的时间问题（请求 future 片段）

诊断步骤：

1. 直接针对 MediaTailor 来源测试相同的 URL（绕过 CDN）
2. 验证 CDN 源配置是否与 MediaTailor 播放端点匹配
3. 检查 CDN 缓存行为路径模式和优先级
4. 查看 CDN 访问日志，了解请求路由的详细信息
5. 验证 MediaTailor 配置名称和区域设置

解决方案

- 更正 CDN 源配置以匹配 MediaTailor 播放端点
- 更新缓存行为路径模式以正确路由请求
- 修复 URL 重写规则（如果适用）

- 验证 MediaTailor 配置是否存在且处于活动状态

403 禁止的错误

症状：请求返回 HTTP 403、访问被拒绝消息、身份验证失败

常见原因：

- 所需的查询参数缺失或不正确 MediaTailor
- CDN 未转发所需的标头或参数
- IP 地址限制或地理封锁
- 身份验证令牌问题（如果使用已签名 URLs）

解决方案

- 验证所有必需的查询参数均已包含并已转发
- 检查标头和参数转发的 CDN 配置
- 查看 IP 限制和地理设置
- 验证身份验证令牌和签名流程

400 错误请求错误

症状：请求返回 HTTP 400、请求格式错误、参数验证失败

常见原因：

- 查询参数格式错误或 URL 编码问题
- 无效的参数值或格式
- 缺少特定 MediaTailor 功能所需的参数
- 已超出网址长度限制

解决方案

- 验证查询参数格式和 URL 编码
- 根据 MediaTailor API 要求检查参数值
- 确保包含所有必需的参数
- 查看网址长度并考虑参数优化

5xx 服务器错误

症状：请求返回 HTTP 500、502、503 或 504 错误，间歇性服务故障

常见原因：

- MediaTailor 服务问题或容量限制
- CDN 源连接问题
- 源站响应缓慢导致的超时问题
- 服务暂时降级

解决方案

- 检查 S AWS ervice Health Dashboar MediaTailor d 了解服务状态
- 验证 CDN 源连接和超时设置
- 使用指数退避实现重试逻辑
- 监控服务运行状况的 MediaTailor CloudWatch 指标
- 如果问题仍然存在，请联系 Su AWS pport

1. 衡量基准性能：
   - 直接测试清单请求响应时间 MediaTailor （目标：<200 毫秒）
   - 衡量清单请求的 CDN 响应时间（目标：缓存命中率小于 100 毫秒）
   - 查看来自源站和 CDN 的区段加载时间
2. 分析 CDN 的性能：
   - 检查内容细分的缓存命中率（目标为：热门内容的缓存命中率> 80%）
   - 验证源盾牌（或等效的 CDN 功能）是否已在与您的来源相同的 AWS 区域中启用和配置。不同 CDNs 使用不同的名称来实现此功能
   - 监控 CDN 边缘位置性能和地理分布

性能基准：

- 监控清单生成响应时间，并与您的基准性能进行比较
- CDN 缓存命中率明显快于源站请求
- ADS 响应时间不应导致清单生成延迟

其他疑难解答资源：

- 有关全面的性能优化策略，请参阅 CDN 和 MediaTailor 集成的性能优化指南
- 有关 Origin Shield 实现的详细信息，请参阅 Origin Shield 实现
- 有关 CDN 缓存优化的信息，请参阅 针对 CDN 和 MediaTailor 集成的缓存优化

- 有关性能监控和指标，请参见 监控 MediaTailor CDN 的运营和性能
- 有关性能测试程序，请参见 CDN 和集成的测试和 MediaTailor 验证

成功标准：解决后，响应时间应达到目标基准（清单小于 200 毫秒，分段小于 100 毫秒），大多数内容类型的缓存命中率应超过 90%，原始请求量应小于总请求量的 5%。所有地理区域和设备类型的性能都应保持一致。

修复 CDN 跨设备和平台的不一致行为 MediaTailor

AWS Elemental MediaTailor 内容分发网络 (CDN) 集成应在所有设备和平台上提供一致的广告投放。如果广告在不同设备上的行为有所不同：

1. 确保所有 CDN 行为之间一致的标头转发。
   - 验证 User-Agent X-Forwarded-For、和自定义定位标头的转发是否一致
   - 检查标头转发规则是否适用于所有相关的缓存行为
2. 验证播放器与您的 CDN 配置的兼容性。
   - 使用多种玩家类型（HLS.js、Video.js、原生玩家）进行测试，找出特定于玩家的问题
   - 检查玩家特定的标题要求或网址处理差异
3. 使用多种设备类型进行测试，以确定特定于设备的问题。
   - 在测试中包括移动设备、平板电脑 TVs、智能和桌面浏览器
   - 测试不同的操作系统和浏览器版本
   - 验证特定设备的广告定位是否能正常运行

如果您已按照这些故障排除步骤进行操作但仍需要帮助，请参阅获取 CDN 集成支持。

疑难解答准备

设置工具和流程以简化出现 CDN 集成问题时的故障排除。主动准备可以在出现问题时更快、更有效地进行故障排除。

启用全面日志

详细的日志对于诊断 CDN 集成问题至关重要。配置日志记录以捕获故障排除期间所需的信息。

1. 启用详细的 CDN 访问日志：
   - 为所有处理 MediaTailor请求的缓存行为配置日志记录

- 在日志条目中包含查询字符串和自定义标头
- 设置日志分析工具以识别模式和异常情况
- 启用实时日志，以便在直播活动期间立即检测问题
- 配置日志保留策略以维护历史故障排除数据

2. 配置 MediaTailor 日志：
   - 为您的 MediaTailor 配置启用访问日志
   - 设置 CloudWatch 日志组以便集中管理日志
   - 配置日志过滤器以识别错误模式
3. 设置源服务器日志：
   - 在内容来源服务器上启用详细访问日志
   - 在日志中包含请求标头和响应代码
   - 监控源服务器性能指标

添加诊断请求标头

自定义标头有助于跟踪通过 CDN 的请求并识别路由问题。

1. 配置 CDN 诊断标头：
   - 为每个请求添加唯一标识符（例如，`X-Request-ID`）
   - 在请求标头中包含特定于 CDN 的信息
   - 添加边缘位置或 POP（接入点）信息以跟踪地理路由
   - 包括缓存状态标头（命中、未命中、 RefreshHit），用于缓存行为分析
2. 添加用于调试的响应标头：
   - 包括服务器标识标头
   - 为性能分析添加计时信息
   - 包括清单请求的缓存控制标头

建立基准性能指标

记录正常性能范围，以便在故障排除期间快速识别异常：

1. 记录基线指标：
   - 不同内容类型的缓存命中率

- 响应时间百分位数（P50、P95、P99）
- 按状态码划分的错误率
- 按一天中的时间请求音量模式

2. 记录绩效预期：
   - 目标缓存命中率（内容命中率超过 95%，广告命中率超过 90%）
   - 可接受的响应时间（缓存 <100 毫秒，原点 <500 毫秒）
   - 最大可接受错误率（4xx 为 < 1%，5xx 为 < 0.1%）
3. 创建绩效仪表板：设置监控仪表板，显示当前指标与基准值的比较。

准备疑难解答工具

设置有效排除故障所需的工具和访问权限：

1. 命令行工具：
   - `curl`用于测试 HTTP 请求和响应
   - `dig`或者`nslookup`用于 DNS 故障排除
   - 用于清单验证的 HLS/DASH 验证工具
   - 日志分析工具（grep、awk 或专业日志分析器）
2. 访问权限：
   - 访问 CDN 管理控制台以进行配置审查
   - MediaTailor 用于配置验证的控制台访问权限
   - CloudWatch 访问指标和日志分析
   - 用于后端故障排除的源服务器访问权限
3. 文档：
   - 网络架构图
   - CDN 和 MediaTailor 配置文档
   - 上报程序的联系信息
   - 常见场景的运行手册疑难解答

特定于工作流程的疑难解答指南

这份通用故障排除指南涵盖了所有 MediaTailor CDN 集成的常见问题。有关特定工作流程或服务的问题，请查阅以下专门的疑难解答资源：

服务器端广告插入 (SSAI) 疑难解答

有关特定于 SSAI 的问题，包括广告插入失败、广告过渡问题和个性化问题，请参阅特定于工作流程的 SSAI 疑难解答文档。

常见的 SSAI 特定问题：

- 广告插入失败和空广告中断
- 广告过渡时序和同步问题
- 个性化和定位问题
- 广告跟踪和分析差异

频道组装疑难解答

有关特定于渠道汇编的问题，包括清单生成问题和时移功能，请参阅频道汇编工作流程文档。

常见的频道组装问题：

- 清单生成和编译错误
- 时移窗口和 DVR 功能问题
- 源内容可用性和故障转移问题
- 节目时间表和元数据同步

MediaPackage 集成疑难解答

有关 MediaPackage 具体问题，包括清单筛选和 EMP 端点问题，请参阅CDN 集成疑难解答。

常见的 MediaPackage 集成问题：

- 清单筛选参数错误
- MediaPackage 端点连接问题
- EMP 特有的缓存行为问题
- MediaPackage 源站身份验证问题

CloudFront 具体故障排除

有关 CloudFront 具体的配置和设置问题，请参阅 CloudFront集成文档。

常见 CloudFront 问题：

- 分发配置和缓存行为设置
- 源站访问身份和安全配置
- CloudFront 特定的错误代码和响应
- 地理限制和边缘位置问题

其他资源：

- 有关性能优化指南，请参阅 CDN 性能优化
- 有关监控和警报设置，请参阅 内容分发网络监控
- 如需一般支持和帮助，请参阅 获取 CDN 集成支持

CDN 集成日志分析和错误代码参考 MediaTailor

AWS Elemental MediaTailor 内容分发网络 (CDN) 集成日志提供了有关性能和错误的宝贵见解。本指南涵盖了 CDN 日志（来自您的内容分发网络）以及与 CDN 集成故障排除相关的 MediaTailor 日志和错误代码。当您需要了解内容分发网络日志和错误代码告诉您的 MediaTailor 集成信息时，请使用此参考资料。本指南可帮助您解释日志条目和错误消息，以确定问题的根本原因。

相关话题：

- 有关 step-by-step故障排除程序，请参阅 对 CDN 集成进行故障排除
- 有关主动监控和预防策略，请参阅 监控 MediaTailor CDN 的运营和性能
- 有关升级和获取更多帮助，请参阅 获取 CDN 集成支持
- 有关全面的 MediaTailor 日志记录配置和资源，请参阅 MediaTailor 记录配置资源
- 有关 CloudFront 日志格式的参考，请参阅CloudFront 访问日志格式
- 有关 HTTP 状态码的参考，请参阅 HTTP 响应状态码

CDN 日志解释和分析技术 MediaTailor

AWS Elemental MediaTailor 内容分发网络 (CDN) 集成可生成详细日志，帮助您了解请求模式并识别问题。在分析内容分发网络日志时，请重点关注以下关键指标：

注意：以下示例使用 Amazon CloudFront 术语。其他 CDN 提供商可能会对类似的概念使用不同的术语。

HTTP 状态代码

- 200-请求成功
- 404-未找到资源（请检查路由规则和 MediaTailor 配置）
- 403-访问被拒绝（请检查 CDN 安全设置和源站权限）
- 502/503/504-源服务器错误（检查 MediaTailor服务运行状况和源站连接）

缓存行为指示器

- Hit-从 CDN 缓存中提供的内容
- Miss-从来源获取的内容
- RefreshHit-已通过来源验证的缓存内容
- Error-请求导致错误

要监控的请求模式

- 清单请求通常应导致Miss或由RefreshHit于 TTL 设置较低
- Hit对于热门内容，应显示内容区段
- 广告细分可能会Miss因为个性化而显示

服务运行状况监控

分析 CDN 日志和故障排除问题时，请检查 MediaTailor 服务运行状况以确定问题是否与服务有关：

AWS Service Health 仪表板

检查当前 MediaTailor 服务状态和任何正在进行的服务事件

访问权限：S AWS ervice Health Dashb

在以下情况下使用此选项：您在 CDN 日志中看到广泛的 5xx 错误或服务超时

AWS 个人健康 Dashboard

查看账户特定的服务运行状况通知和维护事件

访问权限：AWS 个人健康 Das hboard

在以下情况下使用此功能：您需要账户特定的服务运行状况信息或维护通知

亚马逊 CloudWatch 指标监控

通过指标监控 MediaTailor 服务运行状况 CloudWatch 指标：

- GetManifest.Errors-跟踪清单生成错误

- GetManifest.Latency-监控响应时间性能
- AdDecisionServer.Errors-监控广告服务器连接问题
- Origin.Errors-追踪源服务器连接问题

访问权限：CloudWatch控制台

在以下情况下使用此功能：您需要详细的服务性能指标和历史趋势

服务运行状况疑难解答工作流程：

1. 查看 S AWS ervice Health Dashboard 了解当前服务状态
2. 查看 Pers AWS onal Health Dashboard 以获取账户特定的通知
3. 监控 CloudWatch 服务性能指标的指标
4. 将服务运行状况与 CDN 日志模式和错误率相关联
5. 如果服务运行状况正常，请重点关注 CDN 配置和源服务器问题

CDN 日志分析资源

- CloudFront 访问日志-CDN 日志格式和字段的完整指南
- CloudFront 实时日志-实时日志流和分析
- 使用 CloudWatch Logs Insights 分析日志数据-高级日志查询和分析
- 监控工作负载资源-Well-Architected 监控模式

主题

- MediaTailor 错误代码和 CDN 集成疑难解答
- CDN 日志分析工具和监控技术 MediaTailor
- MediaTailor 记录配置资源

MediaTailor 错误代码和 CDN 集成疑难解答

AWS Elemental MediaTailor 当通过内容分发网络 (CDN) 时，错误代码提供了有关集成问题的具体信息。当通过您的内容分发网络 MediaTailor 返回错误时，这些代码表示存在具体问题：

注意：这些错误代码由您的 CDN 返回 MediaTailor 并传递。有关特定于 CDN 的错误，请查阅 CDN 提供商的文档。

400 错误请求

常见原因：查询参数格式错误、缺少必填参数、会话无效 IDs

检查：CDN 查询字符串转发配置、参数编码

403 禁止访问

常见原因：客户端玩家请求的区段不存在或缺乏访问权限。对于广告细分， MediaTailor 可能指定了不存在的区段（请联系 AWS Su ppor t 进行调查）。对于内容分段、来源提供商访问限制或身份验证问题

检查：对于广告细分：如果生成了无效的区段，请联系 AWS S MediaTailor upport URLs。对于内容分段：验证源服务器权限、身份验证凭据和访问策略。检查所请求的区段是否存在 URLs

404 未找到

常见原因： MediaTailor 配置不存在或处于非活动状态、播放网址路径不正确、对不存在资源的清单或分段请求

检查：验证 MediaTailor 配置是否存在且处于活动状态，确认播放 URL 与发件人`ManifestEndpointPrefix`匹配`GetPlaybackConfiguration`，检查 CDN 路由规则将请求转发到正确的端点 MediaTailor

500 内部服务器错误

常见原因：源服务器问题、ADS 连接问题、清单处理错误

检查：源服务器运行状况、ADS 响应有效性、 MediaTailor 服务状态

502 错误的网关

常见原因：由`UnsupportedManifestException`于 HLS 播放列表对齐问题，无法访问 Origin 服务器、源站响应无效（播放列表中的 SCTE 标记未对齐、某些播放列表中缺少 SCTE 标记、比特率变体之间的广告中断时间不一致）

检查：Origin Server 连接和防火墙规则、DNS 解析、所有比特率变体的 HLS 播放列表一致性、所有播放列表中相同分段的 SCTE-35 标记对齐情况、验证所有播放列表是否包含所需的广告中断标记

错误代码分析资源：

- 疑难解答 MediaTailor-完整的 MediaTailor 错误代码参考

- CloudFront 错误响应故障排除-特定于 CDN 的错误分析
- HTTP 响应状态代码-全面的 HTTP 状态码参考
- 故障管理-Well-Architected 错误处理模式

CDN 日志分析工具和监控技术 MediaTailor

AWS Elemental MediaTailor 内容分发网络 (CDN) 集成会生成大量日志数据，这些数据需要高效的分析工具和技术。使用以下方法高效分析内容分发网络和 MediaTailor 日志：

- 命令行分析：使用`grep`、`awk`和之类的工具筛选和`sort`分析日志模式
- Amazon L CloudWatch ogs Insights：使用类似 SQL 的语法查询 CDN 和 MediaTailor日志以进行高级分析
- 第三方工具：考虑使用日志分析平台进行全面监控和警报
- 自定义仪表板：创建将 CDN 指标与性能数据相结合的可视化效果 MediaTailor

如果您在日志分析或解释复杂错误模式方面需要其他帮助，请参阅获取 CDN 集成支持。

日志分析工具和资源：

- CloudWatch 日志见解-用于日志分析的类似 SQL 的查询
- Amazon OpenSearch 服务-高级日志搜索和分析
- CloudWatch 仪表板-自定义可视化和监控
- 对分布式系统进行探测以实现运营可见性-高级可观测性模式
- 设计工作负载可观察性 ——Well-Architected 可观测性指南

MediaTailor 记录配置资源

除了 CDN 日志外，还 MediaTailor 提供全面的日志记录功能，用于监控广告插入、清单生成和服务交互。使用以下资源来配置和分析 MediaTailor 日志：

MediaTailor 清单和服务日志

配置和分析清单生成、源交互和服务事件的日志：

- AWS Elemental MediaTailor 清单日志描述和事件类型- MediaTailor 清单日志和事件类型的完整参考

- 查看 AWS Elemental MediaTailor logs-查看和解释 MediaTailor 日志的指南

已售日志配置

通过成本优化，配置向多个目的地的灵活日志传输：

- 使用已售日志发送 AWS Elemental MediaTailor logs-配置向 Amazon S3、Firehose 或日志的日志传输 CloudWatch
- 迁移您的 AWS Elemental MediaTailor 记录策略-现有日志配置的迁移指南

CloudWatch 日志集成

将 MediaTailor 日志与集成 CloudWatch 以进行监控和分析：

- 查看 AWS Elemental MediaTailor logs- CloudWatch 日志配置和分析
- CloudWatchInsights-高级日志查询和分析

特定于广告的日志记录

监控广告插入效果并对广告相关问题进行故障排除：

- ADS 互动日志-监控广告决策服务器的通信和错误

CDN 和 MediaTailor 日志之间的主要区别：

- CDN 日志：显示内容分发网络中的 request/response 模式、缓存行为和网络级错误
- MediaTailor 日志：显示广告插入详情、清单生成事件、来源互动和特定于服务的错误
- 组合分析：同时使用两种日志类型，以全面了解您的流媒体工作流程

有关将 CDN 和 MediaTailor 日志记录结合在一起的全面监控，请参阅监控 MediaTailor CDN 的运营和性能。

使用自动化 MediaTailor 和 CDN CloudFormation

使用内容分发网络 (CDN) 进行自动化 AWS Elemental MediaTailor 可以 AWS CloudFormation 简化广告插入工作流程并提高可靠性。

本节介绍如何使用 AWS CloudFormation （ AWS 基础设施即代码服务 ）自动设置 AWS Elemental MediaTailor 内容分发网络 (CDN)。尽管您可以按照前几节所述手动配置此集成，但使用通过使用单个模板自动完成整个过程，从而 CloudFormation 节省时间并减少错误。

如果您不熟悉 CloudFormation，该服务可让您创建模板文件，定义所需的所有 AWS 资源。部署此模板时， CloudFormation 会自动为您创建和配置这些资源，确保它们可以正常协作。

有关的更多信息 CloudFormation，请参阅《CloudFormation 用户指南》。有关中 MediaTailor 资源类型的信息 CloudFormation，请参阅AWS:: MediaTailor 资源类型参考。

主题

- 为什么要使用 CloudFormation MediaTailor 和 CDN 集成
- 为 CloudFormation 部署 CDN 和 MediaTailor 集成做好准备
- 部署 CDN 和 MediaTailor 集成的 CloudFormation 模板
- 使用 CloudFormation 已部署的资源进行 CDN 和集成 MediaTailor
- 测试和验证您的 CDN CloudFormation 部署和集成 MediaTailor
- 排除 CDN 和 MediaTailor 集成的常见 CloudFormation 部署问题
- 自定义 CDN 和 MediaTailor 集成的 CloudFormation 模板
- AWS CloudFormation AWS Elemental MediaTailor 和 Amazon CloudFront 集成的模板参考

为什么要使用 CloudFormation MediaTailor 和 CDN 集成

AWS Elemental MediaTailor 自动化为管理流媒体工作流程的广播专业人员 AWS CloudFormation 提供了显著的优势。手动配置 MediaTailor 内容分发网络 (CDN) 可能既耗时又容易出错。使用 CloudFormation 自动化具有以下优点。

- 一致性：确保每次部署相同的配置，从而减少人为错误。
- 版本控制：将您的基础架构作为代码存储在版本控制系统中，以跟踪更改。
- 快速部署：只需几分钟即可部署复杂的配置，而无需花费数小时的手动配置。
- 环境复制：在开发、测试和生产环境中轻松复制配置。
- 文档：模板本身可用作基础架构的文档。

以下是自动化工作流程与手动配置的对比：

手动设置（多个步骤）	自动设置（单一模板）
创建 MediaTailor 播放配置	部署一个带有参数的 CloudFormation 模板
创建 CloudFront 发行版	

手动设置（多个步骤）	自动设置（单一模板）
配置缓存行为	
设置安全配置	

用于设置的自动工作流程 MediaTailor CloudFront 遵循以下步骤：

1. 使用您的内容来源和广告服务器参数部署 CloudFormation 模板
2. CloudFormation 创建并配置所有必要的资源：
   - MediaTailor 广告插入的播放配置
   - CloudFront 使用适当的缓存行为进行分发
   - 内容保护的安全配置
3. 使用 CloudFormation 输出来访问你支持广告的直播 URLs
4. 使用动态插入的广告直播您的内容

为 CloudFormation 部署 CDN 和 MediaTailor 集成做好准备

AWS Elemental MediaTailor 使用进行部署 AWS CloudFormation 需要特定的先决条件和准备步骤。在开始使用 CloudFormation 整合 MediaTailor 和 Amazon 之前 CloudFront，请确保您具备以下条件。

- 具有创建 MediaTailor CloudFront、和 CloudFormation 资源的权限的 AWS 账户
- 托管视频内容的内容来源（例如 AWS Elemental MediaPackage Amazon S3 或其他源服务器）
- 可以响应 VAST 请求的广告决策服务器 (ADS)

在部署 CloudFormation 模板之前，请收集以下必需的参数：

`AdServerUrl`

用于动态广告插入的 VAST 广告服务器的网址。提供了用于测试的静态 VAST 端点。

`ContentOriginDomainName`

不含协议的内容来源的域名（例如*mediapackage-domain.mediapackagev2.us-west-2.amazonaws.com*、*mybucket.s3.amazonaws.com*、或*custom-origin.example.com*）。请勿包含 http://或 https://前缀或任何路径。

`ContentOriginType`

内容来源的类型：

- mediapackagev2：For Origin s AWS Elemental MediaPackage
- s3：适用于亚马逊 S3 存储桶来源
- 自定义：适用于任何其他来源类型

该模板将创建多个 AWS 资源，这些资源可以协同工作，通过个性化广告提供您的内容。以下描述了每个组件的作用：

源站访问控制

Origin Access Control (OAC) 是一项安全功能，可确保您的内容只能通过源服务器进行访问 CloudFront，而不能直接从源服务器访问。这有助于保护您的内容免遭未经授权的访问。

对于 MediaPackage 和 Amazon S3 来源，该模板会创建源访问控制 (OAC) 资源来保护对您的内容的访问。

MediaTailor 播放配置

MediaTailor 播放配置是处理广告插入的核心组件。它接收来自您的来源的内容，从您的广告服务器请求广告，然后将它们组合成针对每位观众的个性化直播流。

该模板使用以下设置创建 MediaTailor 播放配置：

- 指向您的 CloudFront 发行版的视频内容来源
- 为您指定的 VAST 端点配置的广告决策服务器网址
- 直播前置片配置，用于在直播期间插入广告
- 带有适当分段 URL 前缀的 CDN 配置

CloudFront 分布

该 CloudFront 发行版以低延迟将您的内容交付给全球观众。它处理不同类型的请求（清单、内容区段、广告区段），并将它们路由到相应的来源。

对于刚接触的广播专业人士 CDNs，以下是一些关键术语：

Origin

存储您的原始内容的服务器（例如 MediaPackage 或 Amazon S3）

缓存行为

确定如何缓存和交付不同类型内容的规则

缓存策略

控制缓存内容的时间以及哪些请求组件会影响缓存的设置

该模板创建了一个包含以下组件的 CloudFront 分配：

- 三个起源：
  - 内容来源（MediaPackage、Amazon S3 或自定义）
  - MediaTailor 表明起源
  - MediaTailor 区段起源
- 使用适当的模式缓存行为：
  - 内容分段的默认行为
  - MediaTailor 广告细分的行为 (/tm/*)
  - MediaTailor 插页式媒体的行为 (/v1/i-media/*)
  - 个性化清单的行为 (/v1/*)
  - 区段重定向请求的行为 (/segment/*)
- 针对每种行为优化了缓存策略：
  - `CachingOptimized`用于可缓存的内容
  - `CachingDisabled`用于个性化清单
- 原始请求策略可确保正确转发标头
- CORS 支持的响应标头策略

部署 CDN 和 MediaTailor 集成的 CloudFormation 模板

AWS Elemental MediaTailor 一旦你了解了 AWS CloudFormation 模板将创建什么，就可以使用模板进行部署了。此过程大约需要 15-30 分钟，大部分时间都花在等待 CloudFront分发部署上。

要部署 CloudFormation 模板并设置自动广告插入工作流程，请执行以下操作：

部署 MediaTailor CloudFormation 模板

1. 从 AWS Elemental MediaTailor GitHub 存储库下载 CloudFormation 模板或从中复制AWS CloudFormation AWS Elemental MediaTailor 和 Amazon CloudFront 集成的模板参考。
2. 打开 CloudFormation 控制台。
3. 选择创建堆栈 > 使用新资源（标准）。
4. 在指定模板下，选择上传模板文件并上传模板。
5. 输入堆栈名称并提供所需参数的值：
   - AdServerUrl: 你的 VAST 广告服务器的网址（例如，https: *your-ad-server.com* ///vast ）
   - ContentOriginDomainName: 不含协议的内容来源的域名（例如*mediapackage-domain.mediapackagev2.us-west-2.amazonaws.com* ）
   - ContentOriginType：选择内容来源的类型：
     - mediapackagev2：For Origin s AWS Elemental MediaPackage
     - s3：适用于亚马逊 S3 存储桶来源
     - 自定义：适用于任何其他来源类型
6. 查看配置并选择创建堆栈。
7. 等待堆栈创建完成，通常需要 5-10 分钟。您可以在 “事件” 选项卡中监控进度。
8. 完成后，导航到 “输出” 选项卡，查找 HLS 和 DASH 清单的。 URLs

> **Note**
>
> 如果您使用 AWS Elemental MediaPackage 作为内容来源，请确保您的 MediaPackage 终端节点配置正确且可访问。有关更多信息，请参阅 MediaPackage 内容分发网络集成。

使用 CloudFormation 已部署的资源进行 CDN 和集成 MediaTailor

AWS Elemental MediaTailor AWS CloudFormation 堆栈部署的资源提供了几个重要的输出，您可以使用这些输出通过广告插入来访问您的内容。成功创建 CloudFormation 堆栈后，您需要了解如何使用输出来访问插入广告的内容。这与您在手动设置 MediaTailor URLs 中使用的方式类似，但是 CloudFormation 部署会 URLs自动提供这些设置。

成功部署后， CloudFormation 堆栈会提供几个重要的输出，您可以使用这些输出通过广告插入来访问您的内容：

`CloudFrontDomainName`

您的 CloudFront 发行版的域名（例如*d1234abcdef.CloudFront.net*）

`HlsManifestUrl`

带有广告插入功能的 HLS 清单的基本网址（例如，https: *d1234abcdef.CloudFront.net* ///v *12345* 1/master//) *my-playback-config*

`DashManifestUrl`

带有广告插入功能的 DASH 清单的基本网址（例如，https: *d1234abcdef.CloudFront.net* /// v1/d *12345* ash///) *my-playback-config*

`MediaTailorPlaybackConfigName`

已创建的 MediaTailor 播放配置的名称（例如*my-stack-PlaybackConfig*）

构造回放 URLs

要为包含广告的内容创建完整的播放网址，您需要将 CloudFormation 输出中的基本网址与您的特定清单路径相结合。这是广播专业人士需要理解的关键步骤，因为它可以将您现有的内容与广告插入系统联系起来。

1. 从输出中相应的清单 URL 开始：

```
HlsManifestUrl: https://d1234abcdef.CloudFront.net/v1/master/12345/my-playback-config/
```

2. 追加您的特定清单路径：

```
Your manifest path: channel/index.m3u8
```

3. 完整的播放网址变为：

```
https://d1234abcdef.CloudFront.net/v1/master/12345/my-playback-config/channel/index.m3u8
```

在视频播放器中使用此网址来播放带有动态插入广告的内容。

> **Tip**
>
> 如果您不确定清单路径应该是什么，请检查您的源服务器。对于 MediaPackage 源站来说，这是通往终端节点的 HLS 或 DASH 清单的路径。对于 Amazon S3 来源，这是存储桶内清单文件的路径。

有关 MediaTailor URL 结构的更多信息，请参阅通过以下方式设置 CDN 集成 MediaTailor。

配置视频播放器

获得播放 URL 后，您需要配置视频播放器才能使用它。对于广播专业人士来说，这类似于为任何 HLS 或 DASH 直播配置播放器，但现在直播将包含个性化广告。以下是使用流行HLS.js 播放器的简单示例：

```
<!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>
```

你也可以使用专业的广播播放器，比如：

- JW Player
- Bitmovin 播放器
- THEOplayer
- Video.js

有关玩家与集成的更多信息 MediaTailor，请参阅MediaTailor 广告服务器集成要求。

测试和验证您的 CDN CloudFormation 部署和集成 MediaTailor

AWS Elemental MediaTailor 部署验证是广播专业人员在上线之前的关键步骤。本节将指导您完成部署测试，以确保正确插入广告并顺利交付内容。

部署 CloudFormation 模板后，请按照以下步骤验证您的设置是否正常运行：

测试您的 MediaTailor 和 CloudFront 集成

1. 确认所有资源均已在 CloudFormation 控制台中成功创建。
2. 检查MediaTailor 控制台中的 MediaTailor 播放配置是否处于活动状态。
3. 验证 CloudFront 发行版是否已在CloudFront 控制台中部署并启用。
4. 使用示例清单测试播放：
    1. 按照中所述构造完整的播放网址构造回放 URLs。
    2. 使用支持 HLS 或 DASH 的视频播放器（比如 VLC、JW Player 或 AWS 主机播放器）。
    3. 验证内容是否在预期的断点播放和广告插入。
5. 请检查 MediaTailor 登录信息中 CloudWatch 是否存在任何广告插入错误。

在测试广告插入时，请注意以下成功指标：

- 内容和广告之间的平滑过渡
- 广告出现在预期的断点处（前置片段、片中片段、片后片段）
- 广告质量与内容质量相匹配
- 广告过渡期间没有缓冲或播放错误

有关更详细的测试程序，请参阅了解 AWS Elemental MediaTailor 广告插入行为。有关全面的 CDN 集成测试和验证，请参阅CDN 和集成的测试和 MediaTailor 验证。

排除 CDN 和 MediaTailor 集成的常见 CloudFormation 部署问题

AWS Elemental MediaTailor 即使在部署或播放期间实现自动化，也可能出现部署问题。作为广播专业人士，了解如何解决这些问题将有助于您在广告插入方面保持可靠的流媒体服务。

如果您在 CloudFormation 部署或由此产生的 MediaTailor 和 CloudFront集成中遇到问题，请参阅以下常见问题和解决方案：

CloudFormation 部署问题

堆栈创建失败，出现 “资源创建失败” 错误

可能的原因：

- 内容来源域名格式无效
- 创建资源的权限不足

解决方案：在 “ CloudFormation 事件” 选项卡中检查特定的资源错误。验证内容来源域名的格式是否正确，没有协议前缀或路径。确保您的 IAM 角色拥有足够的权限来创建所有必需的资源。

CloudFront 分发需要很长时间才能部署

原因： CloudFront 分发通常需要 15-30 分钟才能完全部署。

解决方案：这是常规行为。等待分发达到 “已部署” 状态，然后再进行测试。

播放和广告插入问题

内容已播放，但未插入广告

可能的原因：

- 广告决策服务器没有响应或返回空的 VAST
- 内容不包含广告标记

解决方案：验证您的广告服务器是否可访问并返回有效的 VAST 响应。检查您的内容是否有正确的广告标记（直播内容的 SCTE-35 标记或视频点播的广告插播标记）。

403 访问内容时出现禁止的错误

可能的原因：

- 源站访问控制配置不正确
- 源存储桶或终端节点权限问题

解决方案：对于 Amazon S3 来源，请验证存储桶策略是否允许从 CloudFront 分配中进行访问。对于 MediaPackage 源，请检查源站访问控制是否配置正确，以及端点是否可访问。

播放错误或缓冲

可能的原因：

- 缓存行为路径模式与内容路径不匹配
- 源域配置不正确

解决方案：验证您的缓存行为是否具有正确的路径模式，以便将请求路由到相应的源。检查 CloudFront 日志以查看哪个源在处理请求并确认这是预期的来源。

对于广播专业人士，以下其他故障排除提示可能会有所帮助：

- 使用 Amazon CloudWatch Logs Insights 在 MediaTailor日志中搜索特定的错误模式
- 在使用制作广告服务器之前，请先使用简单的 VAST 广告服务器（如模板中提供的默认广告服务器）进行测试
- 使用控制台中的 MediaTailor 清单检查器工具验证内容的广告标记
- 在浏览器的开发者工具中检查网络流量，看看向广告服务器发出的请求是否正确

要进行其他故障排除，请查看 CloudWatch 日志 CloudFront 以了解两者 MediaTailor 并确定具体错误。

自定义 CDN 和 MediaTailor 集成的 CloudFormation 模板

AWS Elemental MediaTailor 模板自定义允许广播专业人员调整 AWS CloudFormation 模板以适应特定的工作流程要求。尽管基本模板适用于许多场景，但这些自定义项可以帮助您满足更复杂的需求。

以下示例显示了您可以添加到模板中的 YAML 代码片段。如果您不熟悉 YAML 或 CloudFormation 语法，可以考虑与开发人员或 AWS 解决方案架构师合作进行这些更改。

您可以自定义 CloudFormation 模板以满足您的特定工作流程要求。

添加或修改起点

对于使用多个内容源（例如主源和备用源，或者不同的内容库）的广播工作流程，您可以向 CloudFront 分发添加其他来源：

```
Origins:
  # Add a new origin for additional content
```

```
  - Id: SecondaryContentOrigin
    DomainName: secondary-content.example.com
    CustomOriginConfig:
      OriginProtocolPolicy: 'https-only'
      OriginSSLProtocols:
        - TLSv1.2
```

然后添加相应的缓存行为，将特定的模式路由到这个来源：

```
CacheBehaviors:
  - PathPattern: '/secondary-content/*'
    TargetOriginId: SecondaryContentOrigin
    ViewerProtocolPolicy: 'https-only'
    CachePolicyId: 658327ea-f89d-4fab-a63d-7e88639e58f6  # Managed-CachingOptimized
```

创建自定义缓存策略

对于具有特定缓存要求（例如质量选择参数或观众身份验证）的广播工作流程，您可以创建自定义缓存策略，而不是使用托管缓存策略。有关 TTL 值和缓存策略的详细指导，请参阅针对 CDN 和 MediaTailor 集成的缓存优化。

```
# Define a custom cache policy
CustomCachePolicy:
  Type: AWS::CloudFront::CachePolicy
  Properties:
    CachePolicyConfig:
      Name: !Sub '${AWS::StackName}-CustomCachePolicy'
      DefaultTTL: 86400  # 24 hours
      MaxTTL: 31536000   # 1 year
      MinTTL: 1          # 1 second
      ParametersInCacheKeyAndForwardedToOrigin:
        CookiesConfig:
          CookieBehavior: none
        HeadersConfig:
          HeaderBehavior: none
        QueryStringsConfig:
          QueryStringBehavior: whitelist
          QueryStrings:
            - quality
            - format

# Reference the custom policy in a cache behavior
```

```
CacheBehaviors:
  - PathPattern: '/custom-path/*'
    TargetOriginId: ContentOrigin
    ViewerProtocolPolicy: 'https-only'
    CachePolicyId: !Ref CustomCachePolicy
```

增强 MediaTailor 配置

对于需要高级广告插入功能的广播工作流程，您可以使用广告预取（以减少延迟）、个性化阈值和导视广告等选项来增强 MediaTailor 配置。

```
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...
```

有关 MediaTailor 配置选项的更多信息，请参阅使用 AWS Elemental MediaTailor 插入广告。

添加安全功能

对于具有特定安全要求（例如地理限制或防护 DDo S 攻击）的广播工作流程，您可以添加 AWS WAF 集成和地理限制：

```
# 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...
```

有关 CloudFormation 模板的更多信息，请参阅《AWS CloudFormation 用户指南》。

有关特定于广播的 CloudFormation 模板和示例，请参阅 AW S 媒体服务工具存储库。 GitHub

AWS CloudFormation AWS Elemental MediaTailor 和 Amazon CloudFront 集成的模板参考

AWS Elemental MediaTailor CloudFront 可以使用以下完整 AWS CloudFormation 模板自动与 Amazon 集成：

```
AWSTemplateFormatVersion: '2010-09-09'
```

```
Description: | CloudFormation template that sets up AWS Elemental MediaTailor
 integration with CloudFront Distribution
  for server-side ad insertion. This template supports various content origins
 including MediaPackage, Amazon S3,
  and custom origins, making it versatile for different streaming architectures.

Parameters:
  AdServerUrl:
    Type: String
    Default: 'https://d1kbmkziz9rksx.CloudFront.net/VASTEndpoint.xml'
    Description: URL of the VAST ad server for dynamic ad insertion. Static VAST
 endpoint provided for testing.

  ContentOriginDomainName:
    Type: String
    Description: |
      Domain name of your content origin without protocol (e.g., mediapackage-
domain.mediapackagev2.us-west-2.amazonaws.com,
      mybucket.s3.amazonaws.com, or custom-origin.example.com).
      Do not include http:// or https:// prefixes or any paths.
    AllowedPattern: "^(([a-zA-Z0-9]|[a-zA-Z0-9][a-zA-Z0-9\\-]*[a-zA-Z0-9])\\.)*([A-Za-
z0-9]|[A-Za-z0-9][A-Za-z0-9\\-]*[A-Za-z0-9])$"
    ConstraintDescription: Must be a valid domain name (e.g., example.com) without
 protocol or path components. IP addresses are not allowed.

  ContentOriginType:
    Type: String
    AllowedValues:
      - mediapackagev2
      - s3
      - custom
    Default: mediapackagev2
    Description: |
      The type of content origin:
      - mediapackagev2: AWS Elemental MediaPackage V2
      - s3: Amazon S3 bucket
      - custom: Any other custom origin

Resources:
  #-------------------------------------------------------------------------
  # Origin Access Control (for securing MediaPackage V2 and Amazon S3 origins)
  #-------------------------------------------------------------------------
  CloudFrontOriginAccessControl:
    Type: AWS::CloudFront::OriginAccessControl
```

```
    Condition: IsNotCustomOrigin
    Properties:
      OriginAccessControlConfig:
        Name: !Sub '${AWS::StackName}-OAC'
        OriginAccessControlOriginType: !Ref ContentOriginType
        SigningBehavior: always
        SigningProtocol: sigv4
        Description: Origin Access Control for content origin

  #------------------------------------------------------------------------
  # MediaTailor Playback Configuration
  #------------------------------------------------------------------------
  MediaTailorPlaybackConfig:
    Type: AWS::MediaTailor::PlaybackConfiguration
    Properties:
      Name: !Sub '${AWS::StackName}-PlaybackConfig'
      # The video content source should point to your CloudFront distribution
      VideoContentSourceUrl: !Sub 'https://${CloudFrontDistribution.DomainName}/'
      # The Ad Decision Server URL is where MediaTailor will request ads
      AdDecisionServerUrl: !Ref AdServerUrl
      # Configuration for pre-roll ads during live streams
      LivePreRollConfiguration:
        AdDecisionServerUrl: !Ref AdServerUrl
        MaxDurationSeconds: 30
      # CDN configuration for integrating with CloudFront
      CdnConfiguration:
        AdSegmentUrlPrefix: '/'
        ContentSegmentUrlPrefix: '/'
      # Set a reasonable manifest segment timeout
      ManifestProcessingRules:
        AdMarkerPassthrough:
          Enabled: false

  #------------------------------------------------------------------------
  # CloudFront Distribution
  #------------------------------------------------------------------------
  CloudFrontDistribution:
    Type: AWS::CloudFront::Distribution
    Properties:
      DistributionConfig:
        Enabled: true
        HttpVersion: http2and3
        IPV6Enabled: true
```

```
        Comment: !Sub 'Distribution for MediaTailor ad insertion with
 ${ContentOriginType} origin'

        # Default cache behavior points to the content origin
        DefaultCacheBehavior:
          TargetOriginId: ContentOrigin
          ViewerProtocolPolicy: 'https-only'
          # Using managed policies for optimal performance and simplicity
          CachePolicyId: 658327ea-f89d-4fab-a63d-7e88639e58f6  # Managed-
CachingOptimized
          OriginRequestPolicyId: 88a5eaf4-2fd4-4709-b370-b4c650ea3fcf # Managed-
HostHeaderOnly
          ResponseHeadersPolicyId: eaab4381-ed33-4a86-88ca-d9558dc6cd63  # Managed-
CORS-with-preflight-and-SecurityHeadersPolicy
          Compress: true

        # Define all the origins needed for the workflow
        Origins:
          # Main content origin (MediaPackage, Amazon S3, or Custom)
          - Id: ContentOrigin
            DomainName: !Ref ContentOriginDomainName
            # Apply Origin Access Control for secure origins
            OriginAccessControlId: !If [IsNotCustomOrigin, !GetAtt
 CloudFrontOriginAccessControl.Id, !Ref "AWS::NoValue"]
            # For custom origins, we need a CustomOriginConfig
            CustomOriginConfig:
              OriginProtocolPolicy: 'https-only'
              OriginSSLProtocols:
                - TLSv1.2
              OriginKeepaliveTimeout: 5
              OriginReadTimeout: 30
              HTTPPort: 80
              HTTPSPort: 443

          # MediaTailor Manifests Origin - handles manifest manipulation for ad
 insertion
          - Id: MediaTailorManifests
            DomainName: !Sub 'manifests.mediatailor.${AWS::Region}.amazonaws.com'
            CustomOriginConfig:
              OriginProtocolPolicy: 'https-only'
              OriginSSLProtocols:
                - TLSv1.2
              OriginKeepaliveTimeout: 5
              OriginReadTimeout: 30
```

```
            # Origin Shield improves caching efficiency
            OriginShield:
              Enabled: true
              OriginShieldRegion: !Ref AWS::Region

          # MediaTailor Segments Origin - handles personalized ads
          - Id: MediaTailorSegments
            DomainName: !Sub 'segments.mediatailor.${AWS::Region}.amazonaws.com'
            CustomOriginConfig:
              OriginProtocolPolicy: 'https-only'
              OriginSSLProtocols:
                - TLSv1.2
              OriginKeepaliveTimeout: 5
              OriginReadTimeout: 30

        # Cache behaviors to route specific request patterns to the right origin
        CacheBehaviors:
          # Handle MediaTailor segment requests for ad content which are cache-able
          - PathPattern: '/tm/*'
            TargetOriginId: MediaTailorSegments
            ViewerProtocolPolicy: 'https-only'
            CachePolicyId: 658327ea-f89d-4fab-a63d-7e88639e58f6  # Managed-
CachingOptimized
            OriginRequestPolicyId: 88a5eaf4-2fd4-4709-b370-b4c650ea3fcf  # Managed-
HostHeaderOnly
            ResponseHeadersPolicyId: eaab4381-ed33-4a86-88ca-d9558dc6cd63  # Managed-
CORS-with-preflight-and-SecurityHeadersPolicy
            Compress: true

          # Handle MediaTailor interstitial (SGAI) media requests which are cache-able
          - PathPattern: '/v1/i-media/*'
            TargetOriginId: MediaTailorManifests
            ViewerProtocolPolicy: 'https-only'
            CachePolicyId: 658327ea-f89d-4fab-a63d-7e88639e58f6  # Managed-
CachingOptimized
            OriginRequestPolicyId: 88a5eaf4-2fd4-4709-b370-b4c650ea3fcf  # Managed-
HostHeaderOnly
            ResponseHeadersPolicyId: eaab4381-ed33-4a86-88ca-d9558dc6cd63  # Managed-
CORS-with-preflight-and-SecurityHeadersPolicy
            Compress: true

          # Handle MediaTailor Personalized manifests which are not cache-able
          - PathPattern: '/v1/*'
            TargetOriginId: MediaTailorManifests
```

```
            ViewerProtocolPolicy: 'https-only'
            CachePolicyId: 4135ea2d-6df8-44a3-9df3-4b5a84be39ad # Managed-
CachingDisabled
            OriginRequestPolicyId: 59781a5b-3903-41f3-afcb-af62929ccde1 # Managed-
AllViewer
            ResponseHeadersPolicyId: eaab4381-ed33-4a86-88ca-d9558dc6cd63  # Managed-
CORS-with-preflight-and-SecurityHeadersPolicy
            Compress: true

          # Handle MediaTailor segment *redirect* requests which are not cache-able
 (used for server side reporting)
          - PathPattern: '/segment/*'
            TargetOriginId: MediaTailorManifests
            ViewerProtocolPolicy: 'https-only'
            CachePolicyId: 4135ea2d-6df8-44a3-9df3-4b5a84be39ad # Managed-
CachingDisabled
            OriginRequestPolicyId: 59781a5b-3903-41f3-afcb-af62929ccde1 # Managed-
AllViewer
            ResponseHeadersPolicyId: eaab4381-ed33-4a86-88ca-d9558dc6cd63  # Managed-
CORS-with-preflight-and-SecurityHeadersPolicy
            Compress: true

Conditions:
  IsNotCustomOrigin: !Not [!Equals [!Ref ContentOriginType, 'custom']]

Outputs:
  CloudFrontDomainName:
    Description: Domain name of the CloudFront distribution
    Value: !GetAtt CloudFrontDistribution.DomainName

  HlsManifestUrl:
    Description: URL for HLS manifest with ads inserted (append your manifest path)
    Value: !Sub 'https://
${CloudFrontDistribution.DomainName}${MediaTailorPlaybackConfig.HlsConfiguration.ManifestEndpoi

  DashManifestUrl:
    Description: URL for DASH manifest with ads inserted (append your manifest path)
    Value: !Sub 'https://
${CloudFrontDistribution.DomainName}${MediaTailorPlaybackConfig.DashConfiguration.ManifestEndpo

  MediaTailorPlaybackConfigName:
    Description: Name of the MediaTailor playback configuration
    Value: !Ref MediaTailorPlaybackConfig
```

Production-ready CloudFront 的配置 MediaTailor

这种 CloudFront 分发配置提供了大规模投放服务器端广告 MediaTailor 内容所需的一切。复制此配置并根据您的特定来源和要求对其进行自定义。

此配置可以完成什么

此配置创建了生产就绪的 CloudFront 发行版，该发行版以最佳的缓存和性能处理所有 MediaTailor请求类型。它包括三个来源（您的内容、 MediaTailor 区段和 MediaTailor 清单），以及四种缓存行为，可以正确路由请求并适当地缓存内容。

何时使用此配置

此设置非常适合需要在服务器端插入广告的直播、视频点播和混合工作流程。

Three-origin 建筑

MediaTailor 使用三源架构模式来优化内容交付和广告插入性能。在广告插入工作流程中，每个来源都有特定的用途：

你的内容来源

这是你真正的内容来源，供你参考 MediaTailor。例如，这可能是 AWS Elemental MediaPackage V2 或其他内容分发服务。该来源在广告插入之前提供您的原创内容。常见示例包括：

- MediaPackage V2 打包配置
- Third-party 内容分发网络
- On-premises 流媒体服务器
- 包含静态内容的 Amazon S3 存储桶

MediaTailor 区段起源

该来源使用主机名，`segments.mediatailor.region.amazonaws.com`并在对实际的广告段进行编码后 MediaTailor 提供这些广告段。这些是包含广告的视频片段。这个 Origin 处理：

- 转码后的广告细分格式与您的内容相同
- 来自`/segment/*`路径模式的重定向请求
- 经过处理以进行服务器端广告插入的广告区段

MediaTailor 表明起源

此来源使用主机名`manifests.mediatailor.region.amazonaws.com`，可用作指定 AWS 区域中播放配置的区域主机名。 MediaTailor 根据请求中的路径选择正确的播放配置。该起源提供：

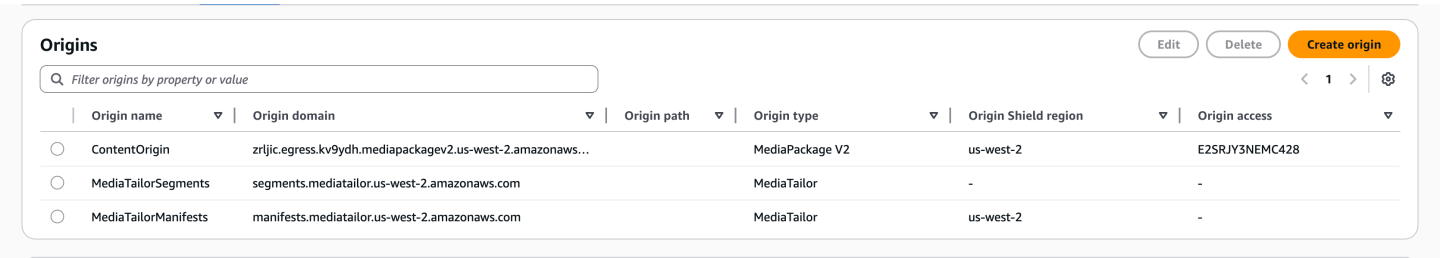
- 个性化的 HLS 和 DASH 清单，其中包含针对特定观众的广告插入
- Server-guided 可缓存内容的广告插入 (SGAI) 清单
- 服务器端报告的广告跟踪和信标处理

使用 origin 主机名`manifests.mediatailor.region.amazonaws.com`，如果多个播放配置位于指定区域，并且您在请求路径中包含播放配置名称，则可以拥有适用于相同 CloudFront 发行版的多个播放配置。例如：

- `https://your-distribution.cloudfront.net/v1/master/playback-config-1/manifest.m3u8`
- `https://your-distribution.cloudfront.net/v1/master/playback-config-2/manifest.m3u8`

根据内容类型选择源请求策略，以防止缓存中毒，同时确保正常运行。关键区别在于可缓存和不可缓存的内容：

- 清单（不可缓存）：用于`AllViewer`转发动态内容所需的所有标题。由于清单未被缓存，因此不存在缓存中毒的风险。
- 区段（可缓存）：`None`用于获得最佳性能。
- S3 来源：`CORS-S3Origin`用于亚马逊 S3 存储桶
- MediaPackage 来源：用`CORS-S3Origin`于 MediaPackage V2 端点



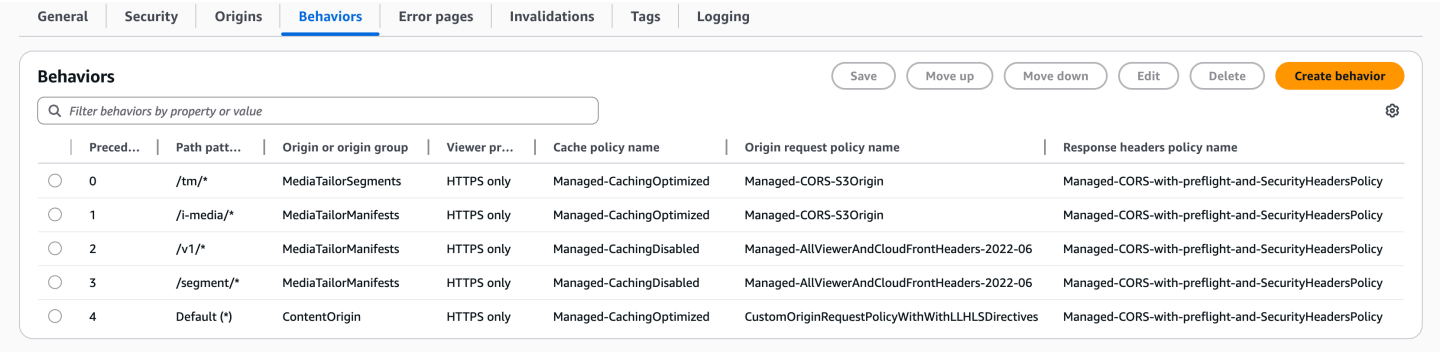
缓存行为的优先级和配置

MediaTailor 需要特定的缓存行为配置才能正确处理不同类型的请求。缓存行为的优先级至关重要，因为 CDN 按顺序（从小到大）处理缓存行为，并将该行为用于第一个匹配的路径模式。要进行故障排除，必须了解此优先顺序：

- 优先级 0：首先评估最具体的模式（比如`/tm/*`）

- 更高的优先级数字：按顺序评估不太具体的模式
- 默认行为：捕获所有与其他模式不匹配的请求

如果请求的行为不符合预期，请检查您的路径模式是否不会以意想不到的方式重叠。



优先级 0：广告区段路径行为

此行为处理来自细分路径行为的重定向请求，为实际的广告细分提供服务。 CloudFront 将以下行为应用于所有具有/tm/*路径模式的请求。这是最高优先级的行为，因为广告区段投放对于不间断播放至关重要。

- 路径模式：`/tm/*`

  与此模式匹配的 URL 示例：

  - `https://your-distribution.cloudfront.net/tm/ad-segment-001.ts`
  - `https://your-distribution.cloudfront.net/tm/transcoded-ad.m4s`
- 来源：您使用`segments.mediatailor.region.amazonaws.com`域名创建的来源。

  前面关于起源的章节中的示例就是MediaTailorSegments这样。
- 缓存策略：`Managed-CachingOptimized`

  有关 CloudFront 托管缓存策略中包含的内容的详细信息，请参阅 CloudFront 用户指南CachingOptimized中的。您也可以在第三方 CDN 中使用托管策略中的相同设置。
- 起源请求政策：None
- 响应标头政策：`Managed-CORS-with-preflight-and-SecurityHeadersPolicy`

  有关响应标头策略中包含的内容的详细信息，请参阅 CloudFront 用户指南CORS-with-preflight-and-SecurityHeadersPolicy中的。

调整这些设置以适应其他 CDN

如果您使用的是以外的 CDN CloudFront，请查找可完成以下操作的等效设置。

路径模式匹配

为/tm/*路径模式配置特定的行为以处理 MediaTailor 广告细分

缓存密钥配置

在缓存密钥中包含Origin标头，以确保针对不同来源分别缓存响应

标头转发

将标Origin题和其他标 CORS-related题转发到原点

响应标头管理

配置您的 CDN 以确保响应中Access-Control-Allow-Origin包含标头

具体的术语和配置选项因CDN提供商而异，但基本原理保持不变。

优先级 1： Server-guided 广告插入行为

当客户配置引导模式MediaTailor 服务器引导的广告插入概述和实现（提供可缓存的清单）时，此行为会处理 (SGAI) 请求。 CloudFront 将以下行为应用于所有具有/v1/i-media/*路径模式的请求。SGAI 允许更好的缓存性能，因为清单不是特定于查看者的。

- 路径模式：/v1/i-media/*（SGAI 的 iMedia 路径）

  与此模式匹配的 URL 示例：

  - https://your-distribution.cloudfront.net/v1/i-media/your-config/manifest.m3u8
  - https://your-distribution.cloudfront.net/v1/i-media/your-config/playlist.mpd
- 来源：您使用manifests.mediatailor.*region*.amazonaws.com域名创建的来源。

  前面关于起源的章节中的示例就是MediaTailorManifests这样。
- 缓存策略：Managed-CachingOptimized

  有关 CloudFront 托管缓存策略中包含的内容的详细信息，请参阅 CloudFront 用户指南CachingOptimized中的。您也可以在第三方 CDN 中使用托管策略中的相同设置。

- 起源请求政策：None
- 响应标头政策：Managed-CORS-with-preflight-and-SecurityHeadersPolicy

  有关响应标头策略中包含的内容的详细信息，请参阅 CloudFront 用户指南CORS-with-preflight-and-SecurityHeadersPolicy中的。

优先级 2：个性化清单行为

此行为处理个性化清单请求。 CloudFront 将以下行为应用于所有具有/v1/*路径模式的请求。CloudFront 应用以下行为并且不缓存个性化清单，因为它们包含特定于观众的广告内容网址。这些行为适用于所有具有/v1/*路径模式的请求。这是核心 MediaTailor 功能，每位观看者都会收到一份带有个性化广告插页的独特清单。

- 路径模式：/v1/*（标准 V1 MediaTailor 请求）

  与此模式匹配的 URL 示例：

  - https://your-distribution.cloudfront.net/v1/master/your-config/manifest.m3u8
  - https://your-distribution.cloudfront.net/v1/dash/your-config/manifest.mpd
- 来源：您使用manifests.mediatailor.*region*.amazonaws.com域名创建的来源。

  前面关于起源的章节中的示例就是MediaTailorManifests这样。
- 缓存策略：Managed-CachingDisabled

  有关缓存策略中包含的内容的详细信息，请参阅 CloudFront 用户指南CachingDisabled中的。
- 起源请求政策：AllViewer

  对于个性化清单，请使用该AllViewer政策转发动态内容所需的所有标题。
- 响应标头政策：Managed-CORS-with-preflight-and-SecurityHeadersPolicy

  有关响应标头策略中包含的内容的详细信息，请参阅 CloudFront 用户指南CORS-with-preflight-and-SecurityHeadersPolicy中的。

Accept-Encoding 标题

我们建议您的 CDN 保留观看者的`Accept-Encoding`标题。此标题提供了有关压缩个性化清单的 MediaTailor 说明。

在中 CloudFront，`AllViewerAndCloudFrontHeaders`原始请求策略包括来自查看者的`Accept-Encoding`标头的直通。如果您使用其他 CDN，请确保它保留此标头。

以下是如何 MediaTailor 处理标`Accept-Encoding`题。

- 传统设备：不支持 gzip 的较旧智能电视不会发送 Accept-Encoding 标头，因此会 MediaTailor 返回未压缩的清单
- 现代设备：iPhone、Chrome 浏览器和其他现代客户端会发送 Accept-Encoding 标题， MediaTailor 允许在交付之前压缩清单

优先级 3： Server-side 信标路径行为

此行为处理的请求 MediaTailor 会导致重定向。Server-side 追踪这些请求对于跟踪信标至关重要，因此每个请求都必须由 MediaTailor处理。 CloudFront 将以下行为应用于所有具有`/segment/*`路径模式的请求。

- 路径模式：`/segment/*`

  与此模式匹配的 URL 示例：

  - `https://your-distribution.cloudfront.net/segment/tracking-beacon-123`
  - `https://your-distribution.cloudfront.net/segment/ad-request-456.ts`
- 来源：您使用`manifests.mediatailor.region.amazonaws.com`域名创建的来源。

  前面关于起源的章节中的示例就是MediaTailorManifests这样。
- 缓存策略：`Managed-CachingDisabled`

  有关缓存策略中包含的内容的详细信息，请参阅 CloudFront 用户指南CachingDisabled中的。
- 起源请求政策：`AllViewer`

  对于服务器端信标请求，使用`AllViewer`策略转发跟踪所需的所有标头。由于这些请求未被缓存，因此不存在缓存中毒的风险。
- 响应标头政策：`Managed-CORS-with-preflight-and-SecurityHeadersPolicy`

  有关响应标头策略中包含的内容的详细信息，请参阅 CloudFront 用户指南CORS-with-preflight-and-SecurityHeadersPolicy中的。

MediaTailor 处理这些请求时，它会返回 302 重定向响应，其路径指向实际分段位置。例如，对的请求`/segment/ad123.ts`可能会在区段源`/tm/encoded-ad-segment.ts`上重定向到。

优先级 4：内容来源路径行为

如果请求路径与任何其他模式都不匹配， CloudFront 则应用默认行为。此行为将请求直接发送到内容来源，不进行任何处理 MediaTailor。这允许在需要时直接访问您的内容来源（例如 MediaPackage V2）。 CloudFront 将以下行为应用于所有不包含任何先前路径模式的请求。

- 路径模式：`(*)`
- 来源：您使用域名为内容来源创建的来源。

  前面关于起源的章节中的示例就是ContentOrigin这样。
- 缓存策略：`Managed-CachingOptimized`

  有关 CloudFront 托管缓存策略中包含的内容的详细信息，请参阅 CloudFront 用户指南CachingOptimized中的。您也可以在第三方 CDN 中使用托管策略中的相同设置。

  > **Note**
  >
  > 对于低延迟 HLS 实现，请考虑使用带有低延迟 HLS (LLH) 指令的自定义缓存策略，而不是标准策略。`CachingOptimized`
- 起源请求政策：None
- 响应标头政策：`Managed-CORS-with-preflight-and-SecurityHeadersPolicy`

  尽管默认内容来源行为通常不会面临与广告区段行为相同的 CORS 缓存中毒风险，但仍建议使用`Managed-CORS-with-preflight-and-SecurityHeadersPolicy`响应标头策略并将标头包含在缓存密钥中。`Origin`这样可以确保对所有内容类型进行一致的 CORS 处理，并防止基于 Web 的播放器中出现潜在的播放问题。

  对于内容分段，`Managed-CachingOptimized`缓存策略可提供良好的性能，而`Managed-CORS-with-preflight-and-SecurityHeadersPolicy`响应标头策略可确保正确处理 CORS。这种组合可以实现高效缓存，同时保持与需要 CORS 标头的基于 Web 的播放器的兼容性。

  在广告细分和内容细分中应用一致的 CORS 处理可创建更可靠的播放体验并简化故障排除。如果没有正确的 CORS 配置，玩家在内容和广告之间切换时可能会遇到不一致的行为。

有关响应标头策略中包含的内容的详细信息，请参阅 CloudFront 用户指南CORS-with-preflight-and-SecurityHeadersPolicy中的。

获取 CDN 和集成的支持和 MediaTailor 故障排除帮助

AWS Elemental MediaTailor CDN 集成问题可能难以诊断和解决。如果您需要其他帮助，解决无法通过自助故障排除解决的 CDN 和 MediaTailor 集成问题，请使用本指南。

在升级之前，请尝试以下自助服务选项：

- 按照中的故障排除步骤进行操作 对 CDN 集成进行故障排除
- 使用分析您的日志和错误代码 CDN 集成日志分析参考
- 使用以下命令查看您的监控设置 设置监控工具
- 查看MediaTailor 故障排除指南，了解特定于服务的问题
- 在 AWS re: Post 上搜索类似问题和社区解决方案
- 查看 AWS 知识中心，了解常见的集成模式

主题

- 何时将 MediaTailor CDN 问题上报给 AWS Support
- 在联系支持人员之前收集 MediaTailor CDN 信息
- MediaTailor CDN 集成、支持资源和渠道
- MediaTailor CDN 集成支持案例最佳实践

何时将 MediaTailor CDN 问题上报给 AWS Support

AWS Elemental MediaTailor 如果自助式故障排除无法解决问题，则应将 CDN 集成问题上报给 Supp AWS ort。在以下情况下，可以考虑升级到 S AWS upport

注意：技术支持需要付费的 Supp AWS ort 计划。有关支持计划的详细信息和响应时间，请参阅 AWS Support 计划。

- 问题会影响制作流量或收入
- 你已经完成了所有故障排除步骤，但没有解决问题
- 错误模式表明存在服务级别问题

- 在复杂的配置场景中，您需要帮助
- 尽管进行了优化，但性能问题仍然存在

在联系支持人员之前收集 MediaTailor CDN 信息

AWS Elemental MediaTailor CDN 集成支持案例需要特定的配置和诊断信息才能进行有效的故障排除。在联系 Su AWS pport 之前，请收集以下信息以帮助加快解决问题：

提示：在创建支持案例之前准备好这些信息将大大缩短解决时间，并有助于支持工程师了解您的特定配置。

- MediaTailor 配置详情：
  - 配置名称和 AWS 区域
  - 播放配置 ARN
  - 广告网址和集成类型
- CDN 配置详情：
  - CDN 分发 ID 或配置名称
  - 清单和区段的缓存行为配置
  - 源站配置和路由规则
- 错误信息：
  - 特定的错误消息和 HTTP 状态码
  - 问题发生的时间戳
  - 演 URLs 示问题的示例
  - 与问题相关的 CDN 和 MediaTailor 日志条目
- 测试信息：
  - 您已经采取的故障排除步骤
  - 出现问题的设备和播放器
  - 问题是影响所有内容还是影响特定直播
  - 问题的频率和模式（间歇性、一致性、基于时间）

MediaTailor CDN 集成、支持资源和渠道

AWS Elemental MediaTailor CDN 集成支持可通过多种渠道提供，以帮助您解决问题并优化实施：

- AWS S@@ upport：通过 AWS 管理控制台创建支持案例以获得技术帮助

  访问：Su AWS pport Center
- AWS re: post：社区驱动的问答平台，提供 AWS相关问题和社区支持

  访问权限：re: P AWS ost
- AWS 文档： MediaTailor 和 CDN 服务的综合指南

  访问权限：MediaTailor文档和CloudFront文档
- AWS 培训：媒体服务和 CDN 优化的课程和认证

  访问权限：AWS 培训和认证
- AWS 知识中心：关于常见 AWS 问题和最佳实践的精选文章

  访问权限：AWS 知识中心
- AWS Trusted Advisor：针对优化和最佳实践的自动建议

  访问：T AWS rusted Advisor
- AWS Pers@@ onal Health Dashboard：服务运行状况和维护通知

  访问权限：AWS 个人健康 Das hboard

MediaTailor CDN 集成支持案例最佳实践

AWS Elemental MediaTailor 如果您遵循这些最佳实践，则可以更有效地解决 CDN 集成支持案例。要获得最快的分辨率，请执行以下操作：

- 根据业务影响选择适当的严重性级别
- 在首次提交案例时提供所有相关信息
- 包括具体的示例和复制步骤
- 附上相关的日志文件和配置屏幕截图
- 及时回应支持工程师关于提供更多信息的请求

其他支持资源：

- AWS Support 案例管理-创建和管理支持案例的指南
- AWS S@@ upp ort plans-比较支持计划的功能和响应时间

- 为支持工作负载做好准备-Wel l-Architected 的运营准备指南
- AWS Well-Architected Frame work-在上构建和操作工作负载的最佳实践 AWS

中的安全性 AWS Elemental MediaTailor

的云安全 AWS 是重中之重。作为 AWS 客户，您可以受益于专为满足大多数安全敏感组织的要求而构建的数据中心和网络架构。

安全是您 AWS 和您的共同责任。责任共担模式将其描述为云的安全性和云中的安全性：

- 云安全 — 负责保护 AWS 在云中运行 AWS 服务的基础架构 AWS Cloud。 AWS 还为您提供可以安全使用的服务。 Third-party 作为AWS 合规计划合规计划合。要了解适用于的合规计划 AWS Elemental MediaTailor，请参阅合规计划范围内的AWS 服务（按合规性计划划分的范围内的。
- 云端安全 — 您的责任由您使用的 AWS 服务决定。您还需要对其他因素负责，包括您的数据的敏感性、您公司的要求以及适用的法律法规。

本文档可帮助您了解在使用时如何应用分担责任模型 MediaTailor。以下主题向您展示如何进行配置 MediaTailor 以满足您的安全和合规性目标。您还将学习如何使用其他 AWS 服务来监控和保护您的 MediaTailor 资源。

主题

- 中的数据保护 AWS Elemental MediaTailor
- 的身份和访问管理 AWS Elemental MediaTailor
- 的合规性验证 AWS Elemental MediaTailor
- 弹性在 AWS Elemental MediaTailor
- 中的基础设施安全 MediaTailor
- Cross-service 困惑的副手预防
- 登录和监控 MediaTailor

中的数据保护 AWS Elemental MediaTailor

AWS 责任共担模式适用于保护 AWS Elemental MediaTailor中的数据。如本模型所述 AWS ，负责保护运行所有网络的全球基础设施 AWS Cloud。您负责维护对托管在此基础结构上的内容的控制。您还负责您所使用的 AWS 服务 的安全配置和管理任务。有关数据隐私的更多信息，请参阅数据隐私常见问题。有关欧洲数据保护的信息，请参阅通用数据保护条例（GDPR）中心。

出于数据保护的目的，我们建议您保护 AWS 账户 证书并使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 设置个人用户。这样，每个用户只获得履行其工作职责所需的权限。还建议您通过以下方式保护数据：

- 对每个账户使用多重身份验证（MFA）。
- 用于 SSL/TLS 与 AWS 资源通信。我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 使用设置 API 和用户活动日志 AWS CloudTrail。有关使用 CloudTrail 跟踪捕获 AWS 活动的信息，请参阅AWS CloudTrail 用户指南中的使用跟 CloudTrail 踪。
- 使用 AWS 加密解决方案以及其中的所有默认安全控件 AWS 服务。
- 使用高级托管安全服务（例如 Amazon Macie），它有助于发现和保护存储在 Amazon S3 中的敏感数据。
- 如果在 AWS 通过命令行接口或 API 进行访问时需要经过 FIPS 140-3 验证的加密模块，请使用 FIPS 终端节点。有关可用的 FIPS 端点的更多信息，请参阅《美国联邦信息处理标准（FIPS）第 140-3 版》https://aws.amazon.com/compliance/fips/。

强烈建议您切勿将机密信息或敏感信息（如您客户的电子邮件地址）放入标签或自由格式文本字段（如名称字段）。这包括当您 AWS 服务 使用控制台 AWS CLI、API MediaTailor 或 AWS SDK 或其他方式使用时。在用于名称的标签或自由格式文本字段中输入的任何数据都可能会用于计费或诊断日志。如果您向外部服务器提供 URL，强烈建议您不要在网址中包含凭证信息来验证对该服务器的请求。

数据加密

AWS Elemental MediaTailor 在其内容清单管理或与服务器、CDN 或玩家的通信中，不会加密或解密数据。 MediaTailor 不需要您提供任何客户数据或其他敏感信息。

不要将敏感信息（例如客户账号、信用卡信息或登录凭证）放入自由格式字段或查询参数中。这适用于的所有用途 AWS Elemental MediaTailor，包括控制台、API、软件开发工具包和 AWS Command Line Interface (AWS CLI)。您输入到该服务的任何数据都可能被选取以包含在诊断日志中。

当您向外部服务器提供 URL 时，请勿在 URL 中包含未加密的凭证信息来验证您对该服务器的请求。

的身份和访问管理 AWS Elemental MediaTailor

AWS Identity and Access Management (IAM) AWS 服务 可以帮助管理员安全地控制对 AWS 资源的访问权限。IAM 管理员控制谁可以进行身份验证（登录）和授权（有权限）使用 MediaTailor 资源。IAM AWS 服务 是您可以免费使用的。

主题

- 受众
- 使用身份进行身份验证
- 使用策略管理访问
- 操作方法 AWS Elemental MediaTailor 与 IAM 一起使用
- Identity-based 的策略示例 AWS Elemental MediaTailor
- Resource-based 的策略示例 AWS Elemental MediaTailor
- AWS 的托管策略 AWS Elemental MediaTailor
- 将服务相关角色用于 MediaTailor
- 问题排查 AWS Elemental MediaTailor 身份和访问权限

受众

您的使用方式 AWS Identity and Access Management (IAM) 因您的角色而异：

- 服务用户：如果您无法访问功能，请从管理员处请求权限（请参阅问题排查 AWS Elemental MediaTailor 身份和访问权限）
- 服务管理员：确定用户访问权限并提交权限请求（请参阅操作方法 AWS Elemental MediaTailor 与 IAM 一起使用）
- IAM 管理员：编写用于管理访问权限的策略（请参阅Identity-based 的策略示例 AWS Elemental MediaTailor）

使用身份进行身份验证

身份验证是您 AWS 使用身份凭证登录的方式。您必须以 AWS 账户根用户、IAM 用户身份或通过代入 IAM 角色进行身份验证。

您可以使用来自 AWS IAM Identity Center（IAM 身份中心）、单点登录身份验证或证书等身份源的 Google/Facebook 证书以联合身份身份登录。有关登录的更多信息，请参阅《AWS 登录 用户指南》中的如何登录您的 AWS 账户。

对于编程访问， AWS 提供 SDK 和 CLI 以加密方式签署请求。有关更多信息，请参阅《IAM 用户指南》中的适用于 API 请求的AWS 签名版本 4。

AWS 账户 根用户

创建时 AWS 账户，首先要使用一个名为 AWS 账户 root 用户的登录身份，该身份可以完全访问所有资源 AWS 服务 和资源。我们强烈建议不要使用根用户进行日常任务。有关需要根用户凭证的任务，请参阅《IAM 用户指南》中的需要根用户凭证的任务。

联合身份

最佳做法是，要求人类用户使用联合身份提供商来 AWS 服务 使用临时证书进行访问。

联合身份是来自您的企业目录、Web 身份提供商的用户 Directory Service ，或者 AWS 服务 使用来自身份源的证书进行访问的用户。联合身份代入可提供临时凭证的角色。

要集中管理访问权限，建议使用。 AWS IAM Identity Center有关更多信息，请参阅《AWS IAM Identity Center 用户指南》中的什么是 IAM Identity Center？。

IAM 用户和群组

IAM 用户是对某个人员或应用程序具有特定权限的一个身份。建议使用临时凭证，而非具有长期凭证的 IAM 用户。有关更多信息，请参阅 IAM 用户指南中的要求人类用户 AWS 使用联合身份提供商以使用临时证书进行访问。

IAM 组指定一组 IAM 用户，便于更轻松地对大量用户进行权限管理。有关更多信息，请参阅《IAM 用户指南》中的 IAM 用户使用案例。

IAM 角色

IAM 角色是具有特定权限的身份，可提供临时凭证。您可以通过从用户切换到 IAM 角色（控制台）或调用 AWS CLI 或 AWS API 操作来代入角色。有关更多信息，请参阅《IAM 用户指南》中的担任角色的方法。

IAM 角色对于联合用户访问、临时 IAM 用户权限、跨账户访问、跨服务访问以及在 Amazon EC2 上运行的应用程序非常有用。有关更多信息，请参阅《IAM 用户指南》中的 IAM 中的跨账户资源访问。

使用策略管理访问

您可以 AWS 通过创建策略并将其附加到 AWS 身份或资源来控制中的访问权限。策略定义了与身份或资源关联时的权限。 AWS 委托人提出请求时评估这些政策。大多数策略以 JSON 文档的 AWS 形式存储在。有关 JSON 策略文档的更多信息，请参阅《IAM 用户指南》中的 JSON 策略概述。

管理员使用策略，通过定义哪个主体可以在什么条件下对哪些资源执行哪些操作来指定谁有权访问什么。

默认情况下，用户和角色没有权限。IAM 管理员创建 IAM 策略并将其添加到角色中，然后用户可以担任这些角色。IAM 策略定义权限，与执行操作所用的方法无关。

Identity-based 政策

Identity-based 策略是您附加到身份（用户、群组或角色）的 JSON 权限策略文档。这些策略控制身份可以执行什么操作、对哪些资源执行以及在什么条件下执行。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的使用客户管理型策略定义自定义 IAM 权限。

Identity-based 策略可以是内联策略（直接嵌入到单个身份中）或托管策略（附加到多个身份的独立策略）。要了解如何在托管策略和内联策略之间进行选择，请参阅《IAM 用户指南》中的在托管策略与内联策略之间进行选择。

Resource-based 政策

Resource-based 策略是您附加到资源的 JSON 策略文档。示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。您必须在基于资源的策略中指定主体。

Resource-based 策略是位于该服务中的内联策略。您不能在基于资源的策略中使用 IAM 的 AWS 托管策略。

其他策略类型

AWS 支持其他策略类型，这些策略类型可以设置更常见的策略类型授予的最大权限：

- 权限边界 – 设置基于身份的策略可以授予 IAM 实体的最大权限。有关更多信息，请参阅《 IAM 用户指南》中的 IAM 实体的权限边界。
- 服务控制策略（SCP）– 指定 AWS Organizations中组织或组织单元的最大权限。有关更多信息，请参阅《AWS Organizations 用户指南》中的服务控制策略。
- 资源控制策略（RCP）– 设置对账户中资源的最大可用权限。有关更多信息，请参阅《AWS Organizations 用户指南》中的资源控制策略（RCP）。
- 会话策略 – 在为角色或联合用户创建临时会话时，作为参数传递的高级策略。有关更多信息，请参阅《IAM 用户指南》中的会话策略。

多个策略类型

当多个类型的策略应用于一个请求时，生成的权限更加复杂和难以理解。要了解如何在涉及多种策略类型时 AWS 确定是否允许请求，请参阅 IAM 用户指南中的策略评估逻辑。

操作方法 AWS Elemental MediaTailor 与 IAM 一起使用

在使用 IAM 管理访问权限之前 MediaTailor，请了解哪些 IAM 功能可供使用 MediaTailor。

您可以与之配合使用的 IAM 功能 AWS Elemental MediaTailor

IAM 功能	MediaTailor 支持
Identity-based 政策	是
Resource-based 政策	是
策略操作	是
策略资源	否
策略条件键（特定于服务）	是
ACL	否
ABAC（策略中的标签）	部分
临时凭证	是
主体权限	是
服务角色	否
Service-linked 角色	是

要详细了解大多数 IAM 功能 MediaTailor 以及其他 AWS 服务的工作原理，请参阅 IAM 用户指南中的与 IAM 配合使用的AWS 服务。

Identity-based 的政策 MediaTailor

支持基于身份的策略：是

Identity-based 策略是 JSON 权限策略文档，您可以将其附加到身份，例如 IAM 用户、用户组或角色。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的使用客户管理型策略定义自定义 IAM 权限。

通过使用 IAM 基于身份的策略，您可以指定允许或拒绝的操作和资源以及允许或拒绝操作的条件。要了解可在 JSON 策略中使用的所有元素，请参阅《IAM 用户指南》中的 IAM JSON 策略元素引用。

Identity-based 的策略示例 MediaTailor

要查看 MediaTailor 基于身份的策略的示例，请参阅。Identity-based 的策略示例 AWS Elemental MediaTailor

Resource-based 内部的政策 MediaTailor

支持基于资源的策略：是

该 MediaTailor 服务仅支持一种基于资源的策略。之所以称之为频道政策，是因为它附加到一个频道。该政策定义了哪些委托人可以在频道上执行操作。

Resource-based 策略是您附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。您必须在基于资源的策略中指定主体。委托人可以包括账户、用户、角色、联合用户或 AWS 服务。

要启用跨账户访问，您可以将整个账户或其它账户中的 IAM 实体指定为基于资源的策略中的主体。有关更多信息，请参阅《IAM 用户指南》中的 IAM 中的跨账户资源访问。

要了解如何将基于资源的政策附加到频道，请参阅使用 MediaTailor 控制台创建频道。

Resource-based 中的政策示例 MediaTailor

要查看 MediaTailor 基于资源的策略的示例，请参阅Resource-based 的策略示例 AWS Elemental MediaTailor。

的政策行动 MediaTailor

支持策略操作：是

管理员可以使用 AWS JSON 策略来指定谁有权访问哪些内容。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

JSON 策略的 Action 元素描述可用于在策略中允许或拒绝访问的操作。在策略中包含操作以授予执行关联操作的权限。

要查看 MediaTailor 操作列表，请参阅《服务授权参考》 AWS Elemental MediaTailor中定义的操作。

正在进行的策略操作在操作之前 MediaTailor 使用以下前缀：

```
mediatailor
```

要在单个语句中指定多项操作，请使用逗号将它们隔开。

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

要查看 MediaTailor 基于身份的策略的示例，请参阅。Identity-based 的策略示例 AWS Elemental MediaTailor

的政策资源 MediaTailor

支持策略资源：否

AWS Elemental MediaTailor 不支持在策略中指定资源 ARN。

MediaTailor 的策略条件键

支持特定于服务的策略条件键：是

管理员可以使用 AWS JSON 策略来指定谁有权访问哪些内容。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

Condition 元素根据定义的条件指定语句何时执行。您可以创建使用条件运算符（例如，等于或小于）的条件表达式，以使策略中的条件与请求中的值相匹配。要查看所有 AWS 全局条件密钥，请参阅 IAM 用户指南中的AWS 全局条件上下文密钥。

有关 MediaTailor 条件密钥的列表，请参阅《服务授权参考》 AWS Elemental MediaTailor中的条件密钥。要了解可以在哪些操作和资源中使用条件键，请参阅操作定义者 AWS Elemental MediaTailor。

AWS Elemental MediaTailor 不提供特定服务的条件密钥，但它支持使用一些全局条件密钥。要查看所有 AWS 全局条件键，请参阅《AWS Identity and Access Management 用户指南》中的AWS 全局条件上下文密钥。

ACL 已加入 MediaTailor

支持 ACL：否

访问控制列表（ACL）控制哪些主体（账户成员、用户或角色）有权访问资源。ACL 与基于资源的策略类似，但它们不使用 JSON 策略文档格式。

ABAC 和 MediaTailor

支持 ABAC（策略中的标签）：部分支持

Attribute-based 访问控制 (ABAC) 是一种授权策略，它根据称为标签的属性定义权限。您可以将标签附加到 IAM 实体和 AWS 资源，然后设计 ABAC 策略，允许在委托人的标签与资源上的标签匹配时进行操作。

要基于标签控制访问，您需要使用 `aws:ResourceTag/key-name`aws:RequestTag/*key-name* 或 `aws:TagKeys` 条件键在策略的条件元素中提供标签信息。

如果某个服务对于每种资源类型都支持所有这三个条件键，则对于该服务，该值为是。如果某个服务仅对于部分资源类型支持所有这三个条件键，则该值为部分。

有关 ABAC 的更多信息，请参阅《IAM 用户指南》中的使用 ABAC 授权定义权限。要查看设置 ABAC 步骤的教程，请参阅《IAM 用户指南》中的使用基于属性的访问权限控制（ABAC）。

对于 MediaTailor，使用值 P artial 。

使用临时证书 MediaTailor

支持临时凭证：是

临时证书提供对 AWS 资源的短期访问权限，并在您使用联合或切换角色时自动创建。 AWS 建议您动态生成临时证书，而不是使用长期访问密钥。有关更多信息，请参阅《IAM 用户指南》中的 IAM 中的临时安全凭证和使用 IAM 的。AWS 服务

Cross-service 的主要权限 MediaTailor

支持转发访问会话（FAS）：是

前向访问会话 (FAS) 使用调用主体的权限 AWS 服务，同时请求 AWS 服务 向下游服务发出请求。有关发出 FAS 请求时的策略详情，请参阅转发访问会话。

的服务角色 MediaTailor

支持服务角色：否

AWS Elemental MediaTailor 不支持服务角色。

Service-linked 的角色 MediaTailor

支持服务关联角色：是

服务相关角色是一种链接到的服务角色。 AWS 服务该服务可以代入代表您执行操作的角色。 Service-linked 角色显示在您的中 AWS 账户 且归服务所有。IAM 管理员可以查看但不能编辑服务关联角色的权限。

有关创建或管理 MediaTailor 服务相关角色的详细信息，请参阅将服务相关角色用于 MediaTailor。

Identity-based 的策略示例 AWS Elemental MediaTailor

默认情况下，用户和角色没有创建或修改 MediaTailor 资源的权限。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM 策略。

要了解如何使用这些示例 JSON 策略文档创建基于 IAM 身份的策略，请参阅《IAM 用户指南》中的创建 IAM 策略（控制台）。

有关由定义的操作和资源类型的详细信息 MediaTailor，包括每种资源类型的 ARN 格式，请参阅服务授权参考 AWS Elemental MediaTailor中的操作、资源和条件密钥。

主题

- 策略最佳实践
- 使用控制 MediaTailor 台
- 允许用户查看他们自己的权限

策略最佳实践

Identity-based 策略决定是否有人可以在您的账户中创建、访问或删除 MediaTailor 资源。这些操作可能会使 AWS 账户产生成本。创建或编辑基于身份的策略时，请遵循以下指南和建议：

- 开始使用 AWS 托管策略并转向最低权限权限 — 要开始向用户和工作负载授权，请使用为许多常见用例授予权限的AWS 托管策略。它们可以在您的 AWS 账户. 我们建议您通过定义特定于您的用例的 AWS 客户管理策略来进一步减少权限。有关更多信息，请参阅《IAM 用户指南》中的 AWS 托管策略或工作职能的AWS 托管策略。
- 应用最低权限：在使用 IAM 策略设置权限时，请仅授予执行任务所需的权限。为此，您可以定义在特定条件下可以对特定资源执行的操作，也称为最低权限许可。有关使用 IAM 应用权限的更多信息，请参阅《IAM 用户指南》中的 IAM 中的策略和权限。
- 使用 IAM 策略中的条件进一步限制访问权限：您可以向策略添加条件来限制对操作和资源的访问。例如，您可以编写策略条件来指定必须使用 SSL 发送所有请求。如果服务操作是通过特定方式使用的，则还可以使用条件来授予对服务操作的访问权限 AWS 服务，例如 CloudFormation。有关更多信息，请参阅《IAM 用户指南》中的 IAM JSON 策略元素：条件。
- 使用 IAM Access Analyzer 验证您的 IAM 策略，以确保权限的安全性和功能性：IAM Access Analyzer 会验证新策略和现有策略，以确保策略符合 IAM 策略语言（JSON）和 IAM 最佳实践。IAM Access Analyzer 提供 100 多项策略检查和可操作的建议，以帮助您制定安全且功能性强的策略。有关更多信息，请参阅《IAM 用户指南》中的使用 IAM Access Analyzer 验证策略。
- 需要多重身份验证 (MFA) — 如果您的场景需要 IAM 用户或根用户 AWS 账户，请启用 MFA 以提高安全性。若要在调用 API 操作时需要 MFA，请将 MFA 条件添加到您的策略中。有关更多信息，请参阅《IAM 用户指南》中的使用 MFA 保护 API 访问。

有关 IAM 中的最佳实操的更多信息，请参阅《IAM 用户指南》中的 IAM 中的安全最佳实践。

使用控制 MediaTailor 台

要访问 AWS Elemental MediaTailor 控制台，您必须拥有一组最低权限。这些权限必须允许您在中列出和查看有关 MediaTailor 资源的详细信息 AWS 账户。如果创建比必需的最低权限更为严格的基于身份的策略，对于附加了该策略的实体（用户或角色），控制台将无法按预期正常运行。

您无需为仅调用 AWS CLI 或 AWS API 的用户提供最低控制台权限。相反，只允许访问与其尝试执行的 API 操作相匹配的操作。

为确保用户和角色仍然可以使用 MediaTailor 控制台，还要将 MediaTailor*ConsoleAccess*或*ReadOnly* AWS 托管策略附加到实体。有关更多信息，请参阅《IAM 用户指南》中的为用户添加权限。

允许用户查看他们自己的权限

该示例说明了您如何创建策略，以允许 IAM 用户查看附加到其用户身份的内联和托管式策略。此策略包括在控制台上完成此操作的权限，或使用 AWS CLI 或 AWS API 以编程方式完成此操作的权限。

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

Resource-based 的策略示例 AWS Elemental MediaTailor

要了解如何将基于资源的政策附加到频道，请参阅使用 MediaTailor 控制台创建频道。

主题

- 匿名访问
- Cross-account 访问

匿名访问

考虑以下Allow政策。此政策生效后， MediaTailor允许匿名访问政策中对频道资源的mediatailor:GetManifest操作。这发生在哪里 *region* AWS 区域，*accountID*是你的 AWS 账户 ID，*channelName*是频道资源的名称。

Cross-account 访问

考虑以下Allow政策。此政策生效后， MediaTailor允许跨账户对政策中的渠道资源进行mediatailor:GetManifest操作。这发生在哪里 *region* AWS 区域，*accountID*是你的 AWS 账户 ID，*channelName*是频道资源的名称。

AWS 的托管策略 AWS Elemental MediaTailor

要为用户、群组和角色添加权限，使用 AWS 托管策略比自己编写策略更容易。创建仅为团队提供所需权限的 IAM 客户管理型策略需要时间和专业知识。要快速入门，您可以使用我们的 AWS 托管策略。这些策略涵盖常见使用案例，可在您的 AWS 账户中使用。有关 AWS 托管策略的更多信息，请参阅 IAM 用户指南中的AWS 托管策略。

AWS 服务维护和更新 AWS 托管策略。您无法更改 AWS 托管策略中的权限。服务偶尔会向 AWS 托管式策略添加额外权限以支持新特征。此类更新会影响附加策略的所有身份（用户、组和角色）。当启动新特征或新操作可用时，服务最有可能会更新 AWS 托管式策略。服务不会从 AWS 托管策略中移除权限，因此策略更新不会破坏您的现有权限。

此外， AWS 支持跨多个服务的工作职能的托管策略。例如，ReadOnlyAccess AWS 托管策略提供对所有 AWS 服务和资源的只读访问权限。当服务启动新功能时， AWS 会为新操作和资源添加只读权限。有关工作职能策略的列表和说明，请参阅 IAM 用户指南中的适用于工作职能的AWS 托管式策略。

AWS 管理策略： AWSElementalMediaTailorFullAccess

您可以将 AWSElementalMediaTailorFullAccess 策略附加到 IAM 身份。它对于需要创建和管理回放配置和频道汇编资源（例如节目和频道）的用户非常有用。此策略授予允许完全访问的权限 AWS Elemental MediaTailor。这些用户可以创建、更新和删除 MediaTailor 资源。

要查看此策略的权限，请参阅《AWS 托管式策略参考》中的 AWSElementalMediaTailorFullAccess。

AWS 管理策略： AWSElementalMediaTailorReadOnly

您可以将 `AWSElementalMediaTailorReadOnly` 策略附加到 IAM 身份。它对于需要查看播放配置和频道汇编资源（例如节目和频道）的用户很有用。此策略授予的权限允许以只读方式访问 AWS Elemental MediaTailor。这些用户无法创建、更新或删除 MediaTailor 资源：

要查看此策略的权限，请参阅《AWS 托管式策略参考》中的 AWSElementalMediaTailorReadOnly。

MediaTailor 更新到 AWS 托管策略

查看 MediaTailor 自该服务开始跟踪这些更改以来 AWS 托管策略更新的详细信息。要获得有关此页面更改的自动提示，请订阅 MediaTailor 的文档历史记录 AWS Elemental MediaTailor 上的 RSS 源。

更改	描述	日期
MediaTailor 添加了新的托管策略	MediaTailor 添加了以下托管策略： • AWSElementalMediaTailorReadOnly • AWSElementalMediaTailorFullAccess	2021 年 11 月 24 日
MediaTailor 已开始跟踪更改	MediaTailor 开始跟踪其 AWS 托管策略的更改。	2021 年 11 月 24 日

将服务相关角色用于 MediaTailor

AWS Elemental MediaTailor 使用 AWS Identity and Access Management (IAM) 服务相关角色。服务相关角色是一种独特的 IAM 角色类型，可直接关联到 MediaTailor。 Service-linked 角色由预定义 MediaTailor ，包括该服务代表您调用其他 AWS 服务所需的所有权限。

服务相关角色使设置变得 MediaTailor 更加容易，因为您无需手动添加必要的权限。 MediaTailor 定义其服务相关角色的权限，除非另有定义，否则 MediaTailor 只能代入其角色。定义的权限包括信任策略和权限策略，以及不能附加到任何其他 IAM 实体的权限策略。

只有在首先删除相关资源后，您才能删除服务关联角色。这可以保护您的 MediaTailor 资源，因为您无法无意中删除访问资源的权限。

有关支持服务相关角色的其它服务的信息，请参阅使用 IAM 的AWS 服务并查找Service-Linked 角色列中显示为是的服务。请选择是与查看该服务的服务关联角色文档的链接。

MediaTailor 服务相关角色的支持区域

MediaTailor 支持在提供服务的所有地区使用服务相关角色。有关更多信息，请参阅AWS 区域和端点。

主题

- Service-linked 的角色权限 MediaTailor
- 为创建服务相关角色 MediaTailor
- 编辑的服务相关角色 MediaTailor
- 删除的服务相关角色 MediaTailor

Service-linked 的角色权限 MediaTailor

MediaTailor 使用名为的服务相关角色 AWSServiceRoleForMediaTailor — MediaTailor 使用此服务相关角色调用 CloudWatch 以创建和管理日志组、日志流和日志事件。此服务相关角色附加到以下托管策略：`AWSMediaTailorServiceRolePolicy`。

AWSServiceRoleForMediaTailor 服务相关角色信任以下服务来代入该角色：

- `mediatailor.amazonaws.com`

角色权限策略 MediaTailor 允许对指定资源完成以下操作：

- 操作：`arn:aws:logs:*:*:log-group:/aws/MediaTailor/*:log-stream:*` 上的 `logs:PutLogEvents`
- 操作：`arn:aws:logs:*:*:log-group:/aws/MediaTailor/*` 上的 `logs:CreateLogStream, logs:CreateLogGroup, logs:DescribeLogGroups, logs:DescribeLogStreams`

必须配置权限，允许 IAM 实体（如用户、组或角色）创建、编辑或删除服务相关角色。有关更多信息，请参阅 IAM 用户指南中的Service-Linked 角色权限。

为创建服务相关角色 MediaTailor

您无需手动创建服务关联角色。当您在 AWS 管理控制台、 AWS Command Line Interface (AWS CLI) 或 AWS API 中启用会话登录时， MediaTailor 会为您创建服务相关角色。

> **Important**
>
> 如果您在其他使用此角色支持的功能的服务中完成某个操作，此服务关联角色可以出现在您的账户中。此外，如果您在 2021 年 9 月 15 日开始支持服务相关角色之前使用该 MediaTailor 服务，则会在您的账户中 MediaTailor 创建该 AWSServiceRoleForMediaTailor 角色。要了解更多信息，请参阅我的 IAM 账户中的新角色。

如果您删除该服务关联角色，然后需要再次创建，您可以使用相同流程在账户中重新创建此角色。当您启用会话日志记录时， MediaTailor 会再次为您创建服务相关角色。

您也可以使用 IAM 控制台为 MediaTailor 使用案例创建服务相关角色。在 AWS CLI 或 AWS API 中，使用服务名称创建服务相关角色。`mediatailor.amazonaws.com`有关更多信息，请参阅 IAM 用户指南中的创建 Service-Linked 角色。如果您删除了此服务相关角色，可以使用同样的过程再次创建角色。

编辑的服务相关角色 MediaTailor

MediaTailor 不允许您编辑 AWSServiceRoleForMediaTailor 服务相关角色。创建服务关联角色后，您将无法更改角色的名称，因为可能有多种实体引用该角色。但是可以使用 IAM 编辑角色描述。有关更多信息，请参阅 IAM 用户指南中的编辑 Service-Linked 角色。

删除的服务相关角色 MediaTailor

如果不再需要使用某个需要服务关联角色的功能或服务，我们建议您删除该角色。这样就没有未被主动监控或维护的未使用实体。但是，必须先清除服务相关角色的资源，然后才能手动删除它。

> **Note**
>
> 如果您尝试清理资源时 MediaTailor 服务正在使用该角色，则删除可能会失败。如果发生这种情况，请等待几分钟后重试。

清理使用的 MediaTailor 资源 AWSServiceRoleForMediaTailor

- 在删除 MediaTailor 为日志配置创建的服务相关角色之前，必须先停用账户中的所有日志配置。要停用日志配置，请将启用百分比值设置为 0。这将关闭所有记录相应播放配置的会话。有关更多信息，请参阅 停用日志配置。

使用 IAM 手动删除服务关联角色

使用 IAM 控制台、 AWS Command Line Interface (AWS CLI) 或 AWS API 删除 AWSServiceRoleForMediaTailor服务相关角色。有关更多信息，请参阅 IAM 用户指南中的删除 Service-Linked 角色。

问题排查 AWS Elemental MediaTailor 身份和访问权限

使用以下信息来帮助您诊断和修复在使用 MediaTailor 和 IAM 时可能遇到的常见问题。

主题

- 我无权在中执行操作 MediaTailor
- 我无权执行 iam：PassRole
- 我想允许我以外的人进来 AWS 账户 访问我的 MediaTailor 资源
- MediaTailor 无法访问我的 Amazon S3 存储桶

我无权在中执行操作 MediaTailor

如果您收到错误提示，指明您无权执行某个操作，则必须更新策略以允许执行该操作。

当 `mateojackson` IAM 用户尝试使用控制台查看有关虚构 *`my-example-widget`* 资源的详细信息，但不拥有虚构 `mediatailor:`*`GetWidget`* 权限时，会发生以下示例错误。

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

在此情况下，必须更新 `mateojackson` 用户的策略，以允许使用 `mediatailor:`*`GetWidget`* 操作访问 *`my-example-widget`* 资源。

如果需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

我无权执行 iam：PassRole

如果您收到一个错误，表明您无权执行 `iam:PassRole` 操作，则必须更新策略以允许您将角色传递给。 MediaTailor

有些 AWS 服务 允许您将现有角色传递给该服务，而不是创建新的服务角色或服务相关角色。为此，您必须具有将角色传递到服务的权限。

当名为 `marymajor` 的 IAM 用户尝试使用控制台在 MediaTailor 中执行操作时，会发生以下示例错误。但是，服务必须具有服务角色所授予的权限才可执行此操作。Mary 不具有将角色传递到服务的权限。

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

在这种情况下，必须更新 Mary 的策略以允许她执行 `iam:PassRole` 操作。

如果需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

我想允许我以外的人进来 AWS 账户 访问我的 MediaTailor 资源

您可以创建一个角色，以便其他账户中的用户或您组织外的人员可以使用该角色来访问您的资源。您可以指定谁值得信赖，可以代入角色。对于支持基于资源的策略或访问控制列表（ACL）的服务，您可以使用这些策略向人员授予对您的资源的访问权。

要了解更多信息，请参阅以下内容：

- 要了解是否 MediaTailor 支持这些功能，请参阅操作方法 AWS Elemental MediaTailor 与 IAM 一起使用。
- 要了解如何提供对您所拥有资源的访问权限，请参阅 IAM 用户指南中的向您拥有的另一 AWS 账户 个 IAM 用户提供访问权限。 AWS 账户
- 要了解如何向第三方提供对资源的访问权限 AWS 账户，请参阅 IAM 用户指南中的向第三方 AWS 账户 拥有的提供访问权限。
- 要了解如何通过身份联合验证提供访问权限，请参阅《IAM 用户指南》中的为经过外部身份验证的用户（身份联合验证）提供访问权限。
- 要了解使用角色和基于资源的策略进行跨账户访问之间的差别，请参阅《IAM 用户指南》中的 IAM 中的跨账户资源访问。

MediaTailor 无法访问我的 Amazon S3 存储桶

如果在尝试访问您的 Amazon S3 存储桶中的内容时 MediaTailor 返回错误，请检查以下常见原因：

Amazon S3 权限问题

错误症状：

- `502 BadGatewayException`包含与 Amazon S3 相关的消息
- `403 Forbidden` MediaTailor 日志中的错误
- 无法从 Amazon S3 源加载内容

解决方法：

1. 验证您的 MediaTailor 服务角色是否具有所需的 Amazon S3 权限：
2. 检查您的 Amazon S3 存储桶政策是否允许 MediaTailor 访问：

HTTPS 访问问题

错误症状：

- SSL/TLS 握手失败
- `502 Bad Gateway`当访问 HTTPS 亚马逊 S3 网址时

解决方法：

1. 确保您的 Amazon S3 存储桶策略需要 HTTPS：
2. 验证 MediaTailor 配置使用 HTTPS 网址获取 Amazon S3 内容。

Cross-account 亚马逊 S3 访问权限

错误症状：

- `Access Denied`Amazon S3 存储桶位于其他账户时出错
- MediaTailor 无法担任跨账户角色

解决方法：

1. 将内容账户中的 Amazon S3 存储桶策略更新为信任 MediaTailor的服务角色
2. 向的服务角色添加跨账户权限 MediaTailor
3. 确保两个账户都配置了正确的信任关系

的合规性验证 AWS Elemental MediaTailor

我们的新 AWS 注册体验不是为受监管的工作负载设计的。如果您正在使用我们的全新 AWS 注册体验，但想 AWS 用于受监管的工作负载，则可以注册 AWS （高级）或激活 AWS 环境的高级功能。

要了解是否属于特定合规计划的范围，请参阅 “按合规计划 AWS 服务 中的 “按范围划分的合规计划”，然后选择您感兴趣的合规计划。 AWS 服务 有关一般信息，请参阅AWS 合规计划。

您可以使用下载第三方审计报告 AWS Artifact。有关更多信息，请参阅中的 “下载报告” 中的 “ AWS Artifact。

您在使用 AWS 服务 时的合规责任取决于数据的敏感性、贵公司的合规目标以及适用的法律法规。有关您在使用时的合规责任的更多信息 AWS 服务，请参阅AWS 安全文档。

弹性在 AWS Elemental MediaTailor

AWS 全球基础设施围绕 AWS 区域 可用区构建。 AWS 区域 提供多个物理上分隔和隔离的可用区，这些可用区通过低延迟、高吞吐量和高度冗余的网络相连。利用可用区，您可以设计和操作在可用区之间无中断地自动实现失效转移的应用程序和数据库。与传统的单个或多个数据中心基础设施相比，可用区具有更高的可用性、容错能力和可扩展性。

有关 AWS 区域 和可用区的更多信息，请参阅AWS 全球基础设施。

除了 AWS 全球基础架构外，还 MediaTailor 提供多种功能来帮助支持您的数据弹性和备份需求。

中的基础设施安全 MediaTailor

作为一项托管服务 AWS Elemental MediaTailor ，受 AWS 全球网络安全保护。有关 AWS 安全服务以及如何 AWS 保护基础设施的信息，请参阅AWS 云安全。要使用基础设施安全的最佳实践来设计您的 AWS 环境，请参阅安全支柱架 AWS 构完善框架中的基础设施保护。

您可以使用 AWS 已发布的 API 调用 MediaTailor 通过网络进行访问。客户端必须支持以下内容：

- 传输层安全性协议（TLS）。我们要求使用 TLS 1.2，建议使用 TLS 1.3。

- 具有完美向前保密 (PFS) 的密码套件，例如 DHE（短暂的）或 ECDHE（椭圆曲线短暂的 Diffie-Hellman）。 Diffie-Hellman大多数现代系统（如 Java 7 及更高版本）都支持这些模式。

您可以从任何网络位置调用这些 API 操作， MediaTailor 但支持基于资源的访问策略，其中可能包括基于源 IP 地址的限制。您还可以使用 MediaTailor 策略来控制来自特定亚马逊虚拟私有云 (Amazon VPC) 终端节点或特定 VPC 的访问。实际上，这仅将网络对给定 MediaTailor 资源的访问与网络中的特定 VPC 隔离开来。 AWS

Cross-service 困惑的副手预防

混淆代理问题是一个安全性问题，即不具有某操作执行权限的实体可能会迫使具有更高权限的实体执行该操作。在中 AWS，跨服务模仿可能会导致副手混乱的问题。 Cross-service 当一项服务（呼叫服务）调用另一项服务（被叫服务）时，可能会发生模拟。可以操纵调用服务，使用其权限以在其他情况下该服务不应有访问权限的方式对另一个客户的资源进行操作。为防止这种情况， AWS 提供可帮助您保护所有服务的数据的工具，而这些服务中的服务主体有权限访问账户中的资源。

我们建议在资源策略中使用 SourceArn a ws: 和 aws: SourceAccount 全局条件上下文密钥来限制向资源 AWS Elemental MediaTailor 提供其他服务的权限。如果使用两个全局条件上下文键，在同一策略语句中使用时，`aws:SourceAccount` 值和 `aws:SourceArn` 值中的账户必须使用相同的账户 ID。

的值`aws:SourceArn`必须是在您的地区和账户中 MediaTailor 发布 CloudWatch 日志的播放配置。但是，仅当您使用允许将亚马逊 CloudWatch 日志 MediaTailor 发布到您的账户的`MediaTailorLogger`角色时，这才适用。如果您使用服务相关角色来 MediaTailor 发布 CloudWatch 日志，则这不适用。

防范混淆代理问题最有效的方法是使用 `aws:SourceArn` 全局条件上下文键和资源的完整 ARN。如果不知道资源的完整 ARN，或者正在指定多个资源，请针对 ARN 未知部分使用带有通配符（*）的 `aws:SourceArn` 全局上下文条件键。例如 `arn:aws:`*`servicename`*`::`*`123456789012`*`:*`。

以下示例说明如何在中使用`aws:SourceArn`和`aws:SourceAccount`全局条件上下文键 MediaTailor 来防止出现混乱的代理问题。

登录和监控 MediaTailor

本部分概括介绍了 AWS Elemental MediaTailor 中出于安全目的而提供的日志记录和监控选项。有关登录和监控的更多信息， MediaTailor 请参阅监控和标记 AWS Elemental MediaTailor 资源。

监控是维护 AWS 解决方案的可靠性、可用性和性能的重要组成部分。 AWS Elemental MediaTailor 您应该收集 AWS 解决方案所有部分的监控数据，这样在出现多点故障时可以更轻松地进行调试。 AWS 提供了多种用于监控 MediaTailor 资源和响应潜在事件的工具：

亚马逊 CloudWatch 警报

使用 CloudWatch 警报，您可以在指定的时间段内监视单个指标。如果指标超过给定阈值，则会向 Amazon SNS 主题或 A AWS uto Scaling 策略发送通知。 CloudWatch 警报不会调用操作，因为它们处于特定状态。而是必须在状态已改变并在指定的若干个时间段内保持不变后才调用。有关更多信息，请参阅 the section called “使用 CloudWatch 指标进行监控”。

AWS CloudTrail logs

CloudTrail 提供用户、角色或 AWS 服务在中采取的操作的记录 AWS Elemental MediaTailor。使用收集的信息 CloudTrail，您可以确定向谁发出了请求 MediaTailor、发出请求的 IP 地址、谁提出了请求、何时提出请求以及其他详细信息。有关更多信息，请参阅 记录 AWS Elemental MediaTailor API 调用。

AWS Trusted Advisor

Trusted Advisor 借鉴了从为数十万 AWS 客户提供服务中学到的最佳实践。 Trusted Advisor 检查您的 AWS 环境，然后在有机会节省资金、提高系统可用性和性能或帮助弥合安全漏洞时提出建议。所有 AWS 客户均有权访问五个 Trusted Advisor 检查。使用“商业”和“企业”支持计划的客户可以查看所有 Trusted Advisor 检查。

有关更多信息，请参阅 AWS Trusted Advisor。

监控和标记 AWS Elemental MediaTailor 资源

监控是维护其他 AWS 解决方案的可靠性、可用性和性能的重要组成部分。 AWS Elemental MediaTailor AWS 提供以下监控工具供您观察 MediaTailor、报告出现问题并在适当时自动采取措施：

- Amazon 实时 CloudWatch监控您的 AWS 资源和您运行 AWS 的应用程序。您可以收集和跟踪指标，创建自定义的控制面板，以及设置警报以在指定的指标达到您指定的阈值时通知您或采取措施。例如，您可以 CloudWatch 跟踪您的 Amazon EC2 实例的 CPU 使用率或其他指标，并在需要时自动启动新实例。有关更多信息，请参阅亚马逊 CloudWatch 用户指南。
- Amazon L CloudWatch ogs 使您能够监控、存储和访问与广告决策服务器 (ADS) 的所有互动中的日志文件。 AWS Elemental MediaTailor 发布广告请求、重定向、响应以及报告请求和响应的日志。来自 ADS 和源服务器的错误也会发送到亚马逊 CloudWatch的日志组。 MediaTailor 还提供了有关跳过的广告及其被跳过的原因的信息。有关更多信息，请参阅 跳过广告疑难解答。您还可以在高持久性存储中检索您的日志数据。有关一般信息，请参阅《亚马逊 CloudWatch 日志用户指南》。有关 ADS 日志以及如何通过 Amazon Logs Insights 访问这些 CloudWatch 日志进行分析的信息，请参阅AWS Elemental MediaTailor 亚马逊日志见解中的 ADS CloudWatch 日志分析。

主题

- 查看 AWS Elemental MediaTailor logs
- 监控 AWS Elemental MediaTailor 使用亚马逊 CloudWatch 指标
- 记录 AWS Elemental MediaTailor API 调用
- 正在接收 AWS Elemental MediaTailor 频道集合警报
- 标签 AWS Elemental MediaTailor 资源
- 监控 AWS 带工作流监视器的媒体服务

查看 AWS Elemental MediaTailor logs

MediaTailor 发出日志，描述频道和回放配置中的各种里程碑和活动。您可以使用这些日志来了解您的工作流程并对服务问题进行故障排除。以下主题介绍日志和日志记录选项。

主题

- AWS Elemental MediaTailor ADS 日志描述和事件类型
- AWS Elemental MediaTailor 清单日志描述和事件类型
- AWS Elemental MediaTailor 转码日志描述和事件类型

- 使用已售日志发送 AWS Elemental MediaTailor logs
- 编写 AWS Elemental MediaTailor 直接登录到亚马逊 CloudWatch 日志
- 控制音量 AWS Elemental MediaTailor logs
- 筛选 AWS Elemental MediaTailor 日志和事件
- 正在生成 AWS Elemental MediaTailor 调试日志

AWS Elemental MediaTailor ADS 日志描述和事件类型

以下部分描述了为描述广告决策服务器 (ADS) 的事件而 MediaTailor 发出的日志。这些是`AdDecisionServerInteractions`日志。

主题

- AdDecisionServerInteractions events
- ADS 日志描述
- ADS 日志 JSON 架构

AdDecisionServerInteractions events

以下事件是在与广告决策服务器 (ADS) MediaTailor 交互期间发出的。

Log	说明
`AD_MARKER_FOUND`	MediaTailor 在清单中找到了一个广告标记。
`BEACON_FIRED`	启动了追踪信标来报告广告事件。在服务器端报告模式（默认）下， MediaTailor 触发信标。在客户端报告模式下，播放设备会触发信标。
`EMPTY_VAST_RESPONSE`	ADS 返回了一个空的 VAST 响应，其中包含零个广告。
`EMPTY_VMAP_RESPONSE`	ADS 返回了一个空的 VMAP 响应。
`ERROR_ADS_INVALID_RESPONSE`	ADS 返回的状态码不是 200。
`ERROR_ADS_IO`	MediaTailor 尝试与 ADS 通信时遇到错误。

Log	说明
ERROR_ADS_RESPONSE_PARSE	MediaTailor 解析 ADS 响应时遇到错误。
ERROR_ADS_RESPONSE_UNKNOWN_ROOT_ELEMENT	ADS 响应包含无效的根元素。
ERROR_ADS_TIMEOUT	向 ADS 发出的 MediaTailor 请求已超时。
ERROR_DISALLOWED_HOST	不允许使用 ADS 主机。
ERROR_FIRING_BEACON_FAILED	MediaTailor 未能发射追踪信标。
ERROR_PERSONALIZATION_DISABLED	此会话禁用广告插入。
ERROR_UNKNOWN	MediaTailor 在 ADS 请求期间遇到未知错误。
ERROR_UNKNOWN_HOST	ADS 主机未知。
ERROR_VAST_INVALID_MEDIA_FILE	VAST Ad 的MediaFile 元素无效或缺失。
ERROR_VAST_INVALID_VAST_AD_TAG_URI	VAST 响应包含无效的VASTAdTagURI 。
ERROR_VAST_MISSING_CREATIVES	VAST Ad 包含零个或多个Creatives 元素。只需要一个Creatives 元素。
ERROR_VAST_MISSING_IMPRESSION	VAST Ad 包含零个Impression 元素。至少需要一个Impression 元素。
ERROR_VAST_MISSING_MEDIAFILES	VAST Ad 包含零个或多个MediaFiles 元素。只需要一个MediaFiles 元素。
ERROR_VAST_MISSING_OVERLAYS	MediaTailor 没有从广告服务器收到任何非线性素材。
ERROR_VAST_MULTIPLE_LINEAR	VAST Ad 包含多个Linear元素。
ERROR_VAST_MULTIPLE_TRACKING_EVENTS	VAST Ad 包含多个TrackingEvents 元素。

Log	说明
ERROR_VAST_REDIRECT_EMPTY_RESPONSE	VAST 重定向请求返回空响应。
ERROR_VAST_REDIRECT_FAILED	VAST 重定向请求遇到错误。
ERROR_VAST_REDIRECT_MULTIPLE_VAST	VAST 重定向请求返回了多个广告。
FILLED_AVAIL	MediaTailor 成功填补了空缺。
FILLED_OVERLAY_AVAIL	MediaTailor 成功填充了叠加层的可用性。
INTERSTITIAL_VOD_FAILURE	ADS 请求或响应在填充 VOD 播放列表的插页式广告空缺时遇到了问题。不会插入任何广告。
INTERSTITIAL_VOD_SUCCESS	MediaTailor 成功填充了 VOD 播放列表的插页空闲空间。
MAKING_ADS_REQUEST	MediaTailor 正在向广告索取广告。
MODIFIED_TARGET_URL	MediaTailor 修改了出站目标 URL。
NON_AD_MARKER_FOUND	MediaTailor 在清单中找到了不可操作的广告标记。
PRE_ADS_REQUEST_FUNCTION_COMPLETED	广告前请求挂钩中的一项单独功能已完成。这是一种选择加入的活动类型。
PRE_ADS_REQUEST_FUNCTION_ERROR	广告前请求挂钩中的单个函数失败。
PRE_ADS_REQUEST_HOOK_ERROR	广告前请求挂钩执行失败。
PRE_ADS_REQUEST_HOOK_SUMMARY	广告前请求挂钩执行摘要，包括成功或错误状态。这是一种选择加入的活动类型。
RAW_ADS_RESPONSE	MediaTailor 收到了原始的 ADS 响应。
REDIRECTED_VAST_RESPONSE	MediaTailor 在执行 VAST 重定向后收到了 VAST 响应。

Log	说明
VAST_REDIRECT	VAST 广告响应包含重定向。
VAST_RESPONSE	MediaTailor 收到了 VAST 的回复。
VOD_TIME_BASED_AVAIL_PLAN_SUCCESS	MediaTailor 成功为 VOD 模板创建了基于时间的可用计划。
VOD_TIME_BASED_AVAIL_PLAN_VAST_RESPONSE_FOR_OFFSET	MediaTailor 正在为VOD模板创建基于时间的可用计划。 MediaTailor 收到了有关时间偏移的 VAST 回复。
VOD_TIME_BASED_AVAIL_PLAN_WARNING_NO_ADVERTISEMENTS	在为 VOD 模板创建基于时间的可用计划时，ADS 请求或响应遇到了问题。不会插入任何广告。
WARNING_NO_ADVERTISEMENTS	MediaTailor 在填补空缺时遇到了问题。未插入任何广告。
WARNING_URL_VARIABLE_SUBSTITUTION_FAILED	MediaTailor 无法替换 ADS 网址中的动态变量。检查 URL 配置。
WARNING_VPAID_AD_DROPPED	VPAID 广告因缺少清单而下架，或者会话使用服务器端报告。

ADS 日志描述

此部分描述了 ADS 日志描述的结构和内容。要在 JSON 编辑器中自行探究，请使用 the section called “ADS 日志 JSON 架构” 上的列表。

ADS 日志中的每个事件都包含由 CloudWatch 日志生成的标准字段。有关信息，请参阅使用日志见解分析 CloudWatch 日志数据。

ADS 日志属性

此部分描述了 ADS 日志的属性。

属性	Type	必需	描述
adsRequestUrl	字符串	false	由发出的广告请求的完整网址 MediaTailor。
avail	效用 类型的对象	false	有关 MediaTailor 充斥广告的工具的信息。当前，对于FILLED_AVAIL 事件类型，这是首次遇到可用性 MediaTailor 时创建的计划。最终填充效用的方式可能与此计划有所不同，具体取决于内容的播放方式。
awsAccountId	字符串	true	用于会话的 MediaTailor 配置的 AWS 账户 ID。
customerId	字符串	true	AWS 账户 ID 的哈希版本，可用于关联多个日志条目。
eventDescription	字符串	true	触发此日志消息的事件的简短描述，由 MediaTailor 服务提供。默认情况下，这为空。示例：Got VAST response。
eventTimestamp	字符串	true	事件的日期和时间。
eventType	字符串	true	已触发此日志消息的事件的代码。示例：VAST_RESPONSE 。

属性	Type	必需	描述
errorType	字符串	false	结构化错误分类。演讲PRE_ADS_REQUEST_HOOK_ERROR 和PRE_ADS_REQUEST_FUNCTION_ERROR 活动。价值观：INTERNAL_ERROR SYNTAX_ERROR 、、RESOURCE_LIMIT_ERROR 、RESTRICTION_ERROR 、TIMEOUT_ERROR 、VALIDATION_ERROR 。
cause	字符串	false	描述特定故障的错误详细信息字符串。演讲PRE_ADS_REQUEST_HOOK_ERROR 和PRE_ADS_REQUEST_FUNCTION_ERROR 活动。
eventId	字符串	false	每次挂钩调用的唯一标识符。使用此字段关联同一执行的挂钩级和函数级事件。仅在 Functions 事件上显示。
executionTimeMs	integer	false	以毫秒为单位的执行时间。仅在 Functions 事件上显示。

属性	Type	必需	描述
functionId	字符串	false	函数的标识符。挂机事件时，这是挂机上配置的函数。在函数事件上，这是执行器中的单个函数。仅在 Functions 事件上显示。
functionType	字符串	false	函数的类型：CUSTOM_OUTPUT 、HTTP_REQUEST SEQUENTIAL_EXECUTOR 、或CONCURRENT_EXECUTOR 。仅在函数级事件上出现。
http	object	false	HTTP 请求和响应的详细信息。仅在函数级事件上显示HTTP_REQUEST 函数。包含request（网址、方法、标头、正文）responseTimeMs 、、response（状态码、标头、正文）和truncated （当请求或响应正文被截断以适应日志事件大小限制时为真）。

属性	Type	必需	描述
input	object	false	Before-values 输出表达式所针对的变量。仅在函数级事件上出现。
output	object	false	该函数生成的值。仅在PRE_ADS_REQUEST_FUNCTION_COMPLETED 活动中出席。
status	字符串	false	钩子执行结果:SUCCESS或ERROR.仅在PRE_ADS_REQUEST_HOOK_SUMMARY 活动中出席。
originId	字符串	true	配置中的 MediaTailor 配置名称。这与视频内容源不同，它也是配置的一部分。
prefetchScheduleName	字符串	false	与此广告事件相关的预取时间表的名称。
requestHeaders	requestheaders 类型的数组	false	ADS 请求 MediaTailor 中包含的标头。通常，当对 ADS 的请求失败时，日志会包含这些内容以帮助进行问题排查。

属性	Type	必需	描述
requestId	字符串	true	MediaTailor 请求 ID，可用于关联同一请求的多个日志条目。
sessionId	字符串	true	MediaTailor 分配给玩家会话的唯一数字标识符。玩家为会话提出的所有请求都具有相同的会话 ID。示例：e039fd39-09f0-46b2-aca9-9871cc116cde 。
sessionType	字符串（合法值：[DASH, HLS]）	true	玩家流类型。
vastAd	vastAd 类型的对象	false	有关从 VAST 响应中解析的单个广告的信息。
vastResponse	vastResponse 类型的对象	false	有关从 ADS MediaTailor 收到的 VAST 回复的信息。
vodCreativeOffsets	vod CreativeOffsets 类型的对象	false	一个基于 VMAP 响应的地图，它指示清单中 MediaTailor 将在何处插入空闲时间偏移量。
vodVastResponseTimeOffset	数字	false	VOD 广告插入的 VMAP 特定时间偏差。

adContent

此部分描述了 ADS 日志 adContent 的属性。

ADS 日志 adContent 属性

属性	Type	必需	说明
adPlaylistUris	广告 PlaylistUris 类型的对象	false	从变体的原始清单到变体的广告清单的映射。对于 DASH，这包含一个条目，因为所有变体都在一个 DASH 清单中表示。

广告 PlaylistUris

本节介绍了 ADS 日志广告的属性PlaylistUris。

ADS 日志和PlaylistUris 属性

属性	Type	必需	描述
<any string>	字符串	false	特定变体的广告清单的 URL。

效用

此部分描述了 ADS 日志效用的属性。

ADS 日志效用属性

属性	Type	必需	描述
availId	字符串	true	此效用的唯一标识符。对于 HLS，这是效用在其中开始的媒体序列号。对于 DASH，这是周期 ID。

属性	Type	必需	描述
creativeAds	creativeAd 类型的数组	true	无济于事 MediaTailor 的广告。
fillRate	数字	true	广告填充效用持续时间（从 0.0（对于 0%）到 1.0（对于 100%））的速率。
filledDuration	数字	true	所有插入到效用中的广告的持续时间的总和。
numAds	数字	true	MediaTailor 插入 Avail 的广告数量。
originAvailDuration	数字	true	来源（CUE_OUT 或 SCTE）中的内容流指定的效用的持续时间。
skippedAds	skippedAd 类型的数组	false	由于TRANSCODE_IN_PROGRESS 和之类的原因，MediaTailor 未插入的广告TRANSCODE_ERROR 。 有关跳过广告的原因列表，请参阅跳过广告疑难解答。
slateAd	slateAd 类型的对象	true	有关 Slate 广告的信息，该广告 MediaTailor 用于填充任何未填充的广告片段。

creativeAd

此部分描述了 ADS 日志 creativeAd 的属性。

ADS 日志 creativeAd 属性

属性	Type	必需	说明
adContent	adContent 类型的对象	true	有关已插入广告的内容的信息。
creativeUniqueId	字符串	true	广告的唯一标识符，用作转码的密钥。这是 VAST 响应中广告素材的 ID 字段（如果可用）。否则，它是广告的 Mezzanine URL。
trackingEvents	trackingEvents 类型的对象	true	广告的各种跟踪事件的跟踪信标 URL。密钥为事件名称，值为信标 URL 的列表。
transcodedAdDuration	数字	true	从转码资产计算的广告持续时间。
uri	字符串	true	Mezzanine 广告的 URL，这是转码器的输入。
vastDuration	数字	true	广告的持续时间（从 VAST 响应解析）。

requestheaders

此部分描述了 ADS 日志 requestheaders 的属性。

ADS 日志 requestheaders 属性

属性	Type	必需	描述
name	字符串	true	标头的名称。
value	字符串	true	标头的值。

skippedAd

此部分描述了 ADS 日志 skippedAd 的属性。

ADS 日志 skippedAd 属性

属性	Type	必需	描述
adMezzanineUrl	字符串	true	跳过的广告的 Mezzanine URL。
creativeUniqueId	字符串	true	广告的唯一标识符，用作转码的密钥。这是 VAST 响应中广告素材的 ID 字段（如果可用）。否则，它是广告的 Mezzanine URL。
skippedReason	字符串	true	用于指示未插入广告的原因的代码。示例：TRANSCODE_IN_PROGRESS 。 有关跳过广告的原因列表，请参阅跳过广告疑难解答。
transcodedAdDuration	数字	false	从转码资产计算的广告持续时间。

属性	Type	必需	描述
vastDuration	数字	true	广告的持续时间（从 VAST 响应解析）。

slateAd

此部分描述了 ADS 日志 slateAd 的属性。

ADS 日志 slateAd 属性

属性	Type	必需	说明
adContent	adContent 类型的对象	true	有关已插入广告的内容的信息。
creativeUniqueId	字符串	true	广告的唯一标识符，用作转码的密钥。这是 VAST 响应中广告素材的 ID 字段（如果可用）。否则，它是广告的 Mezzanine URL。
transcodedAdDuration	数字	true	从转码资产计算的广告持续时间。
uri	字符串	true	Mezzanine 广告的 URL，这是转码器的输入。

trackingEvents

此部分描述了 ADS 日志 trackingEvents 的属性。

ADS 日志 trackingEvents 属性

属性	Type	必需	说明
<any string>	字符串类型的数组	false	指定跟踪事件（展示、完成等）的信标 URL 的列表

vastAd

此部分描述了 ADS 日志 vastAd 的属性。

ADS 日志 vastAd 属性

属性	Type	必需	描述
adSystem	字符串	true	VAST 响应中 AdSystem 标签的值。
adTitle	字符串	true	VAST 响应中对广告可用的媒体文件。
creativeAdId	字符串	true	VAST 响应中 Creative 标签的 adId 属性的值。
creativeId	字符串	true	VAST 响应中 Creative 标签的 id 属性的值。
duration	数字	true	广告的大致持续时间，基于 VAST 响应中 linear 元素的 duration 标签。
trackingEvents	trackingEvents 类型的对象	true	广告的各种跟踪事件的跟踪信标 URL。密

属性	Type	必需	描述
			钥为事件名称，值为信标 URL 的列表。
vastAdId	字符串	true	VAST 响应中 Ad 标签的 id 属性的值。
vastAdTagUri	字符串	false	广告的 VMAP-specific 重定向 URI。
vastMediaFiles	巨大的 MediaFile 类型的数组	true	VAST 响应中对广告可用的媒体文件的列表。

巨大的 MediaFile

本节介绍了 ADS 大量日志的属性MediaFile。

ADS 记录大量MediaFile 属性

属性	Type	必需	描述
apiFramework	字符串	true	管理媒体文件所需的 API 框架。示例：VPAID。
bitrate	数字	true	媒体文件的比特率。
delivery	字符串	true	用于媒体文件的协议，设置为渐进式或流式传输。
height	数字	true	媒体文件的像素高度。
id	字符串	true	MediaFile 标签的 id 属性的值。

属性	Type	必需	描述
type	字符串	true	媒体文件的 MIME 类型，取自 MediaFile 标签的 type 属性。
uri	字符串	true	Mezzanine 广告的 URL，这是转码器的输入。
width	数字	true	媒体文件的像素宽度。

vastResponse

此部分描述了 ADS 日志 vastResponse 的属性。

ADS 日志 vastResponse 属性

属性	Type	必需	说明
errors	字符串类型的数组	true	从 VAST 响应中的 Error 标签解析的错误 URL。
vastAds	vastAd 类型的数组	true	从 VAST 响应解析的广告。
version	字符串	true	VAST 规范版本（从响应中 VAST 标签的 version 属性解析）。

vod CreativeOffsets

本节介绍了 ADS 日志 vod CreativeOffsets 的属性。

ADS 日志 vod 属性 CreativeOffsets

属性	Type	必需	说明
<any string>	vod CreativeOffset 类型的数组	false	从清单中的时间偏差到此时要插入的广告列表的映射。

vod CreativeOffset

本节介绍了 ADS 日志 vod CreativeOffset 的属性。

ADS 日志 vod 属性 CreativeOffset

属性	Type	必需	说明
adContent	adContent 类型的对象	true	有关已插入广告的内容的信息。
creativeUniqueId	字符串	true	广告的唯一标识符，用作转码的密钥。这是 VAST 响应中广告素材的 ID 字段（如果可用）。否则，它是广告的 Mezzanine URL。
trackingEvents	trackingEvents 类型的对象	true	广告的各种跟踪事件的跟踪信标 URL。密钥为事件名称，值为信标 URL 的列表。
transcodedAdDuration	数字	true	从转码资产计算的广告持续时间。
uri	字符串	true	Mezzanine 广告的 URL，这是转码器的输入。

属性	Type	必需	说明
vastDuration	数字	true	广告的持续时间（从 VAST 响应解析）。

ADS 日志 JSON 架构

以下列出了 AWS Elemental MediaTailor ADS 日志的 JSON 架构。

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "http://amazon.com/elemental/midas/mms/adsLogSchema.json",
  "type": "object",
  "title": "AWS Elemental MediaTailor ADS Log JSON Schema",
  "required": [
    "eventType",
    "eventTimestamp",
    "requestId",
    "sessionType",
    "eventDescription",
    "awsAccountId",
    "customerId",
    "originId",
    "sessionId"
  ],
  "additionalProperties": false,
  "properties": {
    "eventType": {
      "$id": "#/properties/eventType",
      "type": "string",
      "description": "The code for the event that triggered this log message. Example:
 <code>VAST_RESPONSE</code>.",
      "examples": [
        "FILLED_AVAIL"
      ]
    },
    "eventTimestamp": {
      "$id": "#/properties/eventTimestamp",
      "type": "string",
      "description": "The date and time of the event.",
      "examples": [
        "1970-01-01T00:00:00Z"
```

```
      ],
      "format": "date-time"
    },
    "requestId": {
      "$id": "#/properties/requestId",
      "type": "string",
      "description": "The MediaTailor request ID, which you can use to correlate
multiple log entries for the same request.",
      "examples": [
        "c7c7ae8c-a61e-44e0-8efd-7723995337a1"
      ],
      "pattern": "^(.*)$"
    },
    "sessionType": {
      "$id": "#/properties/sessionType",
      "type": "string",
      "enum": [
        "HLS",
        "DASH"
      ],
      "description": "The player's stream type."
    },
    "eventDescription": {
      "$id": "#/properties/eventDescription",
      "type": "string",
      "description": "A short description of the event that triggered this log message,
provided by the MediaTailor service. By default, this is empty. Example: <code>Got
VAST response</code>.",
      "default": "",
      "examples": [
        "Got VAST response"
      ],
      "pattern": "^(.*)$"
    },
    "awsAccountId": {
      "$id": "#/properties/awsAccountId",
      "type": "string",
      "description": "The AWS account ID for the MediaTailor configuration that was
used for the session."
    },
    "customerId": {
      "$id": "#/properties/customerId",
      "type": "string",
```

```
      "description": "The hashed version of the AWS account ID, which you can use to
correlate multiple log entries.",
      "pattern": "^(.*)$"
    },
    "originId": {
      "$id": "#/properties/originId",
      "type": "string",
      "description": "The configuration name from the MediaTailor configuration. This
is different from the video content source, which is also part of the configuration.",
      "examples": [
        "external-canary-dash-serverside-reporting-onebox"
      ],
      "pattern": "^(.*)$"
    },
    "sessionId": {
      "$id": "#/properties/sessionId",
      "type": "string",
      "description": "The unique numeric identifier that MediaTailor assigned to the
player session. All requests that a player makes for a session have the same session
ID. Example: <code>e039fd39-09f0-46b2-aca9-9871cc116cde</code>.",
      "examples": [
        "120b9873-c007-40c8-b3db-0f1bd194970b"
      ],
      "pattern": "^(.*)$"
    },
    "avail": {
      "$id": "#/properties/avail",
      "type": "object",
      "title": "avail",
      "description": "Information about an avail that MediaTailor fills with ads.
Currently, for the <code>FILLED_AVAIL</code> event type, this is the plan created by
MediaTailor when it first encounters the avail. How the avail is eventually filled may
vary from this plan, depending on how the content plays out.  ",
      "required": [
        "creativeAds",
        "originAvailDuration",
        "filledDuration",
        "fillRate",
        "driftMillisecondsAtAvailStart",
        "numAds",
        "slateAd",
        "availId"
      ],
      "additionalProperties":  false,
```

```
    "properties": {
      "originAvailDuration": {
        "$id": "#/properties/avail/originAvailDuration",
        "type": "number",
        "description": "The duration of the avail as specified in the content stream
from the origin (<code>CUE_OUT</code> or <code>SCTE</code>)."
      },
      "filledDuration": {
        "$id": "#/properties/avail/filledDuration",
        "type": "number",
        "description": "The sum of the durations of all the ads inserted into the
avail."
      },
      "fillRate": {
        "$id": "#/properties/avail/fillRate",
        "type": "number",
        "description": "The rate at which the ads fill the avail duration, from 0.0
(for 0%) to 1.0 (for 100%)."
      },
      "driftMillisecondsAtAvailStart": {
        "$id": "#/properties/avail/driftMillisecondsAtAvailStart",
        "type": "number",
        "description": "The cumulative drift at the beginning of this avail. A
positive value implies that we are moving away from the live edge, a negative value
implies that we are moving towards the live edge."
      },
      "creativeAds": {
        "$id": "#/properties/avail/creativeAds",
        "type": "array",
        "description": "The ads that MediaTailor inserted into the avail.",
        "items": {
          "type": "object",
          "title": "creativeAd",
          "description": "Information about a single inserted ad.",
          "required": [
            "uri",
            "creativeUniqueId",
            "adSystem",
            "adContent",
            "trackingEvents",
            "vastDuration",
            "transcodedAdDuration"
          ],
          "additionalProperties": false,
```

```
            "properties": {
              "uri": { "$ref": "#/definitions/adMezzanineUri" },
              "creativeUniqueId": { "$ref": "#/definitions/creativeUniqueId" },
              "adSystem": { "$ref": "#/definitions/adSystem" },
              "adContent": { "$ref": "#/definitions/adContent" },
              "trackingEvents": { "$ref": "#/definitions/trackingEvents" },
              "vastDuration": { "$ref": "#/definitions/vastDuration" },
              "transcodedAdDuration": { "$ref": "#/definitions/transcodedAdDuration" }
            }
          }
        },
        "numAds": {
          "$id": "#/properties/avail/numAds",
          "type": "number",
          "description": "The number of ads that MediaTailor inserted into the avail."
        },
        "slateAd": {
          "$id": "#/properties/avail/slateAd",
          "type": ["object", "null"],
          "title": "slateAd",
          "description": "Information about the slate ad, which MediaTailor uses to
fill any unfilled segments in the avail.",
          "additionalProperties": false,
          "required": [
            "uri",
            "creativeUniqueId",
            "adContent",
            "transcodedAdDuration"
          ],
          "properties": {
            "uri": { "$ref": "#/definitions/adMezzanineUri" },
            "creativeUniqueId": { "$ref": "#/definitions/creativeUniqueId" },
            "adContent": { "$ref": "#/definitions/adContent" },
            "transcodedAdDuration": { "$ref": "#/definitions/transcodedAdDuration" }
          }
        },
        "availId": {
          "$id": "#/properties/avail/availId",
          "type": "string",
          "description": "The unique identifier for this avail. For HLS, this is the
media sequence number where the avail begins. For DASH, this is the period ID."
        },
        "skippedAds": {
          "$id": "#/properties/avail/skippedAds",
```

```
          "type": "array",
          "description": "The ads that MediaTailor didn't insert, for reasons like
<code>TRANSCODE_IN_PROGRESS</code> and <code>TRANSCODE_ERROR</code>.",
          "items": {
            "type": "object",
            "title": "skippedAd",
            "description": "Information about a single skipped ad.",
            "required": [
              "creativeUniqueId",
              "adMezzanineUrl",
              "skippedReason",
              "vastDuration"
            ],
            "additionalProperties": false,
            "properties": {
              "creativeUniqueId": { "$ref": "#/definitions/creativeUniqueId" },
              "adMezzanineUrl": {
                "type": "string",
                "description": "The mezzanine URL of the skipped ad."
              },
              "skippedReason": {
                "type": "string",
                "description": "The code that indicates why the ad wasn't inserted.
Example: <code>TRANSCODE_IN_PROGRESS</code>."
              },
              "vastDuration": { "$ref": "#/definitions/vastDuration" },
              "transcodedAdDuration": { "$ref": "#/definitions/transcodedAdDuration" },
              "targetVariant": {
                "type": "object",
                "title": "targetVariant",
                "description": "The target variant of the source content. This key is
present when an ad wasn't inserted due to the source content containing a variant that
could not match to any variants present in this ad.",
                "required": [
                  "mediaProtocol",
                  "mediaType",
                  "bitrate",
                  "mediaResolution",
                  "codecs"
                ],
                "additionalProperties": false,
                "properties": {
                  "mediaProtocol": {
                    "type": "string",
```

```
      "description": "The media protocol of this variant, such as HLS.",
      "enum": [
        "HLS",
        "DASH"
      ]
    },
    "mediaType": {
      "type": "array",
      "description": "The media type of this variant, such as VIDEO.",
      "items": {
        "type": "string",
        "enum": [
          "VIDEO",
          "AUDIO",
          "SUBTITLES",
          "TRICK_PLAY"
        ],
        "description": "Media type, such as VIDEO."
      }
    },
    "bitrate": {
      "$ref": "#/definitions/bitrate"
    },
    "mediaResolution": {
      "type": "object",
      "title": "mediaResolution",
      "description": "The media resolution of this variant.",
      "required": [
        "width",
        "height"
      ],
      "additionalProperties": false,
      "properties": {
        "width": {
          "$ref": "#/definitions/width"
        },
        "height": {
          "$ref": "#/definitions/height"
        }
      }
    },
    "codecs": {
      "type": "array",
      "description": "The codecs of this variant.",
```

```
                    "items": {
                      "type": "string",
                      "description": "Codec, such as avc1."
                    }
                  }
                }
              }
            }
          }
        },
        "adBreakTrackingEvents": {
          "$id": "#properties/avail/adBreakTrackingEvents",
          "type": "object",
          "title": "adBreakTrackingEvents",
          "description": "VMAP/ad break tracking events and corresponding URL",
          "properties": {
            "items": {
              "$ref": "#/definitions/adBreakTrackingEvents"
            }
          }
        }
      }
    },

    "vastResponse": {
      "$id": "#/properties/vastResponse",
      "type": "object",
      "title": "vastResponse",
      "description": "Information about the VAST response that MediaTailor received
from the ADS.",
      "required": [
        "version",
        "vastAds",
        "errors",
        "nonLinearAdsList"
      ],
      "additionalProperties":  false,
      "properties": {
        "version": {
          "$id": "#/properties/vastResponse/version",
          "type": "string",
          "description": "The VAST specification version, parsed from the
<code>version</code> attribute of the <code>VAST</code> tag in the response.",
          "examples": [
```

```
            "3.0"
          ],
          "pattern": "^(.*)$"
        },
        "vastAds": {
          "$id": "#/properties/vastResponse/vastAds",
          "type": "array",
          "description": "The ads parsed from the VAST response.",
          "items": {
            "$ref": "#/definitions/vastAd"
          }
        },
        "errors": {
          "$id": "#/properties/vastResponse/errors",
          "type": "array",
          "description": "The error URLs parsed from the <code>Error</code> tags in the
VAST response.",
          "items": {
            "type": "string",
            "description": "A single error URL."
          }
        },
        "nonLinearAdsList": {
          "$id": "#/properties/vastResponse/nonLinearAds",
          "type": "array",
          "description": "A list of NonLinearAds as they are read from the VAST
response.",
          "items": {
            "$ref": "#/definitions/nonLinearAds"
          }
        }
      }
    },

    "vastAd": {
      "$ref": "#/definitions/vastAd"
    },

    "vodVastResponseTimeOffset": {
      "$id": "#/properties/vodVastResponseTimeOffset",
      "type": "number",
      "description": "The VMAP specific time offset for VOD ad insertion.",
      "examples": [
        5.0
```

```
      ]
    },

    "vodCreativeOffsets": {
      "$id": "#/properties/vodCreativeOffsets",
      "type": "object",
      "title": "vodCreativeOffsets",
      "description": "A map that indicates the time offsets in the manifest where
MediaTailor will insert avails, based on the VMAP response.",
      "additionalProperties": {
        "type": "array",
        "$id": "#/properties/vodCreativeOffsets/entry",
        "description": "A mapping from a time offset in the manifest to a list of ads
to insert at this time.",
        "items": {
          "type": "object",
          "$id": "#/properties/vodCreativeOffsets/entry/items",
          "title": "vodCreativeOffset",
          "description": "The list of ads to insert at the specified time offset.",
          "additionalProperties": false,
          "required": [
            "uri",
            "creativeUniqueId",
            "vastDuration",
            "transcodedAdDuration",
            "adContent",
            "trackingEvents"
          ],
          "properties": {
            "uri": { "$ref": "#/definitions/adMezzanineUri" },
            "creativeUniqueId": { "$ref": "#/definitions/creativeUniqueId" },
            "vastDuration": { "$ref": "#/definitions/vastDuration" },
            "transcodedAdDuration": { "$ref": "#/definitions/transcodedAdDuration" },
            "adContent": { "$ref": "#/definitions/adContent" },
            "trackingEvents": { "$ref": "#/definitions/trackingEvents" }
          }
        }
      }
    },

    "adsRequestUrl": {
      "$id": "#/properties/adsRequestUrl",
      "type": "string",
      "description": "The full URL of the ADS request made by MediaTailor."
```

```
    },
    "adMarkers": {
      "$id": "#/properties/adMarkers",
      "type": "string",
      "description": "Found Ad Marker in the Manifest."
    },
    "segmentationUpid": {
      "$id": "#/properties/segmentationUpid",
      "type": "string",
      "description": "Value of segmentation upid parsed from ad markers in manifest."
    },
    "segmentationTypeId": {
      "$id": "#/properties/segmentationTypeId",
      "type": "integer",
      "description": "Value of segmentation typeId parsed from ad markers in manifest."
    },
    "requestHeaders": {
      "$id": "#/properties/requestHeaders",
      "type": "array",
      "description": "The headers that MediaTailor included with the ADS request.
Typically, the logs include these when a request to the ADS fails, to help with
troubleshooting.",
      "items": {
        "type": "object",
        "title": "requestheaders",
        "description": "The name and value for a single header included in the ADS
request.",
        "required": [
          "name",
          "value"
        ],
        "additionalProperties": false,
        "properties": {
          "name": {
            "type": "string",
            "description": "The name of the header."
          },
          "value": {
            "type": "string",
            "description": "The value of the header."
          }
        }
      }
    },
```

```
   "originalTargetUrl": {
     "$id": "#/properties/originalTargetUrl",
     "type": "string",
     "description": "The old URL to which MediaTailor was going to make a request."
   },
   "prefetchScheduleName": {
     "$id": "#/properties/prefetchScheduleName",
     "type": "string",
     "description": "The name of the prefetch schedule associated with this ad
event.",
     "examples": [
       "PrefetchScheduleName"
     ],
     "pattern": "^(.*)$"
   },
   "updatedTargetUrl": {
     "$id": "#/properties/updatedTargetUrl",
     "type": "string",
     "description": "The new URL to which MediaTailor is making a request."
   },

   "rawAdsResponse": {
     "$id": "#/properties/rawAdsResponse",
     "type": "string",
     "description": "Paginated ADS response as it's exactly returned to MediaTailor."
   },
   "rawAdsResponseIndex": {
     "$id": "#/properties/rawAdsResponseIndex",
     "type": "integer",
     "description": "Integer value denoting this rawAdsResponse's index into the
full ADS response. This value is used to order the paginated messages for this ADS
response."
   },

   "eventId": {
     "$id": "#/properties/eventId",
     "type": "string",
     "description": "Unique identifier per hook invocation. Use this field to
correlate hook-level and function-level events for the same execution.",
     "examples": [
       "5dc6f040-0f72-4e8c-a64e-25eeef62708c"
     ]
   },
```

```
    "functionId": {
      "$id": "#/properties/functionId",
      "type": "string",
      "description": "The identifier of the function. On hook events, this is the
function configured on the hook. On function events, this is the individual function
within the executor.",
      "examples": [
        "my_first_function"
      ]
    },
    "functionType": {
      "$id": "#/properties/functionType",
      "type": "string",
      "enum": [
        "CUSTOM_OUTPUT",
        "HTTP_REQUEST",
        "SEQUENTIAL_EXECUTOR",
        "CONCURRENT_EXECUTOR"
      ],
      "description": "The type of function."
    },
    "executionTimeMs": {
      "$id": "#/properties/executionTimeMs",
      "type": "integer",
      "description": "Execution time in milliseconds."
    },
    "status": {
      "$id": "#/properties/status",
      "type": "string",
      "enum": [
        "SUCCESS",
        "ERROR"
      ],
      "description": "Hook execution outcome. Present on HOOK_SUMMARY events only."
    },
    "errorType": {
      "$id": "#/properties/errorType",
      "type": "string",
      "enum": [
        "INTERNAL_ERROR",
        "SYNTAX_ERROR",
        "RESOURCE_LIMIT_ERROR",
        "RESTRICTION_ERROR",
        "TIMEOUT_ERROR",
```

```
        "VALIDATION_ERROR"
      ],
      "description": "Structured error classification. Present on error events only."
    },
    "cause": {
      "$id": "#/properties/cause",
      "type": "string",
      "description": "Error detail string describing the specific failure."
    },
    "input": {
      "$id": "#/properties/input",
      "type": "object",
      "description": "Before-values of variables targeted by output expressions."
    },
    "output": {
      "$id": "#/properties/output",
      "type": "object",
      "description": "Values produced by the function. Present on FUNCTION_COMPLETED
events only."
    },
    "http": {
      "$id": "#/properties/http",
      "type": "object",
      "description": "HTTP request and response details for HTTP_REQUEST functions.",
      "properties": {
        "request": {
          "type": "object",
          "description": "The HTTP request details.",
          "properties": {
            "url": { "type": "string", "description": "The request URL." },
            "method": { "type": "string", "description": "The HTTP method." },
            "headers": {
              "type": "array",
              "description": "The request headers.",
              "items": {
                "type": "object",
                "properties": {
                  "name": { "type": "string" },
                  "value": { "type": "string" }
                }
              }
            },
            "body": { "type": ["string", "null"], "description": "The request body." }
          }
```

```
       },
       "responseTimeMs": { "type": ["integer", "null"], "description": "HTTP response
time in milliseconds." },
       "response": {
         "type": ["object", "null"],
         "description": "The HTTP response details. Null when the server was never
reached.",
         "properties": {
           "statusCode": { "type": ["integer", "null"], "description": "The HTTP
status code. Null when the server was never reached." },
           "headers": {
             "type": ["array", "null"],
             "description": "The response headers.",
             "items": {
               "type": "object",
               "properties": {
                 "name": { "type": "string" },
                 "value": { "type": "string" }
               }
             }
           },
           "body": { "type": ["string", "null"], "description": "The response body." }
         }
       },
       "truncated": { "type": "boolean", "description": "True when request or response
body was truncated to fit within log event size limits." }
     }
   }
 },

 "__COMMENT_oneOf": "The oneOf section defines subtypes for our events. Subtypes can
have different rules, including which fields are required. For more information, see
https://json-schema.org/understanding-json-schema/reference/combining.html#oneof ",

 "oneOf": [
   { "$ref": "#/definitions/eventAdMarkersFound" },
   { "$ref": "#/definitions/eventNonAdMarkerFound" },
   { "$ref": "#/definitions/eventMakingAdsRequest" },
   { "$ref": "#/definitions/eventModifiedTargetUrl" },
   { "$ref": "#/definitions/eventRawAdsResponse" },
   { "$ref": "#/definitions/eventVastResponse" },
   { "$ref": "#/definitions/eventFilledAvail" },
   { "$ref": "#/definitions/eventFilledOverlayAvail" },
   { "$ref": "#/definitions/eventErrorFiringBeaconFailed" },
```

```
    { "$ref": "#/definitions/eventWarningNoAdvertisements" },
    { "$ref": "#/definitions/eventUnknownHost" },
    { "$ref": "#/definitions/eventErrorAdsTimeout" },
    { "$ref": "#/definitions/eventErrorVastMissingOverlays" },
    { "$ref": "#/definitions/eventPlannedAvail" },
    { "$ref": "#/definitions/eventEmptyVastResponse" },
    { "$ref": "#/definitions/eventEmptyVmapResponse" },
    { "$ref": "#/definitions/eventErrorUnknown" },
    { "$ref": "#/definitions/eventVastRedirect" },
    { "$ref": "#/definitions/eventRedirectedVastResponse" },
    { "$ref": "#/definitions/eventErrorAdsMissingImpression" },
    { "$ref": "#/definitions/eventErrorAdsResponseParse" },
    { "$ref": "#/definitions/eventErrorAdsInvalidResponse" },
    { "$ref": "#/definitions/eventErrorDisallowedHost"},
    { "$ref": "#/definitions/eventPersonalizationDisabled"},
    { "$ref": "#/definitions/eventWarningDynamicVariableSubFailed"},
    { "$ref": "#/definitions/eventVodTimeBasedAvailPlanVastResponseForOffset" },
    { "$ref": "#/definitions/eventVodTimeBasedAvailPlanSuccess" },
    { "$ref": "#/definitions/eventPreAdsRequestHookSummary" },
    { "$ref": "#/definitions/eventPreAdsRequestHookError" },
    { "$ref": "#/definitions/eventPreAdsRequestFunctionCompleted" },
    { "$ref": "#/definitions/eventPreAdsRequestFunctionError" }
  ],

  "definitions": {
    "eventAdMarkersFound": {
      "$id": "#/definitions/eventAdMarkersFound",
      "required": [
        "eventType",
        "adMarkers"
      ],
      "properties": {
        "eventType": {
          "type": "string",
          "const": "AD_MARKER_FOUND"
        }
      }
    },
    "eventNonAdMarkerFound": {
      "$id": "#/definitions/eventNonAdMarkerFound",
      "required": [
        "eventType",
        "adMarkers"
      ],
```

```
      "properties": {
        "eventType": {
          "type": "string",
          "const": "NON_AD_MARKER_FOUND"
        }
      }
    },
    "eventMakingAdsRequest": {
      "$id": "#/definitions/eventMakingAdsRequest",
      "required": [
        "eventType",
        "adsRequestUrl"
      ],
      "properties": {
        "eventType": {
          "type": "string",
          "const": "MAKING_ADS_REQUEST"
        }
      }
    },

    "eventModifiedTargetUrl": {
      "$id": "#/definitions/eventModifiedTargetUrl",
      "required": [
        "eventType",
        "originalTargetUrl",
        "updatedTargetUrl"
      ],
      "properties": {
        "eventType": {
          "type": "string",
          "const": "MODIFIED_TARGET_URL"
        }
      }
    },

    "eventRawAdsResponse": {
      "$id": "#/definitions/eventRawAdsResponse",
      "required": [
        "eventType",
        "rawAdsResponse",
        "rawAdsResponseIndex"
      ],
      "properties": {
```

```
        "eventType": {
          "type": "string",
          "const": "RAW_ADS_RESPONSE"
        }
      }
    },

    "eventVastResponse": {
      "$id": "#/definitions/eventVastResponse",
      "_comment": "NOTE: the vastResponse property should ideally be marked as a
required field for this event, but unfortunately, in the case of an empty vast
response, we currently emit an EMPTY_VAST_RESPONSE followed by a VAST_RESPONSE, and
the vastResponse property is not present in the latter. We need to fix this so that we
don't emit both of those events in the empty response case, and update this schema to
flag vastResponse as required for VAST_RESPONSE.",
      "required": [
        "eventType"
      ],
      "properties": {
        "eventType": {
          "type": "string",
          "const": "VAST_RESPONSE"
        }
      }
    },

    "eventFilledAvail": {
      "$id": "#/definitions/eventFilledAvail",
      "required": [
        "eventType",
        "avail"
      ],
      "properties": {
        "eventType": {
          "type": "string",
          "const": "FILLED_AVAIL"
        }
      }
    },

    "eventFilledOverlayAvail": {
      "$id": "#/definitions/eventFilledOverlayAvail",
      "required": [
        "eventType",
```

```
        "avail"
      ],
      "properties": {
        "eventType": {
          "type": "string",
          "const": "FILLED_OVERLAY_AVAIL"
        }
      }
    },

    "eventErrorVastMissingOverlays": {
      "$id": "#/definitions/eventErrorVastMissingOverlays",
      "required": [
        "eventType",
        "adsRequestUrl",
        "requestHeaders"
      ],
      "properties": {
        "eventType": {
          "type": "string",
          "const": "ERROR_VAST_MISSING_OVERLAYS"
        }
      }
    },

    "eventErrorFiringBeaconFailed": {
      "$id": "#/definitions/eventErrorFiringBeaconFailed",
      "required": [
        "eventType",
        "error",
        "beaconInfo"
      ],
      "properties": {
        "eventType": {
          "type": "string",
          "const": "ERROR_FIRING_BEACON_FAILED"
        }
      }
    },

    "eventWarningNoAdvertisements": {
      "$id": "#/definitions/eventWarningNoAdvertisements",
      "required": [
        "eventType"
```

```
      ],
      "_comment": "We should really have a more descriptive error field for these
 events",
      "properties": {
        "eventType": {
          "type": "string",
          "const": "WARNING_NO_ADVERTISEMENTS"
        }
      }
    },

    "eventUnknownHost": {
      "$id": "#/definitions/eventUnknownHost",
      "required": [
        "eventType",
        "requestHeaders"
      ],
      "properties": {
        "eventType": {
          "type": "string",
          "const": "ERROR_UNKNOWN_HOST"
        }
      }
    },

    "eventErrorAdsTimeout": {
      "$id": "#/definitions/eventErrorAdsTimeout",
      "required": [
        "eventType",
        "adsRequestUrl",
        "requestHeaders"
      ],
      "properties": {
        "eventType": {
          "type": "string",
          "const": "ERROR_ADS_TIMEOUT"
        }
      }
    },

    "eventPlannedAvail": {
      "$id": "#/definitions/eventPlannedAvail",
      "required": [
        "eventType"
```

```
      ],
      "_comment": "TODO: Flesh this out as we implement it",
      "properties": {
        "eventType": {
          "type": "string",
          "const": "PLANNED_AVAIL"
        }
      }
    },

    "eventEmptyVastResponse": {
      "$id": "#/definitions/eventEmptyVastResponse",
      "required": [
        "eventType"
      ],
      "properties": {
        "eventType": {
          "type": "string",
          "const": "EMPTY_VAST_RESPONSE"
        }
      }
    },

    "eventEmptyVmapResponse": {
      "$id": "#/definitions/eventEmptyVmapResponse",
      "required": [
        "eventType"
      ],
      "properties": {
        "eventType": {
          "type": "string",
          "const": "EMPTY_VMAP_RESPONSE"
        }
      }
    },

    "eventErrorUnknown": {
      "$id": "#/definitions/eventErrorUnknown",
      "required": [
        "eventType"
      ],
      "_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": "#/properites/adBreakTrackingEvents",
      "type": "object",
      "title": "adBreakTrackingEvents",
      "description": "These are all VMAP ad break tracking events.",
      "additionalProperties": {
        "type": "array",
        "description": "VMAP/ad break tracking events and corresponding URL",
        "items": {
          "type": "string",
          "description": "The beacon URLs for this tracking event."
        }
```

```
      }
    },
    "vastAd": {
      "$id": "#/properties/vastAd",
      "type": "object",
      "title": "vastAd",
      "description": "Information about a single ad parsed from the VAST response.",
      "required": [
        "vastAdId",
        "adSystem",
        "adTitle",
        "creativeId",
        "creativeAdId",
        "duration",
        "vastMediaFiles",
        "trackingEvents"
      ],
      "additionalProperties":  false,
      "properties": {
        "vastAdId": {
          "$id": "#/properties/vastAd/vastAdId",
          "type": "string",
          "description": "The value of the id attribute of the <code>Ad</code> tag in
the VAST response",
          "examples": [
            "ad1"
          ]
        },
        "adSystem": {"$ref": "#/definitions/adSystem" } ,
        "adTitle": {
          "$id": "#/properties/vastAd/adTitle",
          "type": "string",
          "description": "The media files that are available for the ad in the VAST
response.",
          "examples": [
            "External NCA1C1L1 LinearInlineSkippable"
          ]
        },
        "creativeId": {
          "$id": "#/properties/vastAd/creativeId",
          "type": "string",
          "description": "The value of the id attribute of the <code>Creative</code>
tag in the VAST response.",
          "examples": [
```

```
            "creative1"
          ]
        },
        "creativeAdId": {
          "$id": "#/properties/vastAd/creativeAdId",
          "type": "string",
          "description": "The value of the adId attribute of the <code>Creative</code>
tag in the VAST response."
        },
        "duration": {
          "$id": "#/properties/vastAd/duration",
          "type": "number",
          "description": "The approximate duration of the ad, based on the
<code>duration</code> tag in the <code>linear</code> element of the VAST response.",
          "examples": [
            30,
            30.0
          ]
        },
        "vastMediaFiles": {
          "$id": "#/properties/vastAd/vastMediaFiles",
          "type": "array",
          "description": "The list of available media files for the ad in the VAST
response.",
          "items": {
            "$id": "#/properties/vastAd/vastMediaFiles/items",
            "type": "object",
            "title": "vastMediaFile",
            "description": "Information about a media file for the ad.",
            "required": [
              "uri",
              "id",
              "delivery",
              "type",
              "apiFramework",
              "width",
              "height",
              "bitrate"
            ],
            "additionalProperties":  false,
            "properties": {
              "uri": { "$ref": "#/definitions/adMezzanineUri" },
              "id": {
                "$id": "#/properties/vastAd/vastMediaFiles/items/properties/id",
```

```
                "type": "string",
                "description": "The value of the id attribute of the <code>MediaFile</
code> tag.",
                "examples": [
                  "GDFP"
                ]
              },
              "delivery": {
                "$id": "#/properties/vastAd/vastMediaFiles/items/properties/delivery",
                "type": "string",
                "description": "The protocol used for the media file, set to either
 progressive or streaming.",
                "examples": [
                  "progressive"
                ]
              },
              "type": {
                "$id": "#/properties/vastAd/vastMediaFiles/items/properties/type",
                "type": "string",
                "description": "The MIME type of the media file, taken from the type
 attribute of the <code>MediaFile</code> tag.",
                "examples": [
                  "video/mp4"
                ]
              },
              "apiFramework": {
                "$id": "#/properties/vastAd/vastMediaFiles/items/properties/
apiFramework",
                "type": "string",
                "description": "The API framework needed to manage the media file.
 Example: <code>VPAID</code>."
              },
              "width": {
                "$ref": "#/definitions/width"
              },
              "height": {
                "$ref": "#/definitions/height"
              },
              "bitrate": {
                "$ref": "#/definitions/bitrate"
              }
            }
          }
        },
```

```
        "trackingEvents": { "$ref": "#/definitions/trackingEvents" },
        "vastAdTagUri": {
          "$id": "#/properties/vastAd/vastAdTagUri",
          "type": "string",
          "description": "The VMAP-specific redirect URI for an ad.",
          "examples": [
            "https://ads.redirect.com/redirect1"
          ]
        }
      }
    },

    "eventPreAdsRequestHookSummary": {
      "$id": "#/definitions/eventPreAdsRequestHookSummary",
      "required": [
        "eventType",
        "eventId",
        "functionId",
        "executionTimeMs",
        "status"
      ],
      "properties": {
        "eventType": {
          "type": "string",
          "const": "PRE_ADS_REQUEST_HOOK_SUMMARY"
        }
      }
    },

    "eventPreAdsRequestHookError": {
      "$id": "#/definitions/eventPreAdsRequestHookError",
      "required": [
        "eventType",
        "eventId",
        "functionId",
        "executionTimeMs",
        "errorType",
        "cause"
      ],
      "properties": {
        "eventType": {
          "type": "string",
          "const": "PRE_ADS_REQUEST_HOOK_ERROR"
        }
```

```
      }
    },

    "eventPreAdsRequestFunctionCompleted": {
      "$id": "#/definitions/eventPreAdsRequestFunctionCompleted",
      "required": [
        "eventType",
        "eventId",
        "functionId",
        "functionType",
        "executionTimeMs"
      ],
      "properties": {
        "eventType": {
          "type": "string",
          "const": "PRE_ADS_REQUEST_FUNCTION_COMPLETED"
        }
      }
    },

    "eventPreAdsRequestFunctionError": {
      "$id": "#/definitions/eventPreAdsRequestFunctionError",
      "required": [
        "eventType",
        "eventId",
        "functionId",
        "functionType",
        "executionTimeMs",
        "errorType",
        "cause"
      ],
      "properties": {
        "eventType": {
          "type": "string",
          "const": "PRE_ADS_REQUEST_FUNCTION_ERROR"
        }
      }
    }
  }
}
```

AWS Elemental MediaTailor 清单日志描述和事件类型

以下各节描述了在请求和接收清单时为描述源服务器发生的事件而发 MediaTailor 出的日志。这些是`ManifestService`日志。

主题

- ManifestService events
- 清单日志属性

ManifestService events

在与原点 MediaTailor 交互期间会发出以下事件。

Log	说明
`CONFIG_SECURITY_ERROR`	该 MediaTailor 配置存在安全问题。
`CONFIG_SYNTAX_ERROR`	来源和资产路径会导致 URL 格式不正确。
`CONNECTION_ERROR`	与源的 MediaTailor 连接被拒绝或失败。
`GENERATED_MANIFEST`	MediaTailor 生成了清单。 必须启用调试模式才能接收这些日志。有关调试日志模式（包括如何启用该模式）的信息，请参阅生成调试日志。
`HOST_DISALLOWED`	MediaTailor 不允许向该主机发出 HTTP 请求。
`INCOMPATIBLE_HLS_VERSION`	该清单使用不兼容的 HLS 版本。 MediaTailor 需要版本 3 或更高版本。
`INVALID_SINGLE_PERIOD_DASH_MANIFEST`	单周期 DASH 清单无效。 MediaTailor 正在通过单周期 DASH 清单。
`IO_ERROR`	MediaTailor 在与源通信时遇到 IO 错误。
`LAST_PERIOD_MISSING_AUDIO`	`AdaptationSets` 由于原始音频或视频不对齐，DASH 清单中的最后一个时段缺少所有音

Log	说明
	频。为避免播放问题，请至少将发布最后一个时段延迟到下一个请求。
LAST_PERIOD_MISSING_AUDIO_WARNING	AdaptationSets 由于原始音频或视频不对齐，DASH 清单中的最后一个时段缺少所有音频。选择发布（不延迟）最后一个时段。缺少音频可能会导致播放问题。
MANIFEST_ERROR	清 MediaTailor 单请求失败。
NO_MASTER_OR_MEDIA_PLAYLIST	原始响应不包含主播放列表或媒体播放列表。
NO_MASTER_PLAYLIST	原始响应不包含预期的主播放列表。
NO_MEDIA_PLAYLIST	原始响应不包含预期的媒体播放列表。
ORIGIN_MANIFEST	MediaTailor 获取了起源清单。 必须启用调试模式才能接收这些日志。有关调试日志模式（包括如何启用该模式）的信息，请参阅生成调试日志。
PARSING_ERROR	源无法解析清单请求。
PRE_SESSION_INIT_FUNCTION_COMPLETED	会话前初始化挂钩中的单个函数已完成。这是一种选择加入的活动类型。
PRE_SESSION_INIT_FUNCTION_ERROR	会话前初始化挂钩中的单个函数失败。
PRE_SESSION_INIT_HOOK_ERROR	会话前初始化挂钩执行失败。
PRE_SESSION_INIT_HOOK_SUMMARY	会话前初始化挂钩执行摘要，包括成功或错误状态。这是一种选择加入的活动类型。
SCTE35_PARSING_ERROR	MediaTailor 无法解析清单中的Signal Binary元素。

Log	说明
SESSION_INITIALIZED	会话已初始化。 必须启用调试模式才能接收这些日志。有关调试日志模式（包括如何启用该模式）的信息，请参阅生成调试日志。
TIMEOUT_ERROR	清 MediaTailor 单请求超时。
TRACKING_RESPONSE	MediaTailor 提供了追踪响应。 必须启用调试模式才能接收这些日志。有关调试日志模式（包括如何启用该模式）的信息，请参阅生成调试日志。
UNKNOWN_ERROR	MediaTailor 遇到未知错误。
UNKNOWN_HOST	主机未知。
UNSUPPORTED_SINGLE_PERIOD_DASH_MANIFEST	不支持单周期 DASH 清单。 MediaTailor 正在通过单周期 DASH 清单。

清单日志属性

本节介绍清单日志的属性。

属性	Type	必填
awsAccountId	字符串	true
eventTimestamp	字符串	true
originId	字符串	true
customerId	字符串	false
eventType	字符串	false

属性	Type	必填
sessionId	字符串	false
originRequestUrl	字符串	false
mediaTailorPath	字符串	false
requestId	字符串	false
responseBody	字符串	false
sessionType	字符串（合法值：[DASH, HLS]）	false
requestNextToken	字符串	false
eventDescription	字符串	false
errorType	字符串	false
eventId	字符串	false
executionTimeMs	integer	false
functionId	字符串	false
functionType	字符串	false
http	object	false
input	object	false
output	object	false
status	字符串	false
assetPath	字符串	false
originFullUrl	字符串	false
originPrefixUrl	字符串	false

属性	Type	必填
additionalInfo	字符串	false
cause	字符串	false
response	字符串	false
httpCode	字符串	false
errorMessage	字符串	false
adAdsResponse	字符串	false
adAdsRawResponse	字符串	false
adAdsRequest	字符串	false
adNumNewAvails	字符串	false
generatedMediaPlaylist	字符串	false
requestStartTime	字符串	false
requestEndTime	字符串	false
requestStartTimeEpochMillis	字符串	false
requestEndTimeEpochMillis	字符串	false

AWS Elemental MediaTailor 转码日志描述和事件类型

以下各节介绍在准备广告拼接广告素材时 MediaTailor 为描述转码服务的事件而发出的日志。这些是TranscodeService日志。

主题

- TranscodeService events

- 转码日志属性

TranscodeService events

在对广告进行转码时，会在 MediaTailor 互动期间触发以下事件。

Log	说明
IMPORT_ERROR	MediaTailor 在导入任务（预处理广告）期间遇到内部错误。使用一组空的广告。
INITIALIZED	MediaTailor 初始化了转码任务或导入任务（适用于预处理的广告）。
INTERNAL_ERROR	MediaTailor 遇到内部错误。使用一组空的广告。
MISSING_VARIANTS	MediaTailor 由于缺少变体，无法对广告进行转码。使用一组空的广告。
PROFILE_NOT_FOUND	MediaTailor 无法对广告进行转码，因为缺少要转码的个人资料。使用一组空的广告。
TRANSCODE_COMPLETED	视频转码已完成。该广告可用于广告插入。
TRANSCODE_ERROR	MediaTailor 在转码任务期间遇到内部错误。使用一组空的广告。
TRANSCODE_IN_PROGRESS	视频转码正在进行中。转码后的视频尚未准备就绪。使用一组空的广告。

转码日志属性

本节介绍转码日志的属性。

属性	Type	必需	描述
awsAccountId	字符串	true	用于会话的 MediaTailor 配置的 AWS 账户 ID。
eventTimestamp	字符串	true	事件的日期和时间。
originId	字符串	true	配置中的 MediaTailor 配置名称。这与视频内容源不同，它也是配置的一部分。
eventType	字符串	false	已触发此日志消息的事件的代码。示例：TRANSCODE_ERROR 。
eventDescription	字符串	false	触发此日志消息的事件的简短描述，由 MediaTailor 服务提供。默认情况下，这为空。
sessionId	字符串	false	MediaTailor 分配给玩家会话的唯一数字标识符。玩家为会话提出的所有请求都具有相同的会话 ID。示例：e039fd39-09f0-46b2-aca9-9871cc116cde 。
creativeUniqueId	字符串	false	正在转码的广告素材的唯一标识符。
profileName	字符串	false	

属性	Type	必需	描述
adUri	字符串	false	广告素材的 URI。
transcodeRequestId	字符串	false	此转码请求的唯一标识符。
cacheStatus	字符串	false	指明是否 MediaTailor 缓存了转码后的广告。

使用已售日志发送 AWS Elemental MediaTailor logs

您可以使用公开的日志来提高灵活性，并控制从回放配置 MediaTailor 发出的日志的交付位置。

使用已售日志， MediaTailor 将与配置相关的所有日志活动发送到 Amazon L CloudWatch ogs。CloudWatch 然后，日志会将您指定的日志百分比发送到所选目的地。支持的目标是亚马逊 CloudWatch 日志组、亚马逊 S3 存储桶或 Amazon Data Firehose 流。

由于销售的日志以批量折扣价提供，因此与直接向 CloudWatch 日志发送日志相比，您可以节省成本。有关定价，请参阅亚马逊定 CloudWatch 价中“日志”选项卡上的供应商日志。

要使用已售日志，必须执行以下操作：

1. 添加权限.
2. 创建日志传送目的地.
3. 在日志中配置 CloudWatch 日志传输.
4. 启用供应商登录 MediaTailor.

有关已售日志的更多信息，请参阅日志用户指南中的启用 AWS 服务 CloudWatch 日志记录。MediaTailor 支持 V2 版本的已售日志。

第 1 步：添加 MediaTailor 日志传输权限

设置供应商日志的人员必须具有创建传送目的地、配置日志传送和启用供应商登录的权限。MediaTailor使用以下策略确保您拥有设置销售日志的适当权限。

CloudWatch 日志和交付目的地的政策

Amazon L CloudWatch ogs 用户指南中的以下部分提供了使您能够处理日志中的 CloudWatch 日志和配送目的地的政策。如果您将日志发送到多个位置，则可以将策略声明合并为一个策略，而不是创建多个策略。

- 日志已发送到 CloudWatch 日志
- 发送到 Amazon S3 的日志
- 日志已发送到 Firehose

从控制台进行设置的策略

如果您通过控制台而不是 API 设置供应商日志交付 AWS CLI，则必须在政策中拥有以下额外权限。

供应商登录政策 MediaTailor

要在中创建、查看或修改交付的已售日志 MediaTailor，您必须在策略中拥有以下权限。

有关添加权限和使用策略的信息，请参阅的身份和访问管理 AWS Elemental MediaTailor。

第 2 步：为 MediaTailor 日志创建传送目的地

创建用于发送日志的资源。记录资源的 ARN，用于在后续步骤中配置日志传输。

CloudWatch 日志日志组传送目的地

使用以下任一方法来帮助创建日志组。

- 有关控制台，请参阅 Amazon Logs 用户指南中的 CloudWatch CloudWatch 日志中的创建日志组。
- 有关 API，请参阅CreateLogGroup《亚马逊 CloudWatch 日志 API 参考》。
- 有关软件开发工具包和 CLI，请参阅`CreateLogGroup`与 AWS 软件开发工具包一起使用或 AWS CLI《亚马逊 CloudWatch 日志用户指南》。

亚马逊 S3 存储桶交付目的地

使用以下任一方法来帮助创建存储桶。

- 有关控制台、软件开发工具包和 CLI，请参阅亚马逊简单存储服务用户指南中的创建存储桶。
- 有关 API，请参阅CreateBucket《亚马逊简单存储服务 API 参考》。

Firehose 直播交付目的地

有关创建直播的帮助，请参阅 Amazon D ata Firehose 开发者指南中的从控制台创建 Firehose 直播。

第 3 步：为 MediaTailor 回放配置启用供应商日志

创建或更新播放配置，该配置将向您在上一步中创建的传送目标发送日志。记录配置的名称，用于在后续步骤中配置日志传输。

- 要通过控制台启用已售日志，请使用创建配置或编辑配置设置编辑配置来访问日志记录设置。对于日志记录策略，请选择 Vended logs 。
- 要通过 API 启用已售日志，必须具有现有配置。用于`ConfigureLogsForPlaybackConfiguration`添加日志记录策略`Vended logs`。

如果您使用的是将日志直接发送到 CloudWatch 日志的传统日志 MediaTailor 记录策略，并希望迁移到销售日志，请参阅迁移日志策略。

> **Important**
>
> 如果您将日志策略从传统日志更改 CloudWatch 为出售日志， MediaTailor 将在保存更新后立即进行此更改。在完全配置供应商日志记录之前，您将停止接收日志。

步骤 4：在日志中 CloudWatch 配置日志传输

在 CloudWatch 日志中，必须创建三个元素来表示日志传输的各个部分。《亚马逊 CloudWatch 日志 API 参考》CreateDelivery中详细描述了这些元素。使用 CloudWatch 日志 API 配置交付的高级步骤如下所示。

在日志 (API) 中配置 CloudWatch 日志传输

1. `PutDeliverySource`用于添加日志源。

   A `DeliverySource` 代表生成日志的播放配置。您需要播放配置的名称才能创建`DeliverySource`。

2. 用于`PutDeliveryDestination`添加写入日志的目的地。

A DeliveryDestination 代表配送目的地。您需要日志组、存储桶或流的 ARN 才能创建。DeliveryDestination

3. PutDeliveryDestinationPolicy如果您要跨账户传送日志，请使用。

   如果传送目的地与播放配置位于不同的账户中，则需要DeliveryDestinationPolicy. 此策略允许 CloudWatch 日志将日志传送到DeliveryDestination。

4. 用于CreateDelivery将链接DeliverySource到DeliveryDestination.

   A Delivery 代表DeliverySource和之间的连接DeliveryDestination。

迁移您的 AWS Elemental MediaTailor 记录策略

如果您将日志策略从传统日志更改 CloudWatch 为已售日志，则 MediaTailor 将在保存更新后立即进行此更改。为避免日志记录工作流程中断，请使用以下步骤迁移您的日志记录策略。

1. 请按照 使用已售日志 中所述的步骤操作。对于启用供应商登录 MediaTailor，选择两种日志记录策略（Vended 日志和 Legacy CloudWatch）。

   MediaTailor 将通过出售的日志发送日志，并直接发送到 CloudWatch 日志。

2. 根据您的日志策略和交付目的地，对工作流程进行必要的更改。
3. 重新审视启用供应商登录 MediaTailor并 CloudWatch从日志策略中移除 Legacy。

编写 AWS Elemental MediaTailor 直接登录到亚马逊 CloudWatch 日志

MediaTailor 生成包含会话活动和广告决策服务器交互详细信息的日志，并将其写入亚马逊 CloudWatch。日志按顺序描述会话期间发生的活动。

MediaTailor 还可以使用供应商日志来灵活地交付日志和批量折扣定价。有关已售日志的信息，请参阅使用已售日志。

主题

- 亚马逊 CloudWatch 日志的权限
- “As Run” 日志 AWS Elemental MediaTailor 渠道组装
- AWS Elemental MediaTailor 亚马逊日志见解中的 ADS CloudWatch 日志分析

亚马逊 CloudWatch 日志的权限

使用 AWS Identity and Access Management (IAM) 创建向亚马逊提供 AWS Elemental MediaTailor 访问权限的角色 CloudWatch。您必须执行这些步骤才能为您的账户发布 CloudWatch 日志。CloudWatch 自动发布您的账户指标。

允许 MediaTailor 访问 CloudWatch

1. 使用 https://console.aws.amazon.com/iam/ 打开 IAM 控制台。
2. 在 IAM 控制台的导航窗格中，选择角色，然后选择创建角色。
3. 选择其他 AWS 账户角色类型。
4. 在“账户 ID”中，输入您的 AWS 账户 ID。
5. 选择 Require external ID (需要外部 ID) 并输入 **`Midas`**。此选项会在信任策略中自动添加一个条件，即仅当请求包含正确的 `sts:ExternalId` 时，该服务才代入该角色。
6. 选择下一步: 权限。
7. 添加指定此角色可完成的操作的权限策略。从以下选项中选择一项，然后选择 Next: Review (下一步: 审核)：

   - CloudWatchLogsFullAccess提供对亚马逊 CloudWatch 日志的完全访问权限
   - CloudWatchFullAccess提供对亚马逊的完全访问权限 CloudWatch
8. 对于角色名称，输入 **`MediaTailorLogger`**，然后选择创建角色。
9. 在 Roles (角色) 页面上，选择您刚刚创建的角色。
10. 要更新委托人，请编辑信任关系：

    1. 在角色的 Summary (摘要) 页上，选择 Trust relationship (信任关系) 选项卡。
    2. 选择编辑信任关系。
    3. 在政策文件中，将委托人更改为 MediaTailor 服务。它应如下所示：

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

       整个策略的内容现在应如下所示：
    4. 选择更新信任策略。

“As Run” 日志 AWS Elemental MediaTailor 渠道组装

日志组中的 As Run CloudWatch `MediaTailor/Channel/AsRunLog` 日志显示有关程序和广告插播的信息。

创建频道时，As Run 日志在默认情况下处于禁用状态。使用控制台或 AWS Command Line Interface (AWS CLI)，您可以为账户中的每个频道启用和禁用 As Run 日志状态。

启用 As Run 日志时， MediaTailor 会自动创建一个服务相关角色，该角色 MediaTailor 允许在您的日志账户中写入和管理 As Run CloudWatch 日志。有关服务关联角色的更多信息，请参阅将服务相关角色用于 MediaTailor。

> **Note**
>
> As Run Log 目前仅支持默认程序。目前，它不支持由程序规则创建的 AlternateMedia。这意味着它目前不为 AlternateMedia 生成 As Run Log。

主题

- 启用 As Run 日志
- 禁用 As Run 日志

启用 As Run 日志

要启用 As Run 日志，请指定频道名称并为该频道启用 A s Run 日志类型。

Console

在创建频道时启用 As Run 日志

1. 登录 AWS 管理控制台 并打开 MediaTailor 控制台，地址为https://console.aws.amazon.com/mediatailor/。
2. 在导航窗格中，选择 “频道汇编” > “频道” 。
3. 在导航栏上，选择创建频道。
4. 在 “设置频道详细信息” 、“配置输出” 和 “访问控制” 窗格中，根据需要配置您的频道。
5. 在 “访问控制” 窗格中，选择 “下一步” 。
6. 在 “日志” 窗格的 “日志类型” 下，选择 “启用即运行” 以启用 As Run 日志。

更新频道时启用 As Run 日志

> **Note**
>
> 如果该频道当前正在运行，则必须先停止该频道，然后才能对其进行更新。停止频道后，您可以选择 “操作” > “编辑” 开始更新频道。

1. 登录 AWS 管理控制台 并打开 MediaTailor 控制台，地址为https://console.aws.amazon.com/mediatailor/。
2. 在导航窗格中，选择 “频道汇编” > “频道” 。
3. 选择要更新的频道以启用 As Run 日志。
4. 选择操作 > 编辑。
5. 在 “设置频道详细信息” 、“配置输出” 和 “访问控制” 窗格中，根据需要更新您的频道配置。
6. 在 “访问控制” 窗格中，选择 “下一步” 。
7. 在 “日志” 窗格的 “日志类型” 下，选择 “启用即运行” 以启用 As Run 日志。

要启用 As Run 日志，请从 日志记录 选项卡

> **Note**
>
> 如果频道当前正在运行，则必须使用 “记录” 选项卡，而不是选择 “操作” > “编辑” 来启用 As Run 日志。

1. 登录 AWS 管理控制台 并打开 MediaTailor 控制台，地址为https://console.aws.amazon.com/mediatailor/。
2. 在导航窗格中，选择 “频道汇编” > “频道” 。
3. 选择要为其启用 As Run 日志的频道。
4. 在频道名称下方的导航栏中，选择 L ogging 。
5. 在 “日志” > “日志类型” 下 ，选择 As Run 以启用 As Run 日志。

AWS Command Line Interface (AWS CLI)

启用 As Run 日志

运行 configure-logs-for-channel 命令并为所需参数指定相应的值。

此示例是针对 Linux、macOS 或 Unix 进行格式化的，它使用反斜杠 (\ ) 行继续符来提高可读性。

```
$ aws mediatailor configure-logs-for-channel \
--channel-name MyChannel \
--log-types AS_RUN
```

此示例的格式是针对微软 Windows 的，它使用插入符号 (^) 换行符来提高可读性。

```
C:\> aws mediatailor configure-logs-for-channel ^
--channel-name MyChannel ^
--log-types AS_RUN
```

其中：

- *MyChannel*是您拥有并想要为其启用 As Run 日志的频道的名称。

如果命令成功运行，则您将收到类似于以下内容的输出：

```
{
    "ChannelName": "MyChannel",
    "LogTypes": [
       "AS_RUN"
    ]
}
```

禁用 As Run 日志

要为已启用 As Run 的频道禁用 As Run 日志，请指定频道名称并禁用该频道的 As Run 日志类型。

Console

更新频道时禁用 As Run 日志

> **Note**
>
> 如果该频道当前正在运行，则必须先停止该频道，然后才能对其进行更新。停止频道后，您可以选择“操作”>“编辑”开始更新频道。

1. 登录 AWS 管理控制台 并打开 MediaTailor 控制台，地址为https://console.aws.amazon.com/mediatailor/。
2. 在导航窗格中，选择 “频道汇编” > “频道” 。
3. 选择要更新的频道以启用 As Run 日志。
4. 选择操作 > 编辑。
5. 在 “设置频道详细信息” 、“配置输出” 和 “访问控制” 窗格中，根据需要更新您的频道配置。
6. 在 “访问控制” 窗格中，选择 “下一步” 。
7. 在 “日志” 窗格的 “日志类型” 下，清除 “启用即运行” 以禁用 As Run 日志。

要禁用 As Run 日志，请从 日志记录 选项卡

> **Note**
>
> 如果频道当前正在运行，则必须使用 “记录” 选项卡，而不是选择 “操作” > “编辑” 来禁用 As Run 日志。

1. 登录 AWS 管理控制台 并打开 MediaTailor 控制台，地址为https://console.aws.amazon.com/mediatailor/。
2. 在导航窗格中，选择 “频道汇编” > “频道” 。
3. 选择要禁用 As Run 日志的频道。
4. 在频道名称下方的导航栏中，选择 L ogging 。
5. 在 “日志” > “日志类型” 下 ，清除 “运行时运行” 以禁用 As Run 日志。

AWS Command Line Interface (AWS CLI)

禁用 As Run 日志

运行 configure-logs-for-channel 命令并为所需参数指定相应的值。

此示例是针对 Linux、macOS 或 Unix 进行格式化的，它使用反斜杠 (\ ) 行继续符来提高可读性。

```
$ aws mediatailor configure-logs-for-channel \
--channel-name MyChannel \
--log-types
```

此示例的格式是针对微软 Windows 的，它使用插入符号 (^) 续行符来提高可读性。

```
C:\> aws mediatailor configure-logs-for-channel ^
--channel-name MyChannel ^
--log-types
```

其中：

- *MyChannel*是您拥有并想要禁用 As Run 日志的频道的名称。

如果命令成功运行，则您将收到类似于以下内容的输出：

```
{
    "ChannelName": "MyChannel",
    "LogTypes": []
}
```

AWS Elemental MediaTailor 亚马逊日志见解中的 ADS CloudWatch 日志分析

您可以使用 Amazon Logs Insights 查看和查询 AWS Elemental MediaTailor 广告决策服务器 (ADS) CloudWatch 日志。 MediaTailor 将事件日志发送到以 CloudWatch 进行正常处理和出现错误情况。日志遵循 JSON 架构。通过 CloudWatch Logs Insights，您可以按时间范围选择日志，然后对它们运行查询。

有关一般信息，请参阅使用日志见解分析 CloudWatch 日志数据。

> **Note**
>
> 要访问日志，您需要获得访问亚马逊的权限 CloudWatch。有关说明，请参阅亚马逊 CloudWatch 日志的权限。

使用 CloudWatch 控制台查看和查询 ADS 日志

1. 在处打开 CloudWatch 控制台https://console.aws.amazon.com/cloudwatch/。
2. 在导航窗格中的 Logs (日志) 下，选择 Insights (见解)。
3. 在搜索栏中输入**AdDec**，然后从下拉列表中选择MediaTailor/AdDecisionServerInteractions。

4. （可选）调整要研究的时间段。
5. （可选）更改对话框中的查询。有关一般指导，请参阅CloudWatch 日志见解查询语法。有关 MediaTailor ADS 的查询的示例，请参阅 查询 ADS 日志。
6. 选择运行查询。该查询可能需要几秒钟时间，在此期间，将显示 Cancel (取消) 来代替 Run query (运行查询)。
7. （可选）要将结果导出为 CSV 文件，请选择 Actions (操作)，然后选择 Download query results (CSV) (下载查询结果(CSV))。

> **Note**
>
> 控制台限制了在查询结果中返回和导出的记录数量，因此，对于批量数据，请使用 API、 AWS Command Line Interface (AWS CLI) 或 SDK。

主题

- 查询 ADS 日志

查询 ADS 日志

CloudWatch Logs Insights 为查询日志提供了一组丰富的选项。有关查询语法的详细信息，请参阅CloudWatch 日志见解查询语法。此部分提供了一些常见查询示例来帮助您开始使用 ADS 日志查询。所有查询都针对当前时间范围设置的日志运行。

以下查询从 ADS 日志中检索所有信息。

```
fields @timestamp, eventType, sessionId, requestId, @message
| sort sessionId, @timestamp asc
```

以下查询检索对 ADS 的所有请求。此查询显示了一种检索 MediaTailor 日志请求标头内容的方法。

```
fields @timestamp, adsRequestUrl, requestHeaders.0.value as @userAgent,
 requestHeaders.1.value as @xForwardedFor, sessionId, requestId
| filter eventType = "MAKING_ADS_REQUEST"
| sort @timestamp asc
```

以下查询检索为给定会话 MediaTailor 插入的广告。

```
fields @timestamp, sessionId, requestId, @message
| filter eventType = "FILLED_AVAIL"
| sort @timestamp asc
```

以下查询检索代表玩家 MediaTailor 调用的跟踪 URL。

```
fields @timestamp, beaconInfo.trackingEvent, beaconInfo.beaconUri,
 beaconInfo.headers.0.value as @userAgent, beaconInfo.headers.1.value as
 @xForwardedFor, sessionId, requestId
| filter eventType = "BEACON_FIRED"
| sort @timestamp asc
```

以下查询通过按 `sessionId` 筛选结果来检索特定播放会话的信息。

```
fields @timestamp, eventType, sessionId, requestId, @message
| filter sessionId = "0aaf6507-c6f9-4884-bfe7-f2f841cb8195"
| sort @timestamp asc
```

以下查询通过按 `requestId` 筛选结果来检索单个请求的信息。

```
fields @timestamp, eventType, sessionId, requestId, @message
| filter requestId = "f5d3cf39-6258-4cf1-b3f6-a34ff8bf641d"
| sort @timestamp asc
```

以下查询检索记录的每种事件类型的日志条目计数。

```
fields eventType
| stats count() as @eventCount by eventType
```

以下查询检索已跳过广告的所有效用的效用 ID 和已跳过广告列表。

```
fields avail.availId
| parse @message '"skippedAds":[*]' as @skippedAdsList
| filter ispresent(@skippedAdsList)
```

控制音量 AWS Elemental MediaTailor logs

MediaTailor 广告插入会话日志有时会很冗长。为了降低日志成本，您可以定义 MediaTailor 发送到 Amazon Logs 的会话 CloudWatch 日志的百分比。例如，如果您的播放配置有 1000 个广告插入会

话，并且您将启用百分比值设置为60，则会将其中 600 个会话的日志 MediaTailor 发送到 CloudWatch 日志。 MediaTailor 随机决定向哪个会话发送日志。如果要查看特定会话的日志，则可以使用调试日志模式。

当您设置日志记录百分比时， MediaTailor 会自动创建一个服务相关角色， MediaTailor 该角色授予向您的账户写入 CloudWatch 日志所需的权限。有关如何 MediaTailor 使用服务相关角色的信息，请参阅将服务相关角色用于 MediaTailor。

创建日志配置

要控制 MediaTailor 写入日志的会话 CloudWatch 日志的百分比，您可以为回放配置创建日志配置。创建日志配置时，需要指定播放配置名称和启用百分比值。

Console

为创建日志配置 现有 播放配置

1. 登录 AWS 管理控制台 并打开 MediaTailor 控制台，地址为https://console.aws.amazon.com/mediatailor/。
2. 在回放配置窗格上，选择要为其设置日志配置的播放配置。
3. 选择编辑。
4. 在 “日志配置” 下，指定启用百分比值。

为创建日志配置 新 播放配置

- 按照日志配置中的程序操作。

AWS Command Line Interface (AWS CLI)

为现有回放配置创建日志配置

要使用创建日志配置，请运行 configure-logs-for-playback-confor-for-playback-configuration 命令并为所需参数指定相应的值。 AWS CLI

此示例是针对 Linux、macOS 或 Unix 进行格式化的，它使用反斜杠 (\ ) 行继续符来提高可读性。

```
$ aws mediatailor configure-logs-for-playback-configuration \
--percent-enabled 10 \
--playback-configuration-name MyPlaybackConfiguration
```

此示例的格式是针对微软 Windows 的，它使用插入符号 (^) 续行符来提高可读性。

```
C:\> aws mediatailor configure-logs-for-playback-configuration ^
--percent-enabled 10 ^
--playback-configuration-name MyPlaybackConfiguration
```

其中：

- *percent-enabled*是 MediaTailor发送到 Logs 的回放配置会话 CloudWatch 日志的百分比。
- *playback-configuration-name*是用于设置日志配置设置的播放配置的名称。

如果命令成功运行，则您将收到类似于以下内容的输出：

```
{
    "PercentEnabled": 10,
    "PlaybackConfigurationName": "MyPlaybackConfiguration"
}
```

为创建日志配置 新 播放配置

- 使用 put-playback-c onfiguration 命令的`configure-logs-for-playback-configuration`选项。

停用日志配置

创建日志配置后，无法将其删除，只能将其停用。要停用日志配置，请使用 MediaTailor 控制台或 API 将启用百分比值设置为 0。这将关闭该播放配置的所有会话记录。

如果您想删除账户中 MediaTailor 用于日志配置的服务相关角色，则必须先停用所有日志配置。有关如何删除服务相关角色的信息，请参阅将服务相关角色用于 MediaTailor。

Console

停用回放配置上的日志配置

1. 登录 AWS 管理控制台 并打开 MediaTailor 控制台，地址为https://console.aws.amazon.com/mediatailor/。
2. 在回放配置窗格上，选择要停用日志配置的播放配置。
3. 选择编辑。

4. 在“日志配置”下，将启用百分比值设置为0。这将关闭此播放配置的所有会话记录。
5. 选择保存。

AWS Command Line Interface (AWS CLI)

停用日志配置

- 0使用“配置日志以percent-enabled供回放”配置命令将该值设置为。

筛选 AWS Elemental MediaTailor 日志和事件

从中的回放配置发出的日志 MediaTailor 包括有关播放会话期间发生的各种活动的信息。这些活动在日志的事件类型中标识。默认情况下会记录许多事件。为了帮助控制亚马逊中的日志成本 CloudWatch，您可以指定 MediaTailor 发出的日志。

MediaTailor 允许您控制日志筛选，因此您可以执行以下操作：

- 指定要从日志中排除的日志事件
- 启用记录来自广告决策服务器 (ADS) 的原始响应

您可以为每个回放会话单独设置这些日志过滤首选项，或者将这些首选项设置为回放配置的所有回放会话的默认设置。

- 要根据每个会话筛选日志，请在回放会话初始化请求中包含查询参数。
- 要根据每次播放配置筛选日志，请使用 MediaTailor 控制台或 API 在播放配置设置中指明您的偏好。

以下部分提供了在会话和回放配置上启用日志过滤的说明。

Per-session 日志过滤器

要为每个会话定义自定义的日志详细信息级别，请将以下参数附加到您的初始服务器端或客户端播放会话请求中。以逗号分隔的格式向参数中添加值以表示要包含或排除的事件：

- aws.adsInteractionLogPublishOptInEventTypes接收特定广告决策服务器 (ADS) 互动的日志。
- aws.adsInteractionLogExcludeEventTypes停止接收特定 ADS 互动的日志。
- aws.manifestServiceLogExcludeEventTypes停止接收特定清单服务交互的日志。

有关发 MediaTailor 出的日志和事件类型的列表，请参见清单日志ADS 日志、和转码日志。

如果您没有传递任何查询参数进行日志筛选，则 MediaTailor 会将所有日志写入您的传送目的地。

Example使用日志过滤器初始化服务器端会话

要从清单日志GENERATED_MANIFEST和 ADS 日志MAKING_ADS_REQUEST中排除和PARSING_ERROR事件，会话初始化请求将如下所示：

```
GET <mediatailorURL>/v1/master/<hashed-account-id>/<origin-id>/index.m3u8?
aws.logMode=DEBUG&aws.manifestServiceLogExcludeEventTypes=GENERATED_MANIFEST,PARSING_ERROR&aws.
```

要启用来自 ADS 的原始日志，请添加AdsInteractionPublishOptInEventType参数RAW_ADS_RESPONSE值：

```
GET <mediatailorURL>/v1/master/<hashed-account-id>/<origin-id>/index.m3u8?
aws.adsInteractionPublishOptInEventType=RAW_ADS_RESPONSE
```

Example使用日志过滤器初始化客户端会话

要在客户端会话初始化期间排除日志事件，请在您的客户端 POST 请求中包含availSuppression和日志类型参数。 MediaTailor有关如何构建客户端播放会话请求的更多信息，请参阅Client-side 广告追踪。以下示例从您的清单日志CONFIG_SECURITY_ERROR和 ADS 日志中MAKING_ADS_REQUEST排除和PARSING_ERROR事件。

```
POST parent.m3u8
   {
       "adsInteractionLog": {
           ...
           "excludeEventTypes": [
               "MAKING_ADS_REQUEST"
           ]
       },
       "manifestServiceLog": {
           ...
           "excludeEventTypes": [
               "GENERATED_MANIFEST",
               "PARSING_ERROR"
           ]
       },
      "logMode": "DEBUG"
   }
```

要启用来自 ADS 的原始日志，请添加publishOptInEventTypes参数RAW_ADS_RESPONSE 值：

```
POST parent.m3u8
   {
       "adsInteractionLog": {
           "publishOptInEventTypes": ["RAW_ADS_RESPONSE"],
           "excludeEventTypes": [
               "MAKING_ADS_REQUEST"
           ]
       },
       "manifestServiceLog": {
           ...
           "excludeEventTypes": [
               "GENERATED_MANIFEST",
               "PARSING_ERROR"
           ]
       },
       "logMode": "DEBUG"
   }
```

Per-playback 配置日志过滤器

使用回放配置的设置来定义从该回放配置 MediaTailor 发出的默认日志事件类型。 MediaTailor 对所有不包括会话初始化请求筛选查询参数的会话使用这些默认日志筛选设置。

您可以选择执行以下操作：

- 接收特定广告决策服务器 (ADS) 互动的日志。
- 排除特定 ADS 互动的日志。
- 排除特定清单服务交互的日志。

要从 MediaTailor 控制台设置这些设置，请参阅创建配置。有关 MediaTailor API，请参阅 AWS Elemental MediaTailor API 参考PutPlaybackConfiguration中的。

有关发 MediaTailor 出的日志和事件类型的列表，请参见清单日志ADS 日志、和转码日志。

正在生成 AWS Elemental MediaTailor 调试日志

使用调试日志对 MediaTailor 广告插入播放会话问题进行故障排除。要生成调试日志，请在玩家的请求中将日志模式设置为调试 MediaTailor。对于服务器端报告，请在回放请求中设置日志模式。对于客户端报告，请在会话初始化请求中设置日志模式。

当日志模式设置为调试时，将所有日志事件类型 MediaTailor 写入 CloudWatch 日志。日志提供有关以下事件的信息。有关调试日志中生成的数据的完整列表，请参阅调试日志字段。

- 原始交互 -有关与源服务器 MediaTailor 交互的详细信息。例如，源清单响应、清单类型和来源 URL。
- 生成的清单 -有关来自的播放会话响应的详细信息 MediaTailor。例如，生 MediaTailor 成的清单。
- 会话初始化 -会话初始化详细信息，例如会话 ID。

要自定义您在每个会话中收到的日志事件类型，请参阅筛选日志和事件。

先决条件

要将日志模式设置为调试，首先需要授予向其发送日志的 MediaTailor 权限（如果尚未授权）。CloudWatch授予访问权 MediaTailor 限后 CloudWatch，即可启用调试日志模式。有关如何授予访问 MediaTailor 权限的信息， CloudWatch 请参阅为亚马逊设置权限 CloudWatch。

如何将日志模式设置为调试

本节介绍如何将服务器端报告和客户端报告的日志模式设置为调试。

Server-side 报告

要进行服务器端报告，请在玩家向 HLS 或 DASH MediaTailor 终端节点发GET HTTP出的播放请求中包含?aws.logMode=DEBUG查询参数和值。有关服务器端报告的一般信息，请参阅Server-side报告。

> **Important**
>
> DEBUG 值区分大小写。

包含以下内容的播放请求?aws.logMode=DEBUG如下所示：

Example向 HLS 终端节点发送播放请求

```
GET <mediatailorURL>/v1/master/<hashed-account-id>/<origin-id>/<asset-id>?
aws.logMode=DEBUG
```

将日志模式设置为调试后，我们建议您验证调试日志记录会话是否处于活动状态。要验证调试会话是否处于活动状态，请检查会话 ID 是否有任何 CloudWatch 日志。会话 ID 包含在 MediaTailor 提供的播放端点中。有关更多信息，请参阅 Verify that the debug log mode is active for your playback session。

Client-side 报告

对于客户端报告，请在客户端向 MediaTailor /v1/session 端点发送的POST HTTP会话初始化请求正文中包含logMode密钥和DEBUG值。有关客户端报告的一般信息，请参阅Client-Side报告。

> **Important**
>
> DEBUG 值区分大小写。

将日志模式设置为调试后，我们建议您验证调试会话是否处于活动状态。要验证调试会话是否处于SESSION_INITIALIZED活动状态，请确认 CloudWatch 日志中存在与会话 ID 关联的事件。会话 ID 包含在 MediaTailor 提供的播放端点中。有关更多信息，请参阅 Verify that the debug log mode is active for your playback session。

最大活跃调试会话数

您最多可以有 10 个有效的调试日志会话。当您的播放器向发送会话初始化或播放请求时 MediaTailor，会 MediaTailor 检查是否已达到限制。如果有，请 MediaTailor 检查是否有任何过时的会话。如果在一段时间内未访问过会话，则会话已过期。对于直播，此时间段为 10 分钟，对于 VOD 直播，时间段为 30 分钟。

如果已达到最大活动调试日志会话限制，则调试日志不会写入会话的 CloudWatch 日志。如果您在会话日志中看不到调试 CloudWatch 日志，则可能已达到此限制。要确认是否已达到限制，请参阅Verify that the debug log mode is active for your playback session。

调试日志字段

下表列出了 MediaTailor 写入的调试日志字段 CloudWatch。

字段	说明
awsAccountId	你的 AWS 账户 身份证。
customerId	您的 MediaTailor 客户 ID。
eventTimestamp	与调试日志事件关联的 ISO 8601 时间戳。
eventType	调试日志事件的类型。 值：

字段	说明
	• ORIGIN_INTERACTION — 有关与源服务器 MediaTailor 交互的详细信息。例如，源清单响应、清单类型和来源 URL。 • GENERATED_MANIFEST — 有关来自的播放会话响应的详细信息 MediaTailor。例如，生 MediaTailor成的清单。 • SESSION_INITIALIZED — 会话初始化详细信息，例如会话 ID。
originRequestUrl	为该请求检索到的源服务器的 URL。
mediaTailorPath	被调用的 MediaTailor 端点，包括初始清单请求 MediaTailor 中传递给的任何参数。
requestId	向的特定 HTTP 请求的 ID MediaTailor。
responseBody	响应正文中的清单来自 MediaTailor。这要么是原始来源清单，要么是生成的清单 MediaTailor。
sessionId	播放会话 ID。
sessionType	播放会话的类型。 值：HLS、DASH

阅读调试日志

MediaTailor 将调试日志写入亚马逊 CloudWatch 日志。典型的 CloudWatch 日志费用适用。使用 CloudWatch Insights 读取调试日志。有关如何使用 CloudWatch 日志见解的信息，请参阅 AWS 日志用户指南中的使用 CloudWatch 日志见解分析 CloudWatch 日志数据。

> **Note**
>
> 调试日志可能需要几分钟才能显示在中 CloudWatch。如果您没有看到日志，请等待几分钟，然后重试。如果您仍然看不到日志，则可能是您已达到活动调试日志会话的最大数量。要验证是否如此，请运行 CloudWatch 查询以查看是否已为您的播放会话初始化了调试会话。有关更多信息，请参阅 Verify that the debug log mode is active for your playback session。

示例

本节包括可用于读取 MediaTailor调试日志数据的示例查询。

Example 1：验证您的播放会话的调试日志模式是否处于活动状态

```
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：查看来自您的来源的回复

```
fields @timestamp, responseBody, @message, mediaTailorPath
 | filter eventType = "ORIGIN_MANIFEST" and sessionId = "32002de2-837c-4e3e-9660-
f3075e8dfd90"
```

Example 3：查看 MediaTailor 为给定会话生成的清单

```
fields @timestamp, responseBody, @message
 | filter mediaTailorPath like "/v1/master/" and eventType = "GENERATED_MANIFEST" and
 sessionId = "32002de2-837c-4e3e-9660-f3075e8dfd90"
```

Example 4：查看给定 **requestId #####**

使用此查询查看源清单和生成的清单 MediaTailor。

```
fields @timestamp, responseBody, @message, mediaTailorPath
 | filter requestId = "e5ba82a5-f8ac-4efb-88a0-55bed21c45b4"
```

监控 AWS Elemental MediaTailor 使用亚马逊 CloudWatch 指标

您可以使用监控 AWS Elemental MediaTailor 指标 CloudWatch。 CloudWatch 收集有关服务性能的原始数据，并将这些数据处理为可读的、近乎实时的指标。这些统计数据会保存 15 个月，从而使您能够访问历史信息，并能够更好地了解您的 Web 应用程序或服务的执行情况。此外，可以设置用于监测特定阈值的警报，并在达到相应阈值时发送通知或执行操作。有关更多信息，请参阅亚马逊 CloudWatch 用户指南。

当你调查陈旧清单时，指标可能很有用。有关更多信息，请参阅 使用指标诊断陈旧清单。

指标的分组首先依据服务命名空间，然后依据每个命名空间内的各种维度组合。

使用 CloudWatch 控制台查看指标

1. 在处打开 CloudWatch 控制台https://console.aws.amazon.com/cloudwatch/。
2. 在导航窗格中，选择指标。
3. 在 “所有指标” 下，选择MediaTailor命名空间。
4. 选择指标维度以查看指标（例如，originID）。
5. 指定要查看的时间段。

要查看指标，请使用 AWS Command Line Interface (AWS CLI)

- 在命令提示符处输入下面的命令：

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

AWS Elemental MediaTailor CloudWatch 指标

AWS Elemental MediaTailor 命名空间包含以下指标。默认情况下，这些指标将发布到您的账户。

渠道集成 (CA) 指标

在下表中，所有指标均可按渠道或渠道输出获得。

指标	说明
4xxErrorCount	4xx错误数。
5xxErrorCount	5xx错误数。
RequestCount	请求的总数。交易数量在很大程度上取决于玩家请求更新清单的频率以及玩家的数量。每个播放器请求计为一个事务。
TotalTime	应用程序服务器处理请求所用的时间，包括从客户端和网络接收字节以及向客户端和网络写入字节所用的时间。

Server-side Ad-insertion (SSAI) 指标

下表列出了服务器端广告插入指标。

指标	说明
AdDecisionServer.Ads	在您指定的 CloudWatch 时间段内，广告决策服务器 (ADS) 响应中包含的广告数量。 维度： • ConfigurationName • Account-level （无尺寸）
AdDecisionServer.Duration	在您指定的时间段内从 ADS MediaTailor 收到的所有广告的总 CloudWatch 时长，以毫秒为单位。此持续时间可以长于Avail.Duration 您指定的持续时间。 维度：ConfigurationName
AdDecisionServer.Errors	在您指定的时间段内从 ADS MediaTailor 收到的非 HTTP 200 状态码响应、空响应和超 CloudWatch 时响应的数量。 维度：ConfigurationName
AdDecisionServer.FillRate	在您指定的时间段内，ADS 响应填充相应的单个广告效用的速率的简单平均值。 要获得加权平均值，请计算 AdDecisionServer.Duration 占 Avail.Duration 的百分比。有关简单和加权平均值的更多信息，请参阅简单和加权平均值。 维度：ConfigurationName
AdDecisionServer.Latency	向 ADS 提出的请求的响应时间（以毫秒为单位） MediaTailor 。

指标	说明
	维度：ConfigurationName
AdDecisionServer.Timeouts	在您指定的时间段内向 ADS 发出的超 CloudWatch时请求的数量。 维度：ConfigurationName
AdNotReady	ADS 指向在您指定的时间段内，尚未由内部转码器服务转码的广告的次数。 此指标值较大可能会导致较低的整体 Avail.FillRate 。 维度：ConfigurationName
AdsBilled	根据插播量向客户收取 MediaTailor 账单的广告数量。 维度： • ConfigurationName • Account-level （无尺寸）
Avail.Duration	该时间段内可用的广告的计划总毫秒数。CloudWatch 计划总量基于原始清单中的广告可用时长。 维度： • ConfigurationName • Account-level （无尺寸）

指标	说明
Avail.FilledDuration	计划在该时间段内填满广告的广告可用时间毫秒数。 MediaTailor CloudWatch 维度： • ConfigurationName • Account-level （无尺寸）
Avail.FillRate	该 CloudWatch时间段内单个广告可用量填充 MediaTailor 费率的计划简单平均值。 要获得加权平均值，请计算 Avail.FilledDuration 占 Avail.Duration 的百分比。有关简单和加权平均值的更多信息，请参阅简单和加权平均值。 MediaTailor 可实现的最大 Avail.FillRate 受 AdDecisionServer.FillRate 的约束。如果 Avail.FillRate 较低，则将其与 AdDecisionServer.FillRate 进行比较。如果 AdDecisionServer.FillRate 较低，您的 ADS 可能没有在效用持续时间内返回足够的广告。 维度：ConfigurationName
Avail.Impression	服务器端报告期间 MediaTailor 触发曝光信标的广告数量。 MediaTailor 无论这些信标最终是否已交付，都会为每个至少具有一个曝光量 URI 的广告发出一个计数。使用此指标作为分母来计算视频完成率 (VCR) （参见）。Avail.Complete 维度： • ConfigurationName • Account-level （无尺寸）

指标	说明
Avail.Complete	服务器端报告期间 MediaTailor 触发完整信标的广告数量。 MediaTailor 无论信标最终是否已交付，都会为每个至少具有一个完整跟踪 URI 的广告发出一个计数。要计算视频完成率 (VCR)，请将总和除Avail.Complete 以的Avail.Impression 总和。 维度： • ConfigurationName • Account-level （无尺寸）
Avail.ObservedDuration	观察到的时间段内出现的广告可用总毫秒数。 CloudWatch Avail.ObservedDuration 在广告利用率结束时发出，基于广告无效期间清单中报告的区段的持续时间。 维度：ConfigurationName
Avail.ObservedFilledDuration	在此时间段内观察到的 MediaTailor 充斥广告的广告可用时间的毫秒数。 CloudWatch 维度：ConfigurationName
Avail.ObservedFillRate	观察到的单个广告 MediaTailor 填充费率的简单平均值在一段 CloudWatch时间内可用。 仅在 HLS 清单的第一个标签处发出。CUE-IN如果没有CUE-IN标签，则 MediaTailor 不发布此指标。 维度：ConfigurationName
Avail.ObservedSlateDuration	观测到的时间段内插入的 slate 的总毫秒数。 CloudWatch 维度：ConfigurationName

指标	说明
GetManifest.Age	清单的总时长（以毫秒为单位）。从来源创建清单到何时 MediaTailor发送个性化清单来衡量。 有关衡量清单期限的指标的更多信息，请参阅使用指标诊断陈旧清单。 维度：ConfigurationName
GetManifest.Errors	在您指定的时间段 MediaTailor 内生成清单 CloudWatch 时收到的错误数。 维度：ConfigurationName
GetManifest.Latency	请求生成清单的 MediaTailor 响应时间（以毫秒为单位）。 有关衡量清单期限的指标的更多信息，请参阅使用指标诊断陈旧清单。 维度：ConfigurationName
GetManifest.MediaTailorAge	清单的存储时间，以毫秒 MediaTailor 为单位。从何时 MediaTailor 收到原始响应到何时 MediaTailor 发送个性化清单来衡量。 有关衡量清单期限的指标的更多信息，请参阅使用指标诊断陈旧清单。 维度：ConfigurationName

指标	说明
Origin.Age	源拥有清单的时间量，以毫秒为单位。从源创建清单到何时 MediaTailor 发送原始请求来衡量。 所有origin.*指标都是针对直接从源头完成的请求发出的。它们不是为缓存的源响应发出的。 有关衡量清单期限的指标的更多信息，请参阅使用指标诊断陈旧清单。 维度：ConfigurationName
Origin.Errors	在您指定的时间段内从源服务器 MediaTailor 收到的非 HTTP 200 状态码响应和超 CloudWatch 时响应的数量。 所有origin.*指标都是针对直接从源头完成的请求发出的。它们不是为缓存的源响应发出的。 维度：ConfigurationName
Origin.ManifestFileSizeBytes	HLS 和 DASH 的源清单文件大小（以字节为单位）。通常，此指标与一起使用Origin.ManifestFileSizeTooLarge 。 所有origin.*指标都是针对直接从源头完成的请求发出的。它们不是为缓存的源响应发出的。 维度：ConfigurationName
Origin.ManifestFileSizeTooLarge	来自源的清单大小大于配置数量的响应数量。通常，此指标与一起使用Origin.ManifestFileSizeBytes 。 所有origin.*指标都是针对直接从源头完成的请求发出的。它们不是为缓存的源响应发出的。 维度：ConfigurationName

指标	说明
Origin.Timeouts	在您指定的时间段内向源服务器发出的超 CloudWatch 时请求的数量。 所有origin.*指标都是针对直接从源头完成的请求发出的。它们不是为缓存的源响应发出的。 维度：ConfigurationName
Requests	所有请求类型每秒的并发事务数。交易数量主要取决于玩家数量以及玩家请求更新清单的频率。每个播放器请求计为一个事务。 维度： • ConfigurationName • Account-level （无尺寸）
SkippedReason.DurationExceeded	由于 ADS 返回的广告持续时间大于指定的使用时长而未插入的广告数量。该指标的值过高可能会导致AdsBilled 和AdDecisionServer.Ads 指标之间的差异。有关跳过广告的原因的更多信息，请参阅跳过广告疑难解答。 维度： • ConfigurationName • Account-level （无尺寸）
SkippedReason.EarlyCueIn	由于时间过早CUE-IN，跳过的广告数量很多. 维度： • ConfigurationName • Account-level （无尺寸）

指标	说明
SkippedReason.ImportError	由于导入任务中的错误而跳过的广告数量。 维度： • ConfigurationName • Account-level（无尺寸）
SkippedReason.ImportInProgress	由于现有正在进行的导入任务而跳过的广告数量。 维度： • ConfigurationName • Account-level（无尺寸）
SkippedReason.InternalError	由于 MediaTailor 内部错误而跳过的广告数量。 维度： • ConfigurationName • Account-level（无尺寸）
SkippedReason.NewCreative	由于这是客户首次请求该资产，因此未插入的广告数量无济于事。在资产可以成功转码之前，该指标的较高值可能会暂时导致总体Avail.FillRate 水平降低。 维度： • ConfigurationName • Account-level（无尺寸）

指标	说明
SkippedReason.NoVariantMatch	由于广告和内容之间没有变体匹配而跳过的广告数量。 维度： • ConfigurationName • Account-level（无尺寸）
SkippedReason.PersonalizationThresholdExceeded	广告的持续时间超过此配置中的个性化阈值设置。 维度： • ConfigurationName • Account-level（无尺寸）
SkippedReason.ProfileNotFound	由于找不到转码配置文件而跳过的广告数量。 维度： • ConfigurationName • Account-level（无尺寸）
SkippedReason.TranscodeError	由于转码错误而跳过的广告数量。 维度： • ConfigurationName • Account-level（无尺寸）
SkippedReason.TranscodeInProgress	由于广告尚未进行转码而未插入的广告的数量。在资产成功转码之前Avail.FillRate ，该指标的高值可能会暂时导致总体水平降低。 维度： • ConfigurationName • Account-level（无尺寸）

> **Note**
>
> 对于 HLS Interstitials 会话，由于广告决策的延迟约束性，一些指标的表现会有所不同：
>
> - `Avail.ObservedFilledDuration`匹配，`Avail.FilledDuration`因为 MediaTailor 无法观察到实际的客户端播放行为。
> - `Avail.ObservedSlateDuration`根据资产列表的回复而不是观察到的播放来报告计划的暂存时长。
> - 以 “观察” 为前缀的指标提供了 HLS Interstials 会话的估计值。

函数指标

当您在播放配置中使用函数时，会发布以下指标。当在生命周期挂钩上配置函数时，总是会发出这些指标。无需选择加入或登录配置。

Hook-level 指标 — 每个生命周期挂钩执行一个数据点：

指标	说明
`PreSessionInitHook.Invocations`	调用会话前初始化挂钩的次数。 维度：`ConfigurationName`
`PreSessionInitHook.Errors`	导致错误的会话前初始化挂钩执行次数。 维度：`ConfigurationName`
`PreSessionInitHook.Latency`	会话前初始化挂钩的执行时间（以毫秒为单位）。 维度：`ConfigurationName`
`PreAdsRequestHook.Invocations`	广告前请求挂钩被调用的次数。 维度：`ConfigurationName`
`PreAdsRequestHook.Errors`	导致错误的广告前请求挂钩执行次数。 维度：`ConfigurationName`

指标	说明
PreAdsRequestHook.Latency	广告前请求挂钩的执行时间（以毫秒为单位）。 维度：ConfigurationName

Function-level 指标 — 每个函数执行一个数据点。这些指标包括其他维度 (FunctionId、FunctionType、HookType)，因此您可以确定哪个特定功能运行缓慢或失败：

指标	说明
Function.Invocations	单个函数的执行次数。 维度： • ConfigurationName • ConfigurationName ，FunctionId ，FunctionType ，HookType
Function.Errors	导致错误的单个函数执行次数。 维度： • ConfigurationName • ConfigurationName ，FunctionId ，FunctionType ，HookType
Function.Latency	单个函数的执行时间（以毫秒为单位）。 维度： • ConfigurationName • ConfigurationName ，FunctionId ，FunctionType ，HookType

简单和加权平均值

您可以检索广告对广告请求的响应以及广告 MediaTailor 填充情况的简单平均值 MediaTailor 和加权平均值：

- AdDecisionServer.FillRate和中提供了简单的平均值Avail.FillRate。这些是该时间段内各个效用的填充率百分比的平均值。简单平均值不考虑单个效用的持续时间之间的任何差异。
- 加权平均值是所有效用持续时间总和的填充率百分比。计算方法为 (AdDecisionServer.Duration*100)/Avail.Duration 和 (Avail.FilledDuration*100)/Avail.Duration。这些平均值反映了每个广告效用持续时间的差异，给予持续时间较长的效用更多的权重。

对于仅包含一个广告效用的时间段，AdDecisionServer.FillRate 提供的简单平均值等于 (AdDecisionServer.Duration*100)/Avail.Duration 提供的加权平均值。Avail.FillRate 提供的简单平均值等于 (Avail.FilledDuration*100)/Avail.Duration 提供的加权平均值。

示例

假设您指定的时间段具有以下两个广告效用：

- 第一个广告效用具有 90 秒的持续时间：
  - ADS 对该效用的响应提供了 45 秒的广告（50% 已填充）。
  - MediaTailor 占用 45 秒的可用广告时间（已填满 50%）。
- 第二个广告效用具有 120 秒的持续时间：
  - ADS 对该效用的响应提供了 120 秒的广告（100% 已填充）。
  - MediaTailor 占用 90 秒的可用广告时间（填充率为 75%）。

各指标如下所示：

- Avail.Duration 为 210，两个广告效用持续时间的总和：90 + 120。
- AdDecisionServer.Duration 为 165，两个响应持续时间的总和：45 + 120。
- Avail.FilledDuration 为 135，两个已填充持续时间的总和：45 + 90。
- AdDecisionServer.FillRate 为 75%，每个效用已填充百分比的平均数：(50% + 100%)/2。这是简单平均值。
- ADS 填充率的加权平均值为 78.57%，是 AdDecisionServer.Duration 占 Avail.Duration 的百分比：(165* 100)/210。此计算考虑到了持续时间中的差异。

- `Avail.FillRate` 为 62.5%，每个效用已填充百分比的平均数：(50% + 75%)/2。这是简单平均值。
- MediaTailor 可用填充率的加权平均值为64.29%，占以下百分比`Avail.Duration`：（135*100）/210。`Avail.FilledDuration`此计算考虑到了持续时间中的差异。

任何广告利用率最高`Avail.FillRate` MediaTailor 可达到 100%。ADS 可能会返回与效用中可用的广告时间相比更多的广告时间，但 MediaTailor 只能填充可用的时间。

AWS Elemental MediaTailor CloudWatch 维度

您可以使用以下维度筛选 AWS Elemental MediaTailor 数据。

维度	说明
`ConfigurationName`	指示指标所属的配置。适用于所有指标。
`FunctionId`	函数的标识符。仅适用于`Function.*` 指标。
`FunctionType`	函数的类型：`CUSTOM_OUTPUT` 、`HTTP_REQUEST` `SEQUENTIAL_EXECUTOR` 、或`CONCURRENT_EXECUTOR` 。仅适用于`Function.*` 指标。
`HookType`	触发该函数的生命周期挂钩：`PRE_SESSION_INITIALIZATION` 或`PRE_ADS_REQUEST` 。仅适用于`Function.*` 指标。

使用指标来诊断来自的过时清单 AWS Elemental MediaTailor

过时的清单是最近未更新的清单。根据各种因素（例如对下游系统的要求），不同的广告插入工作流程对清单必须经过多长时间才能被视为过时有不同的容忍度。您可以使用亚马逊 CloudWatch 指标来识别超过工作流程过时容限的清单，并帮助确定可能导致清单更新延迟的原因。

以下指标有助于识别陈旧清单及其原因。

有关 MediaTailor 发出的所有指标的信息，请参阅AWS Elemental MediaTailor CloudWatch 指标。

指标	定义	使用
GetManifest.Age	衡量清单的总时长，包括此配置Origin.Age 的GetManifest.MediaTailorAge 和。	您可以使用此指标来识别已超过更新阈值且已过时的清单。 对该指标设置警报，这样在提供过时清单时您会收到提醒。有关警报的信息，请参阅《亚马逊 CloudWatch 用户指南》中的指标警报。当您收到警报时，使用Origin.Age 和GetManifest.MediaTailorAge 来确定是否 MediaTailor 或来源导致了过时。
Origin.Age	衡量源在将清单发送到 MediaTailor 此配置之前有多长时间。 当响应来自内容分发网络 (CDN) 时，不会发出此指标。响应必须来自原点Origin.Age 才能发出。	当您使用识别陈旧清单时GetManifest.Age ，您可以分析Origin.Age 指标和指标，以确定哪些是导致清单过期GetManifest.MediaTailorAge 的原因。 如果您发现Origin.Age 这比原始服务器的典型处理时间长，则可能是上游系统导致了问题，您应该将诊断重点放在那里。
GetManifest.MediaTailorAge	衡量此配置的此清单存储 MediaTailor 了多长时间。	当您使用识别陈旧清单时GetManifest.Age ，您可以分析GetManifest.MediaTailorAge 指标和指标，以确定哪些是导致清单过期Origin.Age 的原因。 如果时间超过了GetManifest.MediaTailorAge 您通常的清单个性化时间 MediaTailor，则很可能是问题所致，您应该将诊断重点放在那里。 MediaTailor

指标	定义	使用
		GetManifest.Latency 可以进一步确定创建个性化清单需要多长时间。MediaTailor
GetManifest.Latency	衡量处理请求和 MediaTailor 为此配置创建个性化清单所花费的时间。	当您比较Origin.Age GetManifest.MediaTailorAge 并确定MediaTailor这是清单交付延迟的原因时，您可以分析该GetManifest.Latency 指标以确定清单个性化流程是否导致了清单过期。 GetManifest.MediaTailorAge 衡量清单存储的总时间 MediaTailor。GetManifest.Latency 衡量存储时间中有多少是根据请求对清单 MediaTailor 进行个性化设置。

记录 AWS Elemental MediaTailor API 调用

AWS Elemental MediaTailor 与 AWS CloudTrail一项服务集成，该服务提供用户、角色或 AWS 服务在中执行的操作的记录 MediaTailor。 CloudTrail 将所有 API 调用捕获 MediaTailor 为事件。捕获的调用包括来自 MediaTailor 控制台的调用和 MediaTailor 对 API 操作的代码调用。如果您创建跟踪，则可以启用向 Amazon S3 存储桶持续交付事件，包括的事件 MediaTailor。 CloudTrail 如果您未配置跟踪，仍可以在 CloudTrail 控制台的事件历史记录中查看最新事件。使用收集的信息 CloudTrail，您可以确定向谁发出了请求 MediaTailor、发出请求的 IP 地址、谁提出了请求、何时提出请求以及其他详细信息。

要了解更多信息 CloudTrail，请参阅AWS CloudTrail 用户指南。

AWS Elemental MediaTailor 中的信息 CloudTrail

CloudTrail 在您创建 AWS 账户时已在您的账户上启用。当活动发生在中时 AWS Elemental MediaTailor，该活动将与 CloudTrail 事件历史记录中的其他 AWS 服务事件一起记录在事件中。您可以在您的 AWS 账户中查看、搜索和下载最近的事件。有关更多信息，请参阅使用事件历史记录查看 CloudTrail 事件。

要持续记录 AWS 账户中的事件（包括的事件） AWS Elemental MediaTailor，请创建跟踪。跟踪可以 CloudTrail 将日志文件传输到 Amazon S3 存储桶。默认情况下，当您在控制台中创建跟踪时，该跟踪适用于所有 AWS 区域。该跟踪记录 AWS 分区中所有区域的事件，并将日志文件传输到您指定的 Amazon S3 存储桶。此外，您可以配置其他 AWS 服务，以进一步分析和处理 CloudTrail 日志中收集的事件数据。有关更多信息，请参阅下列内容：

- 为您的 AWS 账户创建跟踪
- AWS 与日志的服务集成 CloudTrail
- 为配置亚马逊 SNS 通知 CloudTrail
- 接收来自多个区域的 CloudTrail 日志文件和接收来自多个账户的 CloudTrail 日志文件

所有 AWS Elemental MediaTailor 操作均由 API 参考记录 CloudTrail 并记录在 AWS Elemental MediaTailor API 参考中。例如，对 `PutPlaybackConfiguration` 和 `ListPlaybackConfigurations` 操作的调用将在 CloudTrail 日志文件中生成条目。

每个事件或日志条目都包含有关生成请求的人员信息。身份信息有助于您确定以下内容：

- 请求是使用根用户还是 AWS Identity and Access Management (IAM) 凭证发出的
- 请求是使用角色还是联合用户的临时安全凭证发出的
- 请求是否由其他 AWS 服务提出

有关更多信息，请参阅 CloudTrail userIdentity 元素。

了解 AWS Elemental MediaTailor 日志文件条目

跟踪是一种配置，它允许将事件作为日志文件传输到您指定的 Amazon S3 存储桶。 CloudTrail 日志文件包含一个或多个日志条目。一个事件表示来自任何源的一个请求，包括有关所请求的操作、操作的日期和时间、请求参数等方面的信息。 CloudTrail 日志文件不是公用 API 调用的有序堆栈跟踪，因此它们不会以任何特定顺序显示。

以下示例显示了演示`PutPlaybackConfiguration`操作的 CloudTrail 日志条目：

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAEXAMPLE",
```

```
        "arn": "arn:aws:iam::111122223333:user/testuser",
        "accountId": "111122223333",
        "accessKeyId": "AIDAEXAMPLE",
        "userName": "testuser"
    },
    "eventTime": "2018-12-28T22:53:46Z",
    "eventSource": "mediatailor.amazonaws.com",
    "eventName": "PutPlaybackConfiguration",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "1.2.3.4",
    "userAgent": "PostmanRuntime/7.4.0",
    "requestParameters": {
        "VideoContentSourceUrl": "http://examplevideo.com",
        "Name": "examplename",
        "AdDecisionServerUrl": "http://exampleads.com"
    },
    "responseElements": {
        "SessionInitializationEndpointPrefix": "https://777788889999.mediatailor.us-
east-1.amazonaws.com/v1/session/AKIAIOSFODNN7EXAMPLE/examplename/",
        "DashConfiguration": {
            "ManifestEndpointPrefix": "https://777788889999.mediatailor.us-
east-1.amazonaws.com/v1/dash/AKIAIOSFODNN7EXAMPLE/examplename/",
            "MpdLocation": "EMT_DEFAULT"
        },
        "AdDecisionServerUrl": "http://exampleads.com",
        "CdnConfiguration": {},
        "PlaybackEndpointPrefix": "https://777788889999.mediatailor.us-
east-1.amazonaws.com",
        "HlsConfiguration": {
            "ManifestEndpointPrefix": "https://777788889999.mediatailor.us-
east-1.amazonaws.com/v1/master/AKIAIOSFODNN7EXAMPLE/examplename/"
        },
        "VideoContentSourceUrl": "http://examplevideo.com",
        "Name": "examplename"
    },
    "requestID": "1a2b3c4d-1234-5678-1234-1a2b3c4d5e6f",
    "eventID": "987abc65-1a2b-3c4d-5d6e-987abc654def",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```


以下示例显示了演示GetPlaybackConfiguration操作的 CloudTrail 日志条目：

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAEXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/testuser",
        "accountId": "111122223333",
        "accessKeyId": "AIDAEXAMPLE",
        "userName": "testuser"
    },
    "eventTime": "2018-12-28T22:52:37Z",
    "eventSource": "mediatailor.amazonaws.com",
    "eventName": "GetPlaybackConfiguration",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "1.2.3.4",
    "userAgent": "PostmanRuntime/7.4.0",
    "requestParameters": {
        "Name": "examplename"
    },
    "responseElements": null,
    "requestID": "0z1y2x3w-0123-4567-9876-6q7r8s9t0u1v",
    "eventID": "888ddd77-3322-eeww-uuii-abc123jkl343",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```


正在接收 AWS Elemental MediaTailor 频道集合警报

MediaTailor 针对您的频道汇编资源出现的问题或潜在问题创建警报。该警报描述了问题、问题发生的时间以及受影响的资源。

您可以在、 AWS Command Line Interface (AWS CLI) AWS 管理控制台、 AWS 软件开发工具包中查看警报，也可以使用 API 以编程方式查看警报。 MediaTailor ListAlerts

> **Important**
>
> 警报仅适用于 2021 年 7 月 14 日当天或之后创建的频道汇编资源。

渠道集会提醒

警报类型	警报码	警报消息	注意
VOD 来源	NOT_PROCESSED	MediaTailor 尚未处理软件包配置*configurationPath* 。	
	UNREACHABLE	我们无法访问该网址*url*。	
	UNAUTHORIZED	*url*未批准该请求。	
	TIMEOUT	与 *url* 的连接超时。	
	UNPARSABLE_MANIFEST	MediaTailor 在解析来自*url*的清单时遇到问题。	
	VARIANT_DURATION_MISMATCH	MediaTailor 在解析清单时遇到了总持续时间不匹配的变体。*url*这可能会导致播放过程中卡顿。	您的清单的持续时间各不相同。 variants/representations 这可能会导致字幕缺失或不正确，并且 MediaTailor 无法插入广告。
	SEGMENT_DURATION_TOO_LONG	MediaTailor 在解析清单时遇到了持续时间超过三十秒的片段。*url*这可能会导致播放时卡顿、字幕缺失或不正确，以及无法插入广告。	您的清单包含一个超过 30 秒的区段。
	TARGET_DURATION_MISMATCH	MediaTailor 在解析来自的清单时，遇到 HLS 清单中的EXT-X-TARGETDURATION 值不匹配。*url*这可能会导致播放过程中卡顿。	源文件中所有清单的目标持续时间不匹配。
来源位置	NOT_PROCESSED	MediaTailor 尚未处理资源*resourceName* 。	

警报类型	警报码	警报消息	注意
计划	VOD_SOURCE_ALERT	该程序*vodSourceName* 中的 VOD 源有以下警报：*vodSourceAlertCode* *vodSourceAlertMessage*	
	SOURCE_LOCATION_ALERT	该程序中*sourceLocationName* 包含的源位置有以下警报*sourceLocationAlertCode* ：*sourceLocationAlertMessage*	
	CODEC_MISMATCH	MediaTailor 在计划中遇到了不匹配的编解码器。*channelName* 不匹配在清单*manifestUrl* 和*programName2* 清单*sourceGroupName* 之间*programName1* 。*manifestUrl*	
	RESOLUTION_MISMATCH	MediaTailor 在*channelName* 计划中遇到了不匹配的分辨率。不匹配在清单*manifestUrl* 和*programName2* 清单*sourceGroupName* 之间*programName1* 。*manifestUrl*	

警报类型	警报码	警报消息	注意
	BANDWIDTH_MISMATCH	MediaTailor *channelName* 计划中遇到了不匹配的带宽。不匹配在清单*manifestUrl* 和*programName2* 清单*sourceGroupName* 之间*programName1* 。*manifestUrl*	
	FRAMERATE_MISMATCH	MediaTailor 在计划中遇到了不匹配的帧率。*channelName* 不匹配在清单*manifestUrl* 和*programName2* 清单*sourceGroupName* 之间*programName1* 。*manifestUrl*	
	TARGET_DURATION_MISMATCH	MediaTailor 计划中的 HLS 清单中遇到了不匹配的EXT-X-TARGETDURATION 值。*channelName* 不匹配在清单*manifestUrl* 和*programName2* 清单*sourceGroupName* 之间*programName1* 。*manifestUrl*	
	SEGMENT_DURATION_MISMATCH	MediaTailor 在计划清单中*channelName* 遇到了不匹配的区段持续时间值。不匹配在清单*manifestUrl* 和*programName2* 清单*sourceGroupName* 之间*programName1* 。*manifestUrl*	

警报类型	警报码	警报消息	注意
	NO_COMMON_SEGMENT_BOUNDARY_FOR_AD_SLATE	MediaTailor 无法在程序*programName* 的偏移量处插入广告名*offsetMillis* 单。广告列表开始时没有共同的分段边界。	
	NOT_PROCESSED	MediaTailor 尚未处理资源*resourceName* 。	
	TOO_MANY_ALERTS	MediaTailor 已发现太多警报，将不再为提供任何警报*programName* 。清除现有警报以继续接收的警报*programName* 。	
频道	PROGRAM_ALERT	该频道中*programName* 包含的节目有以下警报*programAlertCode* ：*programAlertMessage*	

对与清单相关的警报进行故障排除

当您遇到与清单相关的警报（例如UNPARSABLE_MANIFEST、或）时VARIANT_DURATION_MISMATCHTARGET_DURATION_MISMATCH，请检查以下内容：

1. 验证来源要求：确保您的内容符合中列出的要求the section called “整合内容源”。
2. 查看清单日志：有关详细的清单错误分析和事件类型，请参阅the section called “清单日志”。
3. 检查错误代码：有关详细的错误分析，请参阅问题排查。
4. 查看清单格式：确认您的清单遵循流媒体协议（HLS 或 DASH）的正确格式规范。

查看警报

您可以查看任何 MediaTailor 频道汇编资源的警报。当您查看频道和节目的提醒时， MediaTailor 包括频道或节目中包含的所有相关资源。例如，当您查看特定程序的警报时，您还会看到该程序包含的源位置和 VOD 源的警报。

要查看警报，请执行以下步骤。

Console

在控制台中查看警报

1. 在处打开 MediaTailor 控制台https://console.aws.amazon.com/mediatailor/。
2. 选择要查看其警报的资源。
3. 选择 “警报” 选项卡以查看警报。

AWS Command Line Interface (AWS CLI)

要列出频道集合资源的警报，您需要该资源的亚马逊资源名称 (ARN) 。您可以使用 AWS Command Line Interface (AWS CLI) 中的`describe-`*`resource_type`*命令来获取资源的 ARN。例如，运行 describe-channel 命令来获取特定频道的 ARN：

```
aws mediatailor describe-channel --channel-name MyChannelName
```

然后使用 a ws mediatailor list-alerts 命令列出与资源相关的警报：

```
aws mediatailor list-alerts --resource-arn arn:aws:mediatailor:region:aws-account-
id:resource-type/resource-name
```

API

要列出频道集合资源的警报，您需要该资源的亚马逊资源名称 (ARN) 。您可以使用 MediaTailor API 中的`Describe`*`Resource`*操作来获取资源的 ARN。例如，使用DescribeChannel操作获取特定频道的 ARN。

然后使用 ListAlerts API 列出资源警报。

处理警报

出现警报时，在中查看警报 AWS 管理控制台，或使用 AWS Command Line Interface (AWS CLI)、AWS SDK 或 MediaTailor Alerts API 来确定问题的可能来源。

解决问题后， MediaTailor 清除警报。

标签 AWS Elemental MediaTailor 资源

标签是您分配或分配给 AWS 资源的 AWS 元数据标签。每个标签均包含一个键 和一个值。对于您分配的标签，需要定义键和值。例如，您可以将键定义为 `stage`，将一个资源的值定义为 `test`。

标签可帮助您：

- 识别和整理您的 AWS 资源。许多 AWS 服务都支持标记，因此您可以为来自不同服务的资源分配相同的标签，以表明这些资源是相关的。例如，您可以为 AWS Elemental MediaPackage 频道和终端节点分配与分配给 AWS Elemental MediaTailor 配置相同的标签。
- 追踪您的 AWS 成本。您在 AWS 账单与成本管理 仪表板上激活这些标签。 AWS 使用标签对您的成本进行分类，并向您提供每月成本分配报告。有关更多信息，请参阅AWS Billing 用户指南中的使用成本分配标签。
- 控制对 AWS 资源的访问权限。有关更多信息，请参阅《IAM 用户指南》https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html中的使用标签控制访问。

以下部分提供了有关标签的更多信息 AWS Elemental MediaTailor。

中支持的资源 AWS Elemental MediaTailor

中的以下资源 AWS Elemental MediaTailor 支持标记：

- 渠道
- 配置
- SourceLocations
- VodSources

标签限制

以下基本限制适用于 AWS Elemental MediaTailor 资源上的标签：

- 您可以分配给资源的最大标签数量 - 50
- 最大密钥长度 – 128 个 Unicode 字符
- 最大值长度 – 256 个 Unicode 字符
- 键和值的有效字符 — a-z A-Z、0-9、空格和以下字符：_.:/= +-和 @
- 密钥和值区分大小写
- 不要用aws:作密钥的前缀；它是留给 AWS 使用的

在中管理标签 AWS Elemental MediaTailor

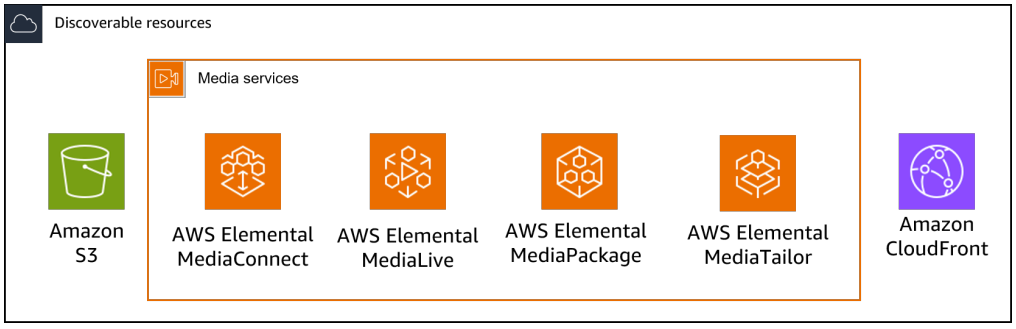
您可在资源上将标签设置为属性。您可以通过 AWS Elemental MediaTailor API 或 AWS Command Line Interface (AWS CLI) 添加、编辑和删除标签。有关更多信息，请参阅 AWS Elemental MediaTailor API 参考。

监控 AWS 带工作流监视器的媒体服务

工作流监控器是用于发现、可视化和监控 AWS 媒体工作流的工具。工作流监控器可在 AWS 控制台和 API 中使用。您可以使用工作流监视器来发现和创建工作流资源的可视化映射，即信号地图。您可以创建和管理亚马逊 CloudWatch 警报和亚马逊 EventBridge规则模板来监控映射的资源。您创建的监控模板将转换为可部署的 AWS CloudFormation 模板，以实现可重复性。 AWS 推荐的警报模板提供预定义的最佳实践监控。

发现

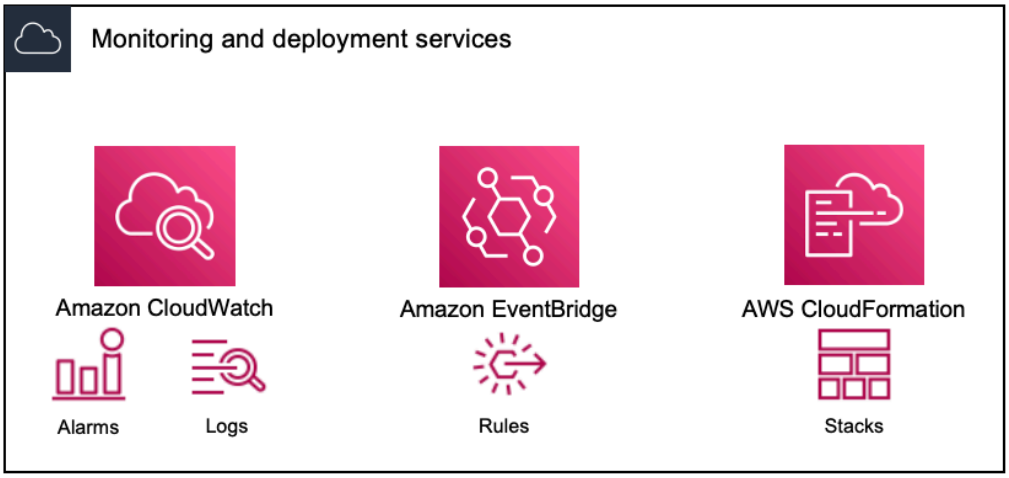
利用信号映射自动发现与您的媒体工作流程相关的互联 AWS 资源。发现可以从任何支持的服务资源开始，并创建工作流的端到端映射。信号地图可用作独立的可视化工具，也可以使用监控模板进行增强。



监控

您可以创建自定义 CloudWatch 警报和 EventBridge 规则模板来监控媒体工作流程的运行状况和状态。可以将最佳实践警报模板导入到您的工作流监视器环境中。您可以按原样使用最佳实践警报模

板，也可以对其进行编辑以更好地适应您的工作流。您创建的任何模板都将转换为可重复部署的 CloudFormation 模板。



> **Note**
>
> 使用工作流监视器没有直接成本。但是，创建和用于监控工作流的资源会产生相关费用。部署监控时，会创建亚马逊 CloudWatch 和亚马逊 EventBridge 资源。使用 AWS 管理控制台时，在将监控部署到信号图之前，系统会通知您将创建多少资源。有关定价的更多信息，请参阅：CloudWatch定价和EventBridge 定价。
>
> 工作流监控器使用 AWS CloudFormation 模板来部署 CloudWatch 和 EventBridge 资源。这些模板存储在由工作流监视器在部署过程中为您创建的标准类 Amazon Simple Storage Service 存储桶中，并将产生对象存储和召回费用。有关定价的更多信息，请参阅 Amazon S3 定价。
>
> 在工作流监视器信号图中生成的信 AWS Elemental MediaPackage 道预览从 MediaPackage Origin 端点传送，将产生数据传出费用。有关定价，请参阅：MediaPackage定价。

工作流监视器的组件

工作流监视器有以下四个主要组件：

- CloudWatch 警报模板-定义您要使用的监控条件 CloudWatch。您可以创建自己的警报模板，也可以导入由创建的预定义模板 AWS。有关更多信息，请参阅：CloudWatch 警报组和模板，用于监控您的 AWS 媒体工作流程
- EventBridge 规则模板-定义触发警报时如何 EventBridge 发送通知。有关更多信息，请参阅：EventBridge 用于监控您的规则组和模板 AWS 媒体工作流程
- 信号地图-使用自动化流程使用现有 AWS 资源创建 AWS Elemental 工作流程图。信号地图可用于发现工作流中的资源，并对这些资源部署监控。有关更多信息，请参阅：工作流监视器信号地图。

- 概览 - 概览页面使您可以从一个位置直接监控多个信号地图的状态。查看工作流的指标、日志和警报。有关更多信息，请参阅：工作流监视器概览。

受支持的服务

工作流监视器支持自动发现与以下服务相关的资源，并进行信号映射：

- AWS Elemental MediaConnect
- AWS Elemental MediaLive
- AWS Elemental MediaPackage
- AWS Elemental MediaTailor
- Amazon S3
- 亚马逊 CloudFront

主题

- 将工作流监视器配置为监控 AWS 媒体服务
- 使用工作流监视器

将工作流监视器配置为监控 AWS 媒体服务

首次设置工作流监视器时，您需要创建警报和事件模板，并发现用于监控媒体工作流的信号地图。以下指南包含设置管理员和操作员级别 IAM 角色、创建工作流监视器资源以及将监控部署到工作流所需的步骤。

主题

- 工作流监视器入门
- 工作流监视器组和模板
- 工作流监视器信号地图
- 工作流监视器配额

工作流监视器入门

以下步骤提供了首次使用工作流监视器的基本概述。

1. 为管理员和操作员级别角色设置工作流监视器 IAM 权限：工作流监视器 IAM 策略

2. 生成警报模板或导入由 AWS以下人员创建的预定义模板：CloudWatch 警报
3. 生成通知事件，这些事件将通过 EventBridge以下方式交付：EventBridge 规则
4. 使用您现有的 AWS 元素资源探索信号地图：信号地图
5. 将警报模板和通知规则附加到您的信号地图上：附加模板
6. 部署模板以开始监控信号地图：部署监控模板
7. 使用 AWS 控制台的概览部分，监控和查看您的工作流监视器资源：概述

工作流监视器 IAM 策略

工作流监视器与多个 AWS 服务进行交互以创建信号地图、构建 CloudWatch 和 EventBridge 资源以及 CloudFormation 模板。由于工作流监控器与各种服务交互，因此必须为这些服务分配特定 AWS Identity and Access Management (IAM) 策略。以下示例说明了管理员和操作员 IAM 角色所必需的 IAM 策略。

管理员 IAM 策略

以下示例策略适用于管理员级别的工作流监视器 IAM 策略。通过此角色可以创建和管理工作流监视器资源，以及与工作流监视器交互的受支持服务资源。

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:ListGroups",
        "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": "*"
    }
  ]
}
```

操作员 IAM 策略

以下示例策略适用于操作员级别的工作流监视器 IAM 策略。此角色可以对工作流监视器资源以及与之交互的受支持服务资源进行有限的只读访问。

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": "*"
    }
  ]
}
```

工作流监视器组和模板

在将工作流监控部署到信号图之前，必须为 CloudWatch 警报和 EventBridge 通知创建组和模板。CloudWatch 模板定义了将使用哪些场景和阈值来触发警报。 EventBridge 模板将决定如何向您报告这些警报。

如果您只想映射连接的资源，而不想使用工作流监视器的监控模板功能，则可以在没有 CloudWatch 和 EventBridge 模板的情况下使用信号映射。有关使用信号地图的更多信息，请参阅：信号地图 。

主题

- CloudWatch 警报组和模板，用于监控您的 AWS 媒体工作流程
- EventBridge 用于监控您的规则组和模板 AWS 媒体工作流程

CloudWatch 警报组和模板，用于监控您的 AWS 媒体工作流程

工作流监控器警报允许您使用现有 CloudWatch 指标作为信号图警报的基础。您可以创建一个警报模板组，以便对您工作流中至关重要的警报类型进行排序和分类。在每个警报模板组中，您可以创建包含要监控的特定 CloudWatch 指标和参数的警报模板。您可以创建自己的警报模板或导入由创建的推荐警报模板 AWS。在创建警报模板组和该组中的警报模板后，您可以将其中一个或多个警报模板组附加到信号地图。

您必须先创建警报模板组。创建警报模板组后，您可以创建自己的模板或使用由创建的推荐模板 AWS。如果您想创建自己的警报模板，请继续浏览本页面。有关导入推荐模板的更多信息，请参阅：推荐的模板

本节介绍使用工作流监控器创建 CloudWatch 警报。有关该 CloudWatch 服务如何处理警报的更多信息以及警报组件的详细信息，请参阅《亚马逊 CloudWatch 用户指南》中的使用 CloudWatch 警报

创建警报模板组

通过警报模板组，您可以对工作流中重要的警报类型进行排序和分类。

创建警报模板组

1. 在工作流监控控制台的导航窗格中，选择CloudWatch警报模板。
2. 选择创建警报模板组。
3. 为警报模板组指定一个唯一的组名和可选的描述。
4. 选择创建，系统将转到新建的警报模板组的详细信息页面。

创建警报模板

您可以使用要监控的 CloudWatch 指标和参数创建警报模板。

创建警报模板

1. 在警报模板组的详细信息页面中，选择创建警报模板。
2. 为警报模板指定一个唯一的模板名称和可选的描述。
3. 在选择指标部分：
    1. 选择目标资源类型。目标资源类型是相应服务的资源，例如渠道 MediaLive 和/ MediaPackage 或流量 MediaConnect。

2. 选择指标名称。这是作为警报基础的 CloudWatch指标。指标列表将根据所选的目标资源类型而变化。

4. 在警报设置部分：

> **Note**
>
> 有关该 CloudWatch 服务如何处理警报的更多信息以及警报组件的详细信息，请参阅《亚马逊 CloudWatch 用户指南》中的使用 CloudWatch 警报

1. 选择统计数据。这是一个将用于监控指标的值，例如总和或平均值。
2. 选择比较运算符。此字段引用您在下一步中设置的阈值。
3. 设置阈值。这是比较运算符用来确定大于、小于还是等于状态的数值。
4. 设置周期。这是一个以秒为单位的时间值。周期是统计数据、比较运算符和阈值相互作用，以确定是否触发警报的时间长度。
5. 设置数据点。此值确定需要多少数据点才能触发警报。
6. 选择如何处理丢失的数据。该选项确定了此警报对丢失的数据的反应方式。

5. 选择创建以完成该过程。

已完成的警报模板示例可能具有以下参数：监控 MediaConnect 流量目标资源类型的断开连接指标名称。统计数据值设置为“总和”，比较运算符为“大于或等于”且阈值为 10。周期设置为 60 秒，只需要 1 个数据点中的 1 个。将处理丢失的数据设置为“忽略”。

这些设置的结果是：工作流监视器将监控流上的断开连接情况。如果 60 秒内发生 10 次或更多次断开连接，则会触发警报。60 秒内发生 10 次或更多断开连接的情况只需出现一次，即可触发警报。

推荐的警报模板，用于监控您的 AWS 媒体工作流程

工作流监控器的推荐模板是精选的 AWS Elemental 服务指标，其中包含适用于该指标的预定义警报设置。如果您不想创建自定义警报模板，推荐模板会为您提供由 AWS创建的最佳实践监控模板。

工作流监视器包含每种受支持服务的推荐模板组。这些组旨在将最佳实践监控应用于特定类型的工作流。每个模板组都包含根据服务特定指标配置的精选警报。例如，为 MediaLive 多路工作流程推荐的模板组将具有与 MediaConnect CDI 工作流程不同的预配置指标集。

使用推荐的警报模板

1. 按照步骤创建警报模板组，或者选择现有的模板组。
2. 在警报模板部分，选择导入。您需要将 AWS 推荐的模板导入到您的模板组中。
3. 使用CloudWatch 警报模板组下拉列表选择 AWS 推荐的群组。这些组包含针对特定服务精选的警报。
4. 使用复选框选择要导入的模板。每个模板都将列出其指标、预配置的监控值并提供指标描述。选择完模板后，选择添加按钮。
5. 所选模板将移至要导入的警报模板部分。查看您的选择，然后选择导入。
6. 导入完成后，所选模板将添加到模板组中。如果要添加更多模板，请重复导入过程。
7. 导入的模板可以在导入后进行自定义。可以修改警报设置以满足您的警报需求。

EventBridge 用于监控您的规则组和模板 AWS 媒体工作流程

CloudWatch 使用亚马逊 EventBridge 规则发送通知。首先要创建一个事件模板组。在该事件模板组中，您可以创建事件模板，以确定哪些条件创建通知以及通知的对象。

本节介绍使用工作流监控器创建 EventBridge 规则。有关该 EventBridge 服务如何使用规则的更多信息，请参阅：亚马逊 EventBridge 用户指南中的EventBridge 规则

创建事件模板组

事件模板组允许您根据自己的应用场景对事件进行排序和分类。

创建事件模板组

1. 在工作流监控控制台的导航窗格中，选择EventBridge 规则模板。
2. 选择创建活动模板组。
3. 为警报模板组指定一个唯一的组名和可选的描述。
4. 选择创建，系统将转到新建的警报模板组的详细信息页面。

创建事件模板

您可以根据自己创建的事件模板发送通知。

创建事件模板

1. 在事件模板组的详细信息页面中，选择创建事件模板。
2. 为事件模板指定一个唯一的模板名称和可选的描述。
3. 在规则设置部分：
    1. 选择事件类型。选择事件类型时，可以在由创建的多个事件之间进行选择， AWS 也可以选择 Signal map 活动警报以使用警报模板创建的警报。
    2. 选择目标服务。这决定了您希望以何种方式收到此事件的通知。您可以选择亚马逊简单通知服务或 CloudWatch 日志。
    3. 选择目标服务后，选择一个目标。这将是 Amazon SNS 主题或 CloudWatch 日志组，具体取决于您的目标服务选择。
4. 选择创建以完成该过程。

工作流监视器信号地图

信号地图是媒体工作流程中 AWS 资源的可视化映射。您可以使用工作流监视器在任何支持的资源类型上启动信号地图发现。在发现过程中，工作流监控器将自动递归地映射所有连接的 AWS 资源。信号地图创建完成后，您可以使用工作流监视器控制台执行诸如部署监控模板、查看指标和查看映射资源的详细信息之类的操作。

主题

- 为以下对象创建信号图 AWS 媒体工作流程
- 查看信号图 AWS 媒体工作流程
- 将警报和事件模板附加到您的信号图中 AWS 媒体工作流程
- 将模板部署到您的信号图 AWS 媒体工作流程
- 更新您的信号图 AWS 媒体工作流程
- 删除您的信号地图 AWS 媒体工作流程

为以下对象创建信号图 AWS 媒体工作流程

您可以使用工作流监视器信号映射来创建媒体工作流程中所有关联 AWS 资源的可视化映射。

创建信号地图

1. 在工作流监视器控制台的导航窗格中，选择信号地图。

2. 选择创建信号地图。
3. 为信号地图指定名称和描述。
4. 在发现新信号地图部分中，将显示当前账户和选定区域中的资源。选择资源以开始发现信号地图。所选的资源将是发现的起点。
5. 选择创建。等待发现过程完成。这个过程完成后，将显示新的信号地图。

> **Note**
>
> 在工作流监视器信号图中生成的信 AWS Elemental MediaPackage 道预览从 MediaPackage Origin 端点传送，将产生数据传出费用。有关定价，请参阅：MediaPackage定价。

查看信号图 AWS 媒体工作流程

工作流监视器信号图允许您查看媒体工作流程中所有连接 AWS 资源的可视化映射。

信号地图视图

选择信号地图后，您有两个视图可用于监控或配置信号地图。监控信号地图和配置信号地图是一个上下文相关按钮，位于信号地图控制台部分的右上角。

如果您使用导航窗格的信号地图部分选择信号地图，则您的信号地图将显示在配置视图中。通过配置视图，您可以更改附加到此信号地图的模板组、部署附加的模板，以及查看信号地图的基本详细信息和标记。

如果您使用导航窗格的概览部分选择信号地图，则您的信号地图将显示在监控视图中。监控视图显示该 CloudWatch 信号图的警报、 EventBridge 规则、警报、日志和指标。

通过选择右上角的Monitor/Configure 信号映射按钮，可以随时更改视图。配置视图需要管理员级别的 IAM 权限。所需的 IAM 权限可以在以下位置查看：工作流监视器 IAM 策略

浏览信号地图

信号地图将包含工作流监视器发现的每个受支持 AWS 资源的节点。如果缩略图预览可用，某些资源（例如 MediaLive 频道和 MediaPackage终端节点）可以显示内容的缩略图预览。

选择资源节点，然后从操作下拉菜单中选择查看所选资源的详细信息，系统将转入该关联服务的详细信息页面。例如，选择一个 MediaLive 频道并选择查看所选资源详细信息将打开该频道 MediaLive 的主机详细信息页面。

选择一个资源节点将筛选活动警报列表，仅显示该节点的警报。如果您在活动警报中选择资源的目标 ARN，系统会转到关联服务的详细信息页面，并打开所选资源。

将警报和事件模板附加到您的信号图中 AWS 媒体工作流程

创建警报和事件模板后，需要将它们附加到信号地图。您创建的任何警报和事件模板都可以附加到任何已发现的信号地图上。

将警报和事件模板附加到信号地图

1. 在工作流监视器控制台的导航窗格中，选择信号地图，然后选择要使用的信号地图。
2. 在信号映射页面右上角的CloudWatch警报模板组选项卡中，选择附加 CloudWatch 警报模板组。
    1. 在打开的新部分中，选择要应用于此信号地图的所有警报模板组，然后选择添加。这将导致所选警报模板组移至 “附加 CloudWatch警报模板组” 部分。
    2. 选择保存将保存您的更改，并返回到信号地图页面。
3. 在信号映射页面的右侧，选择EventBridge 规则模板组选项卡，然后选择附加 EventBridge规则模板组。
    1. 在打开的新部分中，选择要应用于此信号地图的所有事件模板组，然后选择添加。这将导致选定的规则模板组移至 “附加的 EventBridge 规则模板组” 部分。
    2. 选择保存将保存您的更改，并返回到信号地图页面。
4. 您已经为信号图分配了 CloudWatch 警报和 EventBridge 规则模板，但监控尚未部署。下一节将介绍监控资源的部署。

将模板部署到您的信号图 AWS 媒体工作流程

在将警报和事件模板附加到信号地图后，必须部署监控。在部署完成之前，您信号地图的监控将不会启用。

工作流监视器将仅部署与所选信号地图相关的警报。例如，附加的警报模板组可能包含多个服务的警报 MediaLive，例如 MediaPackage、和 MediaConnect。如果所选信号图仅包含 MediaLive 资源，则不会部署 MediaPackage 任何 MediaConnect 警报。

部署监控模板

1. 将警报和事件模板组附加到信号地图并保存更改后，在操作下拉菜单中选择部署监控器。
2. 系统将要求您确认部署，并提供将要创建的数量 CloudWatch 和 EventBridge 资源。如果要继续，请选择部署。

> **Note**
>
> 使用工作流监视器没有直接成本。但是，创建和用于监控工作流的资源会产生相关费用。部署监控时，会创建亚马逊 CloudWatch 和亚马逊 EventBridge 资源。使用 AWS 管理控制台时，在将监控部署到信号图之前，系统会通知您将创建多少资源。有关定价的更多信息，请参阅：CloudWatch定价和EventBridge 定价。
>
> 工作流监控器使用 AWS CloudFormation 模板来部署 CloudWatch 和 EventBridge 资源。这些模板存储在由工作流监视器在部署过程中为您创建的标准类 Amazon Simple Storage Service 存储桶中，并将产生对象存储和召回费用。有关定价的更多信息，请参阅 Amazon S3 定价。

3. 部署状态显示在信号地图名称旁边。部署状态也可以在 CloudFormation 控制台的 “堆栈” 部分中看到。在资源创建和部署几分钟之后，您的信号地图监控将开始。

更新您的信号图 AWS 媒体工作流程

如果对工作流进行了更改，则可能需要重新发现信号地图并重新部署监控资源。工作流监视器是一种可视化和监控工具，无法对您的工作流进行任何更改。信号地图代表您工作流的时间点可视化图像。如果您添加、移除或大幅修改媒体工作流的某些部分，我们建议您重新发现信号地图。如果您在信号地图上附加了监控资源，我们建议您在重新发现过程结束后重新部署监控。

重新发现信号地图

1. 在工作流监视器控制台的导航窗格中，选择信号地图，然后选择要使用的信号地图。
2. 确认您处于配置信号地图视图中。有关更改视图的更多信息，请参阅：查看信号地图 。
3. 在信号地图页面的右上角，选择操作下拉菜单。选择重新发现。
4. 您将看到重新发现屏幕。选择作为您要重新发现的工作流一部分的资源。选择重新发现按钮。
5. 信号地图将根据当前工作流重新构建。如果您需要重新部署监控资源，请继续留在此信号地图页面。之前附加的所有监控模板都将保持附加状态，但需要重新部署。

在重新发现信号地图后重新部署监控模板

1. 重新发现后，您将被引导到更新的信号地图。要重新部署监控模板，请从操作下拉菜单中选择部署监控器。
2. 系统将要求您确认部署，并向您提供将要创建的任一 CloudWatch 和 EventBridge 资源的数量。如果要继续，请选择部署。

3. 部署状态显示在信号地图名称旁边。在资源创建和部署几分钟之后，您的信号地图监控将开始。

删除您的信号地图 AWS 媒体工作流程

如果您不再需要某个信号地图，可以将其删除。如果您在信号图上部署了监控模板，则删除过程将要求您删除已部署到该信号图的所有 CloudWatch 和 EventBridge 资源。删除已部署的资源不会影响创建这些资源的模板。此资源删除是为了确保您没有已 CloudWatch 部署但未使用的 EventBridge 资源。

删除信号地图

1. 从工作流监视器控制台的导航窗格中，选择信号地图，然后选中要删除的信号地图旁边的单选按钮。
2. 选择删除按钮。系统将要求您确认删除监控资源：选择删除以开始监控资源删除过程。
3. 监控部署列将显示当前状态。当状态更改为 DELETE_COMPLETE 时，再次选择删除按钮。
4. 系统将要求您确认删除信号地图。选择删除以继续并删除信号地图。

工作流监视器配额

下一节包含工作流监控资源的配额。每个配额均基于“每个账户”进行计算。如果您需要增加账户配额，则可以使用AWS 服务配额控制台申请增加配额，除非下表中另有说明。

配额

资源类型	配额
CloudWatch 警报模板组	20
CloudWatch 警报模板	200
EventBridge 规则模板组	20
EventBridge 规则模板	200
信号地图	30
信号地图：附加到单个信号图的 CloudWatch 警报模板组	5 您不能增加此限额。

资源类型	配额
信号映射：附加到单个信号图的 EventBridge 规则模板组	5 您不能增加此限额。

使用工作流监视器

使用工作流监视器控制台的概览和信号地图部分，可查看工作流的当前状态，以及任何相关的警报、指标和日志。

主题

- 工作流监视器概览
- 工作流监视器的概览日志和指标
- 使用工作流监视器信号地图

工作流监视器概览

工作流监视器控制台的概览部分是一个仪表板，可提供有关信号地图的概览信息。在概述部分，您可以看到每个信号图监控的当前状态，以及 CloudWatch 指标和任何相关 CloudWatch 日志。您可以选择任何信号地图，以跳转到该信号地图控制台页面。

概览筛选

使用概览部分中的搜索栏，您可以使用上下文相关约束条件来筛选信号地图列表。选择搜索栏后，将显示一个可作为筛选依据的属性列表。选择一个属性将显示诸如“等于”、“包含”、“不等于”和“不包含”等运算符。选择运算符将根据所选属性类型创建资源列表。选择其中一个资源将导致信号地图列表仅显示符合您定义的约束条件的信号地图。

工作流监视器的概览日志和指标

要查看信号图的 CloudWatch 指标和日志，请选择信号图名称旁边的单选按钮。信号地图列表下方将同时显示指标和日志的选项卡式界面。

CloudWatch Metrics

CloudWatch 所选信号图的指标将与上下文相关，仅显示与该信号映射工作流程中使用的服务相关的指标。您可以使用屏幕上的指标工具来自定义显示的指标周期和时间范围。

CloudWatch 日志

如果您将 CloudWatch 日志组与信号映射相关联，则该组将显示在此处。

使用工作流监视器信号地图

在控制台的概览部分，您可以选择特定的信号地图以查看有关该信号地图及其附加监控资源的更多信息。

选择信号地图后，您将看到信号地图和一些包含更多信息的选项卡式部分：

- CloudWatch 警报
- EventBridge 规则
- AWS 元素警报
- 指标
- 日志
- 基本细节

浏览信号地图

信号地图将包含工作流监视器发现的每个受支持 AWS 资源的节点。如果缩略图预览可用，某些资源（例如 MediaLive 频道和 MediaPackage 终端节点）可以显示内容的缩略图预览。

选择资源节点，然后从操作下拉菜单中选择查看所选资源的详细信息，系统将转入该关联服务的详细信息页面。例如，选择一个 MediaLive 频道并选择查看所选资源详细信息将打开该频道 MediaLive 的主机详细信息页面。

选择一个资源节点将筛选活动警报列表，仅显示该节点的警报。如果您在活动警报中选择资源的目标 ARN，系统会转到关联服务的详细信息页面，并打开所选资源。

Dual-stack (IPv6) 的终端节点 AWS Elemental MediaTailor

AWS Elemental MediaTailor 支持双栈端点，您可以使用这些端点通过 IPv4 或 IPv6 进行连接。这些端点在功能上等同于标准 IPv4-only 端点。服务行为相同，只有网络传输层不同。

什么是双栈终端节点？

Dual-stack 端点同时通告 IPv4 和 IPv6 地址。当您使用双栈终端节点时，DNS 会根据您的网络配置和客户端首选项解析为 IPv4 或 IPv6 地址。您无需更改 MediaTailor 资源、播放配置或广告插入行为。

支持的功能

下表列出了对双栈端点的 MediaTailor 功能支持。

功能	Dual-stack 支持
控制平面 API（所有区域）	支持
SSAI 数据平面（会话初始化、清单、区段）	支持
信道集合数据平面	支持
AWS 管理控制台	支持
AWS SDK	即将推出

Dual-stack 端点格式

MediaTailor 双栈端点使用以下`.api.aws`域：

- 控制平面：`mediatailor.`*`region`*`.api.aws`
- SSAI 数据平面：显示在播放配置详细信息的 Dualstack 端点部分中。 AWS 管理控制台
- 通道集合数据平面：显示在频道的输出详细信息中 AWS 管理控制台。

您的现有应用程序将继续使用原始 IPv4-only 端点
（`api.mediatailor.`*`region`*`.amazonaws.com`控制平面、*`unique-`*

id.mediatailor.*region*.amazonaws.com数据平面）。有关可用终端节点的完整列表，请参阅《AWS 一般参考》中的MediaTailor终端节点和配额。

使用双堆栈端点

AWS 管理控制台

Dual-stack 端点显示在 Dualstack 端点部分下的播放配置详细信息页面中。您可以直接复制这些 URL 以用于您的播放器或 CDN 配置。

AWS Command Line Interface

在调用 API 时指定双栈终端节点 URL：

```
aws mediatailor get-playback-configuration \
    --name my-config \
    --endpoint-url https://mediatailor.us-east-1.api.aws
```

*my-config*替换为播放配置的*us-east-1*名称和您的 AWS 区域。

AWS SDK

在 SDK 配置中启用双堆栈。

Python (boto3)：

```
import boto3

client = boto3.client(
    'mediatailor',
    endpoint_url='https://mediatailor.us-east-1.api.aws'
)
```

JavaScript (AWS SDK v3)：

```
import { MediaTailorClient } from "@aws-sdk/client-mediatailor";

const client = new MediaTailorClient({
    useDualstackEndpoint: true
});
```

注意事项

使用双堆栈终端节点时，请记住以下几点：

- Dual-stack 端点在所有可用 AWS 区域 的地方都可 MediaTailor 用。
- 您无需更改播放配置、广告决策服务器设置或 CDN 配置即可使用双栈端点。
- 如果您的 CDN 使用基于主机名模式的路由，请验证您的规则是否符合域格式。`.api.aws`
- 双堆栈端点的标准 AWS 指南适用。有关更多信息，请参阅Dual-stack 《AWS 一般参考》中的 FIPS 端点。

中的配额 AWS Elemental MediaTailor

MediaTailor 资源和操作请求受以下配额（以前称为 “限制”）的约束。

您可以使用 AWS服务配额服务来查看配额并请求增加配额以及许多其他 AWS 服务的配额。MediaTailor有关更多信息，请参阅《Service Quotas 用户指南》。

广告插播配额

下表描述了 AWS Elemental MediaTailor 广告插入的配额。除非另有说明，否则配额不可调整。

Name	默认配额值	说明
广告决策服务器 (ADS) 长度	25000	广告决策服务器 (ADS) 规范中字符的最大数量。
广告决策服务器 (ADS) 重定向	7	VAST 包装器标签中 MediaTailor 遵循的最大重定向深度。MediaTailor 如果还有其他重定向，则放弃。
广告决策服务器 (ADS) 超时	3 秒（默认）	与广告决策服务器 (ADS) 的开放连接超时之前 MediaTailor 等待的最大秒数。当由于广告没有回应而导致连接超时 MediaTailor 时，无法在广告中填充广告。此超时可通过个人 ADS 请求超时设置（范围：250—6,000 毫秒）进行配置。有关更多信息，请参阅 高级设置。

Name	默认配额值	说明	
广告插入请求	10000	执行服务器端广告插入时每秒对个性化清单的最大请求数。广告插入处理传入的清单、会话初始化、跟踪数据和广告段落的请求。此配额可调整。	
配置	1000	MediaTailor 允许的最大配置数量。	
内容源长度	512	内容源规范中字符的最大数量。	
内容源服务器超时	2	请求模板清单时，MediaTailor 与内容源服务器的开放连接超时之前等待的最大秒数。超时生成 HTTP 504 (GatewayTimeoutException) 响应错误。	
清单大小	2	任何源播放清单的最大大小，以 MB 为单位。要确保您不超出配额，请使用 gzip 将您的输入清单压缩到 MediaTailor 中。	
程序包配置	5	每个源的最大套餐配置数量（无论是直播还是视频点播）。	

Name	默认配额值	说明	
预取时间表	25	每个回放配置的最大活跃预取计划数。过期的预取计划不计入此限制。	
服务器端报告信标请求超时	3 秒	触发服务器端报告的信标时，在打开的服务器连接超时之前 MediaTailor 等待的最大秒数。当连接超时 MediaTailor 时，无法触发信标，服务会在MediaTailor/AdDecisionServerInteractions登录时记录一条`ERROR_FIRING_BEACON_FAILED` 消息CloudWatch。	
PRE_SESSION_初始化挂钩超	2 秒	执行 PRE_SESSION_INIALIZATION_INIALIZATION 挂钩所花费的最大秒数，包括所有函数步骤和 HTTP 调用。如果挂接超过其超时时间，则 MediaTailor 丢弃所有输出并继续执行，就好像没有执行任何函数一样。	

Name	默认配额值	说明	
PRE_ADS_REQUEST 挂钩超时	2 秒	执行 PRE_ADS_REQUEST 挂钩所花费的最大秒数，包括所有函数步骤和 HTTP 调用。如果挂接超过其超时时间，则 MediaTailor 丢弃所有输出并继续执行，就好像没有执行任何函数一样。	
每个挂钩的最大功能数	20	每个挂接允许的最大函数总执行次数。	
会话过期	10 倍清单持续时间	MediaTailor 允许会话在结束会话之前保持非活动状态的最大时间。会话活动可以是播放器请求或源服务器的提前。当会话到期时，会 MediaTailor 返回HTTP 400 (Bad Request)响应错误。	

渠道组装配额

下表描述了 AWS Elemental MediaTailor 频道集成的配额。除非另有说明，否则配额是可调整的。

Name	默认配额值	说明	
每个账户的频道清单请求	400	一个账户中所有渠道集合频道每秒的最大出口清单请求数。	

Name	默认配额值	说明	
每个频道的频道清单请求	50	任何单个通道汇编通道每秒的最大出口清单请求数。	
通道输出	5	每个通道的最大输出数。	
每个账户的通道数	100	每个账户的最大通道数。	
直播来源	50	源位置的最大实时源数量。	
每个频道的节目	400	每个频道的最大节目数。	
区段交付配置	5	每个源位置的最大分段交付配置数。	
来源地点	50	每个账户的最大源位置数。	
VOD 源	1000	源位置的最大视频点播 (VOD) 源数量。	

下表描述了 AWS Elemental MediaTailor 信道组装的节流限制。除非另有说明，否则配额是可调整的。

Name	默认每秒交易量上限	说明	
ConfigureLogsForChannel	1	为频道配置日志。	
CreateChannel	1	创建频道。	
CreateLiveSource	1	创建直播源。	

Name	默认每秒交易量上限	说明	
CreateProgram	3	创建程序。	
CreateSourceLocation	1	创建来源位置。	
CreateVodSource	1	创建 VOD 源。	
DeleteChannel	1	删除频道。	
DeleteChannelPolicy	1	删除频道政策。	
DeleteLiveSource	1	删除直播源。	
DeleteProgram	3	删除程序。	
DeleteSourceLocation	1	删除来源位置。	
DeleteVodSource	1	删除 VOD 源。	
DescribeChannel	5	描述一个频道。	
DescribeLiveSoure	5	描述直播来源。	
DescribeProgram	5	描述一个程序。	
DescribeSourceLocation	5	描述来源位置。	
DescribeVodSource	5	描述 VOD 来源。	
GetChannelPolicy	5	获取渠道政策。	
GetChannelSchedule	5	获取频道时间表。	
ListAlerts	5	列出警报。	
ListChannels	5	列出频道。	
ListLiveSources	5	列出实时来源。	
ListPrograms	5	列出程序。	

Name	默认每秒交易量上限	说明	
ListSourceLocations	5	列出来源位置。	
ListTagsForResource	5	列出资源的标签。	
ListVodSources	5	列出 VOD 来源。	
PutChannelPolicy	3	制定渠道政策。	
StartChannel	1	启动一个频道。	
StopChannel	1	停止一个频道。	
TagResource	1	标记资源。	
UntagResource	1	取消标记资源。	
UpdateChannel	1	更新频道。	
UpdateLiveSource	1	更新实时来源。	
UpdateProgram	1	更新程序。	
UpdateSourceLocation	1	更新来源位置。	
UpdateVodSource	1	更新 VOD 来源。	

故障排除 MediaTailor

如果您在使用时遇到播放错误或类似困难 AWS Elemental MediaTailor，请查阅本节中的主题。

主题

- 对 MediaTailor 事件流问题进行故障排除
- 从播放问题进行故障排除 MediaTailor
- 信标重复数据删除
- 参数疑难解答
- MediaTailor 跳过广告疑难解答指南

对 MediaTailor 事件流问题进行故障排除

了解 AWS Elemental MediaTailor 事件流为解决广告插入问题提供了坚实的基础。通过分析事件的顺序、时间和模式，您可以快速确定问题发生的位置并实施有针对性的解决方案。

本节为使用事件流分析诊断问题提供实用指导。要了解基本的事件流概念，请参阅广告插入事件流程。

识别不完整的事件流

当预期的事件序列在成功实现清单个性化（向清单中 MediaTailor 插入个性化广告信息的过程）之前停止时，就会出现不完整的事件流。确定流程中断的地方有助于查明广告插入失败的根本原因。

常见的不完整流程模式

事件流中的不同故障点表示特定类型的问题，例如以下内容。

- 检测到广告机会后流量停止：表示广告标记或清单本身存在问题，导致无法发 MediaTailor 出 ADS 请求。在提出 ADS 请求后，会出现 ADS 连接、配置或超时问题。
- ADS 请求后流量停止：建议 ADS 响应问题、VAST 解析问题、创意处理故障、ADS 超时、连接错误或配置问题，例如仅在发出请求时才会发现的无效 ADS 网址。
- 缺少跟踪信标：可能表示跟踪配置问题、服务器端报告问题或客户端实现差距。

CloudWatch 对不完整流量分析的查询

使用这些 Amazon CloudWatch Logs Insights 查询来识别不完整的事件流。根据所需的分析类型对相应的日志组运行这些查询。

日志组选择：

- MediaTailor/AdDecisionServerInteractions-用于分析广告决策服务器互动、广告机会和 ADS-related 失败的查询。
- MediaTailor/TranscodeService-用于分析由于转码问题、创意处理失败或其他与广告无关的问题而未插入广告的问题。

Example在没有成功实现个性化设置的情况下识别广告机会

日志组： MediaTailor/AdDecisionServerInteractions

以下查询确定了未成功实现清单个性化的广告机会：

```
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分析事件流完成率

日志组： MediaTailor/AdDecisionServerInteractions

以下查询分析了不同事件类型的完成率：

```
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查找缺失信标事件的会话

日志组： MediaTailor/AdDecisionServerInteractions

以下查询标识已填满空白但没有相应信标事件的会话：

```
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识别与转码相关的广告插入故障

日志组： MediaTailor/TranscodeService

以下查询可识别妨碍成功插入广告的转码问题：

```
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
```

分析事件计时问题

事件时序分析有助于识别性能瓶颈并优化广告插入工作流程。不寻常的时间模式通常表示影响观众体验的潜在问题。

性能定时阈值

使用这些定时阈值来识别潜在的性能问题。

- 总流量持续时间超过 5 秒：可能会影响查看者的体验，并可能表明 ADS 性能问题、源服务器问题（例如清单检索超时）或内部 MediaTailor 问题，包括 NAT 网关、DynamoDB、EC2 或其他系统组件的基础设施问题。
- ADS 响应时间超过 2 秒：提示 ADS 性能问题或网络延迟问题。
- 清单个性化超过 1 秒：可能表示创意处理延迟、源服务器问题（例如清单检索超时）或内部 MediaTailor系统问题，包括 NAT 网关、DynamoDB、EC2 或其他组件的基础设施限制。

时序分析查询

使用这些查询来分析事件计时模式。

Example测量总事件流持续时间

以下查询测量事件流的总持续时间，并识别超过 5 秒的事件流：

```
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分析 ADS 响应时间

以下查询分析 ADS 响应时间并确定超过 2 秒的响应时间：

```
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识别缓慢的清单个性化设置

以下查询可识别耗时超过 1 秒的清单个性化流程：

```
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
```

常见的事件流问题和解决方案

本节按问题类型和症状为经常遇到的事件流问题提供解决方案。

广告决策服务器请求失败

症状：检测到广告机会后事件流停止。未记录 ADS 请求事件。

常见原因和解决方案

- ADS 网址配置错误：验证您的播放配置中的 ADS 网址是否正确且可访问。在广告互动日志中，您将看到 ADS 请求事件 (MAKING_ADS_REQUEST)，但没有相应的 VAST 响应，通常伴随着一个 ERROR_UNKNOWN 或类似的错误事件。
- 网络连接问题：检查与您的 ADS MediaTailor 之间的网络连接，包括防火墙规则和 DNS 解析。
- SSL/TLS 证书问题：确保您的 ADS 使用来自可信证书颁发机构的有效 SSL 证书。具体而言，对于谷歌广告管理器，您可能需要联系 AWS 支持部门，以启用接受谷歌 SSL 证书的配置标志。

诊断查询

以下查询通过跟踪事件序列来帮助诊断 ADS 请求失败：

```
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
```

广告决策服务器响应失败

症状：ADS 请求成功但 MediaTailor未收到响应，或者出现解析错误。

常见原因和解决方案

- VAST 格式无效：根据 VAST 规格标准验证您的 ADS VAST 响应。
- ADS 超时问题：增加 ADS 超时设置或优化 ADS 响应时间。
- 清空广告库存：在 ADS 配置中检查广告库存的可用性和定位标准。

诊断查询

以下查询通过检查请求和响应事件来帮助诊断 ADS 响应失败：

```
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
```

明显的个性化失败

症状：收到了 VAST 回复，但明显的个性化设置失败或广告被跳过。

常见原因和解决方案：

- 创意转码问题：检查广告是否是 NEW_CREATIVE，这需要在插入之前进行转码。
  您还可以通过检查 MediaTailor/TranscodeService 日志中是否存在错误事件（例如INTERNAL_ERROR、MISSING_VARIANTS,或PROFILE_NOT_FOUND）来检查是否存在转码错误。
- 时长不匹配问题：验证广告时长是否符合可用的广告时段持续时间。
- 个性化阈值问题：查看播放配置中的个性化阈值设置。

诊断查询

以下查询通过检查 VAST 响应和已填充的空缺来帮助诊断明显的个性化故障：

```
fields @timestamp, eventType, sessionId, skippedAds
| filter sessionId = "your-session-id"
| filter eventType in ["VAST_RESPONSE", "FILLED_AVAIL", "WARNING_NO_ADVERTISEMENTS"]
| sort @timestamp asc
```

查询跳过广告的原因

以下查询提供了有关跳过广告的原因的详细信息：

```
fields @timestamp, eventType, sessionId, skippedAds.reason, skippedAds.creativeUniqueId
| filter sessionId = "your-session-id"
| filter eventType = "WARNING_NO_ADVERTISEMENTS" or ispresent(skippedAds)
| sort @timestamp asc
```

查询跳过广告的原因和创意唯一 ID

以下查询提供了详细的跳过广告信息，包括每种用途的前两个广告的原因和创意唯一 ID：

```
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
```

跟踪信标故障

症状：成功进行清单个性化设置，但跟踪信标缺失或失败。

常见原因和解决方案

- Client-side 实现问题：大多数跟踪信标问题源于客户端实现问题，例如轮询跟踪 URL 的频率不够高，无法进行客户端跟踪，或者玩家特定的信标触发逻辑问题。
- 跟踪 URL 可访问性问题：验证 VAST 响应中的跟踪 URL 是否可访问并返回相应的响应。当无法访问 URL 或 MediaTailor 遇到阻碍成功发送跟踪响应的内部问题时，可能会出现问题。
- 玩家区段请求问题：当客户端玩家实际上没有请求任何区段时，可能会出现明显的跟踪信标故障。这会导致没有发送信标，这看起来像是跟踪失败，但实际上是玩家实现问题，而不是信标问题。

诊断查询

以下查询通过检查已填充的可用空间和信标事件来帮助诊断跟踪信标故障：

```
fields @timestamp, eventType, sessionId
| filter sessionId = "your-session-id"
| filter eventType in ["FILLED_AVAIL", "BEACON_FIRED", "ERROR_FIRING_BEACON_FAILED"]
| sort @timestamp asc
```

事件流监控最佳实践

实施以下监控措施以主动识别和解决事件流问题：

设置 CloudWatch 警报

创建 Amazon CloudWatch 警报以监控关键事件流指标。

- 流量完成率警报：当成功的清单个性化与广告机会的比率降至可接受的阈值以下时发出警报。
- ADS 响应时间警报：监控平均 ADS 响应时间，并在其超过性能阈值时发出警报。
- 错误率警报：跟踪错误事件频率，并在特定错误类型的异常峰值时发出警报。

定期监控查询

定期运行这些查询以保持对事件流运行状况的可见性：

Example每日活动流成功率

以下查询按事件类型提供了事件流成功率的每日概览：

```
fields @timestamp, eventType
| filter @timestamp > datefloor(@timestamp, 1d)
| stats count() as total_events by eventType
| sort total_events desc
```

Example每小时错误率趋势

以下查询按小时跟踪错误率，以确定趋势问题：

```
fields @timestamp, eventType
| filter eventType like /ERROR_/
| stats count() as error_count by datefloor(@timestamp, 1h) as hour
| sort hour desc
```

性能优化指南

使用事件流分析来优化广告插入性能。

- ADS 优化：与您的 ADS 提供商合作，优化响应时间并减少延迟。
- 创意准备： Pre-transcode广告素材可匹配您的内容配置文件并减少处理延迟。
- 配置调整：根据事件流分析调整超时设置、个性化阈值和其他配置参数。

其他故障排除资源

有关事件流分析之外的其他故障排除指导：

- 有关详细的日志格式信息和技术规格，请参阅查看日志。

- 有关常见广告插入问题的全面故障排除，请参阅排查常见 问题。
- 有关监控和警报设置指南，请参阅监控 AWS Elemental MediaTailor 使用亚马逊 CloudWatch 指标。
- 有关调试日志记录程序，请参见生成调试日志。

从播放问题进行故障排除 MediaTailor

本节提供有关在测试播放器软件时以及在正常处理播放器请求期间可能收到的 HTTP 错误代码的信息。

> **Note**
>
> 在PutPlaybackConfiguration和等配置操作期间，您可能还会收到来自 AWS Elemental MediaTailor API 的错误GetPlaybackConfiguration。有关这些错误类型的信息，请参阅 AWS Elemental MediaTailor API 参考。

当您的玩家直接或通过 CDN 向发送请求时，会使用状态码进行 MediaTailor 响应。 AWS Elemental MediaTailor如果 MediaTailor 成功处理请求，它将返回 HTTP 状态码200 OK（表示成功）以及填充的清单。如果请求不成功，则 MediaTailor 返回 HTTP 状态码、异常名称和错误消息。

AWS Elemental MediaTailor 返回两类错误：

- 客户端错误 — 通常由请求本身的问题引起的错误，例如请求格式不正确、参数无效或 URL 错误。这些错误具有一个 HTTP 4xx 响应代码。
- 服务器错误 — 通常由其依赖项问题 MediaTailor 或其依赖项之一引起的错误，例如广告决策服务器 (ADS) 或源服务器。这些错误具有一个 HTTP 5xx 响应代码。

主题

- 返回的客户端播放错误 AWS Elemental MediaTailor
- 返回的服务器播放错误 AWS Elemental MediaTailor
- 播放错误示例

返回的客户端播放错误 AWS Elemental MediaTailor

一般指导：

- 您可以在响应的标题和正文中找到大多数错误的详细信息。
- 对于某些错误，你需要检查配置设置。您可以从中检索播放配置的设置 AWS Elemental MediaTailor。对于 API 来说，资源是`GetPlaybackConfiguration/Name`。有关详细信息，请参阅《AWS Elemental MediaTailor API 参考》https://docs.aws.amazon.com/mediatailor/latest/apireference/。

下表列出了清单操作活动返回的客户端错误代码 AWS Elemental MediaTailor、可能的原因以及您可以采取的解决这些错误的措施。

代码	异常名称	含义	操作
40	`BadRequestException`	MediaTailor 由于格式或内容中存在一个或多个错误，无法为请求提供服务。参数可能格式不正确，或者请求可能包含无效的播放配置或会话 ID。	检查您的请求格式是否正确，是否包含准确的信息。确保播放器上的播放终端节点设置与 `GetPlaybackConfiguration` 返回的 `ManifestEndpointPrefix` 设置相匹配。重试您的请求。
40	`AccessDeniedException`	请求中提供的主机标头与 MediaTailor 播放 URL 中配置的清单终端节点前缀不匹配。您的 CDN 可能配置错误。	检查您的 CDN 设置，并确保为 MediaTailor 使用正确的清单终端节点前缀。重试您的请求。
40	`NotFoundException`	MediaTailor 找不到指定的信息。可能的原因包括未映射服务中任何内容的 URL，未定义的配置或不可用的会话。	检查您的配置和请求的有效性，然后重新初始化此会话。
40	`ConflictException`	播放器尝试同时为单个会话加载多个播放列表。结果， MediaTailor 检测到会话一致性冲突。HLS 播放器会出现此问题。	确保您的播放器一次请求一个播放列表。这符合 HLS 规范。
41	`Gone`	AWS 支持运营商封锁了玩家会话或客户配置。 AWS 当我们检测到	如果您认为不应阻止该请求，请联系 AWS Support 。他们可以

代码	异常名称	含义	操作
		来自单个会话或配置的错误流量的 4xx 请求时，支持人员会在极少数情况下这样做。	调查问题并删除阻止筛选器（如果适用）。

如果您需要进一步的帮助，请联系 AWS 支持。

返回的服务器播放错误 AWS Elemental MediaTailor

一般指导：

- 您可以在响应的标题和正文中找到大多数错误的详细信息。
- 对于某些错误，你需要检查配置设置。您可以从中检索播放配置的设置 AWS Elemental MediaTailor。对于 API 来说，资源是`GetPlaybackConfiguration/`*`Name`*。有关详细信息，请参阅《AWS Elemental MediaTailor API 参考》https://docs.aws.amazon.com/mediatailor/latest/apireference/。

下表列出了清单操作活动返回的服务器错误代码 AWS Elemental MediaTailor、可能的原因以及您可以采取的解决这些错误的措施。

代码	异常名称	含义	操作
50	`InternalServiceError`	未处理的异常。	重试请求。如果问题仍然存在，请在上查看您所在 AWS 地区报告的 MediaTailor 运行状况https://status.aws.amazon.com/。
50	`BadGatewayException`	源服务器地址或广告决策服务器 (ADS) 地址无效。无效地址如私有 IP 地址和 `localhost` 。	确保您的配置具有 ADS 和源服务器的正确设置，然后重试该请求。

代码	异常名称	含义	操作
50	UnsupportedManifestException	要么是源清单已更改， MediaTailor无法对其进行个性化设置，要么 MediaTailor 不支持源清单格式。	这可能仅影响单个会话。重新初始化会话。您通常可以通过在查看器中刷新页面来完成此操作。如果问题仍然存在，请验证 MediaTailor 是否支持来源的清单格式。有关信息，请参阅整合内容源。
50	LoadShed	MediaTailor 在处理您的请求时遇到了资源限制。	重试请求。如果问题仍然存在，请在上查看您所在 AWS 地区报告的 MediaTailor 运行状况https://status.aws.amazon.com/。
50	ThrottlingException	您的每秒交易数已达到配额， MediaTailor 会限制您的使用。	重试请求。您还可以在以下地址查看您所在 AWS 地区的报告运行状况https://status.aws.amazon.com/。 MediaTailor 您可能希望提高每秒交易数的配额。有关更多信息，请参阅 the section called “广告插播配额”。
50	GatewayTimeoutException	联系源服务器 MediaTailor 时出现超时。	重试请求。如果问题仍然存在，请检查源服务器的运行状况，并确保源服务器在the section called “广告插播配额”中列出的内容源服务器超时内做出响应。

如果您需要进一步的帮助，请联系 AWS 支持。

播放错误示例

本节列出了一些在命令行交互中可能会看到的播放错误的示例 AWS Elemental MediaTailor。

以下示例显示广告决策服务器 (ADS) 或源服务器 AWS Elemental MediaTailor 之间发生超时时的结果。

```
~[]> 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/
v1/44445555666611112222333 3/index.mpd], asset path: [index.mpd], sessionId:
[12345678901212345678 9012] customerId:[123456789012]"}%
```

信标重复数据删除

本节提供服务器端跟踪中的信标重复数据删除功能 MediaTailor的故障排除指南。

症状：信标数量低于先前观察到的信标数

原因：Beacon 重复数据删除可以防止以前多次计数的重复事件。

分辨率：这是预期的行为，表明测量结果更准确。新的数量代表实际的广告观看事件，而不是技术细分请求。更新您的报告基准以反映经过重复数据删除的计数。

症状：信标数量因玩家实现而异

原因：不同的玩家可能有不同的分段请求模式，但无论这些差异如何，重复数据删除都能确保信标数量的一致性。

解决方案：这是预期的行为。重复数据删除可对不同的玩家类型和网络条件进行一致的测量。

症状：预期和实际信标数量之间的差异

原因：之前的预期可能是基于信标重复发射导致的虚高计数。

解决方案：查看历史数据以确定重复信标的模式。使用新的重复数据删除计数作为测量和报告的准确基准。

参数疑难解答

要解决与动态广告变量、清单查询参数、字符限制、长度限制和配置别名相关的问题，请参阅MediaTailor 参数疑难解答指南中的MediaTailor ADS 请求的动态广告变量。

MediaTailor 跳过广告疑难解答指南

跳过广告是 MediaTailor 客户报告的最常见问题之一。本指南可帮助您识别广告跳过问题，并提供分步故障排除程序。

症状和影响

出现跳过广告时，您可能会观察到以下症状：

- 广告不会在预期的广告休息时间内出现
- 广告时段显示的是内容而不是广告
- 在不同的观看会话中广告播放不一致
- CloudWatch 显示具有各种跳过原因`AdSkipped`的事件的日志

跳过广告会直接影响收入，并可能造成不良的观众体验。立即解决这些问题以保持最佳性能。

确定跳过的原因

使用 CloudWatch Logs Insights 查询`MediaTailor/AdDecisionServerInteractions`日志组中是否有跳过的广告：

```
fields @timestamp, avail.availId, skippedAds.0.skippedReason,
 skippedAds.0.creativeUniqueId
| filter eventType = "FILLED_AVAIL" and ispresent(skippedAds.0.skippedReason)
| sort @timestamp desc
```

此查询返回最新的广告跳过事件及其具体原因，以帮助您识别模式。

主题

- MediaTailor NEW_CREATIVE 广告跳过故障排除
- MediaTailor ADS 超时广告跳过故障排除
- MediaTailor VAST 解析和跳过故障排除
- MediaTailor 时长不匹配广告跳过故障排除
- MediaTailor 会话变量广告跳过疑难解答
- MediaTailor 广告跳过监控和警报
- MediaTailor 广告跳过预防最佳实践
- MediaTailor 跳过广告参考指南

MediaTailor NEW_CREATIVE 广告跳过故障排除

当`NEW_CREATIVE`由于原因跳过广告时，会 AWS Elemental MediaTailor 遇到一个需要转码才能插入的广告。本故障排除指南解释了原因并提供了分步解决程序。

什么原因导致 NEW_CREATIVE 跳过

MediaTailor 根据三个关键因素对广告进行转码：创意 ID、AWS 账户 ID 和转码变体集（基础内容流的播放副本）。当创意 ID 或转码变体集的任何部分不同时，会将广告 MediaTailor 识别为需要转码的新变体。

比特率变化

当四舍五入到最接近的 8,000 位后比特率不匹配时，比特率变化可能会导致 NEW_CREATIVE 跳过。这通常发生在以下情况下：

- 底层内容流更改了主清单
- 创建新会话时使用与现有会话不同的变体
- 各播放会话中的内容流比特率不一致

收集任务带宽差异

对于收获任务，在将广告与内容流进行匹配时 MediaTailor 使用 15% 的带宽差异阈值。如果直播带宽与采集的 HLS 文件的带宽相差达到 15% 或以上，则广告将再次进行转码。这可能会导致在转码过程中跳过 NEW_CREATIVE。

这种行为通常发生在以下情况下：

- 直播和 VOD 内容均使用相同的预转码广告
- 采集的内容与原始直播的编码参数不同
- 直播内容和采集内容之间的带宽差异超过了 15% 的阈值

创意 ID 冲突

当 MediaTailor 遇到已经转码的媒体文件的其他创意 ID 时，会出现以下顺序：

1. 跳过该广告的原因是 NEW_CREATIVE
2. 这会导致不必要的转码尝试
3. 广告素材被标记为 DUPLICATE_TRANSCODE 或 COPY_DEUP

> **Note**
>
> MediaTailor 不会过期或删除经过转码的广告。我们将它们无限期地存储在 MediaTailor-owned S3 存储桶中。

解决步骤

要解决 NEW_CREATIVE 广告跳过问题，请执行以下操作：

1. 确认您的广告决策服务器为相同的广告内容返回一致的创意 ID。
2. 检查您的内容流是否保持一致的比特率和变体集。
3. 考虑实现广告预取，确保广告在播放前经过转码。有关更多信息，请参阅 the section called “预取广告”。
4. 对于持续存在的问题，请联系 AWS 支持以获取更多故障排除帮助。

监控新创意模式

使用此 CloudWatch Logs Insights 查询来分析 NEW_CREATIVE 广告跳过中的模式：

```
fields @timestamp, sessionId, creativeId, skipReason, MediaFileSourceUrl
| filter skipReason = "NEW_CREATIVE"
| stats count() by creativeId, MediaFileSourceUrl
| sort count desc
| limit 50
```

MediaTailor ADS 超时广告跳过故障排除

当出于ADS_TIMEOUT或相关原因跳过广告时，您的广告决策服务器存在连接或性能问题。 AWS Elemental MediaTailor 需要与您的 ADS 进行可靠的通信才能成功插入广告。本故障排除指南解释了如何识别和解决这些连接问题。

常见的 ADS 连接问题

常见的 ADS 连接问题包括：

- 无法从访问广告决策服务器 MediaTailor
- ADS 在配置的超时时间内未响应
- ADS 无法在高峰时段处理请求量
- MediaTailor 和您的 ADS 之间的网络连接问题

解决步骤

要解决 ADS 超时问题，请执行以下操作：

1. 确认可以从访问您的广告决策服务器 MediaTailor。

2. 检查您的 ADS 是否在配置的超时时间内作出响应。
3. 确保您的 ADS 能够在高峰时段处理请求量。
4. 考虑在主广告不可用时实施备用广告策略。

监控 ADS 性能

为该`AdDecisionServer.Timeouts`指标设置 CloudWatch 警报，以主动检测 ADS 连接问题。

MediaTailor VAST 解析和跳过故障排除

使用`VAST_PARSING_ERROR`或跳过广告时`MEDIA_FILE_UNAVAILABLE`，您的 VAST 响应或广告媒体文件出现问题。 AWS Elemental MediaTailor 需要格式正确的 VAST 响应和可访问的媒体文件才能成功插入广告。本故障排除指南解释了如何识别和解决这些问题。

常见的 VAST 问题

常见的 VAST 问题包括以下内容：

- VAST 响应格式不符合 VAST 规范
- VAST 响应中的媒体文件网址无法公开访问
- 特殊字符在 VAST XML 中未正确编码
- VAST 响应缺少所需的媒体文件或格式

解决步骤

要解决 VAST 解析问题，请执行以下操作：

1. 根据 VAST 规范验证您的 VAST 响应格式。
2. 确保 VAST 响应中的所有媒体文件网址均可公开访问。
3. 检查 VAST XML 中特殊字符的编码是否正确。
4. 验证您的 VAST 响应中是否包含与兼容格式的媒体文件 MediaTailor。

VAST 包装器疑难解答

对于与`INVALID_VAST_WRAPPER_AD`或有关的问题`REJECTED_REPLICA_VAST`：

- 根据 VAST 规范验证 VAST 包装器响应

- 确保所有包装器元素的格式正确且包含有效的 VASTAdTagURI
- 检查广告服务器配置中是否存在重复内容检测策略
- 确保 VAST 回复在同一会话中包含独特的创意内容

MediaTailor 时长不匹配广告跳过故障排除

当出于或格式相关原因跳过广告时 `AVAIL_DURATION_EXCEEDEDLEFTOVER_AVAIL_EXCEEDED_THRESHOLD`，您会遇到持续时间或格式兼容性问题。 AWS Elemental MediaTailor 需要正确的持续时间格式以及广告内容和可用的广告休息时间之间的匹配。本故障排除指南解释了如何识别和解决这些与持续时间相关的问题。

常见的持续时间问题

常见的持续时间问题包括：

- 广告超过可用广告间隔时间
- 内容中的广告插播标记不正确，表明时长不正确
- 广告插入未达到个性化阈值
- EXT-X-CUE-OUT 标签中的持续时间格式不正确

时长格式要求

EXT-X-CUE-OUT 标签持续时间参数必须格式化为整数值，而不是 ISO 8601 持续时间格式。

时长格式要求

格式	示例	Status
整数（正确）	`32`	支持-代表 32 秒
十进制（正确）	`30.000`	支持-代表 30 秒
ISO 8601（不正确）	`PT32S`	不支持-导致插入失败

解决步骤

要解决时长不匹配的问题，请执行以下操作：

1. 确保您的 ADS 返回的广告符合可用的广告间隔时间。
2. 检查内容中的广告插播时间标记是否正确表明了预期的持续时间。
3. 验证 EXT-X-CUE-OUT 持续时间参数是否使用整数格式。
4. 如果适合您的用例，可以考虑调整个性化阈值。

MediaTailor 会话变量广告跳过疑难解答

会话变量在广告定位和选择中起着至关重要的作用 AWS Elemental MediaTailor。会话变量配置不正确是广告跳过问题的常见原因。这份全面的故障排除指南解释了如何识别和解决会阻止成功插入广告的会话变量问题。

常见的会话变量问题

常见的会话变量问题包括：

- 缺少必需的变量：您的广告决策服务器可能需要未提供的特定变量。
- 变量语法不正确：变量必须使用正确的语法（例如，`[session.id]`代替`${session.id}`）。
- URL 编码问题：变量值中的特殊字符可能需要正确的 URL 编码。
- 玩家参数不一致：玩家参数必须一致地跨会话传递。
- 动态变量解析失败：无法解析的变量将替换为空字符串。
- SCTE-35 UPID 解析问题：分段 UPID 处理问题可能会导致会话变量解析失败。

验证会话变量分辨率

要验证您的会话变量是否已正确解析，请执行以下操作：

1. 为您的 MediaTailor 配置启用调试日志
2. 在`MediaTailor/AdDecisionServerInteractions`日志组中查看实际的 ADS 请求网址
3. 验证模板 URL 中的所有变量都已替换为相应的值
4. 查找所有被空字符串替换的变量，这可能表示解析失败

SCTE-35 UPID 解析疑难解答

SCTE-35 分段 UPID 处理出现问题可能会导致会话变量问题：

- 格式要求：**format_identifier**为了正确处理，UPID 必须segmentation_upid_type为 12 且包含在内
- 解析规则：解码后的 UPID 可以包含多个值的冒号分隔符。模板变量和解码后的 UPID 令牌的数量必须相等
- 无效格式：避免使用没有值的双冒号（例如，::或:46175218::4053），因为这会导致解析失败

MediaTailor 广告跳过监控和警报

主动监控可帮助您在广告跳过问题对收入产生重大影响之前发现和解决这些问题。 AWS Elemental MediaTailor 提供全面的指标和日志记录功能，可有效监控广告插入效果。本指南介绍如何为广告跳过问题设置有效的监控。

需要 CloudWatch 监控的关键指标

为以下关键 MediaTailor 指标设置 CloudWatch 警报：

- AdDecisionServer.Ads.Skipped-跳过的广告数量
- AdDecisionServer.Timeouts-ADS 超时次数
- Avail.FilledDuration-已填充广告间隔的持续时间
- Avail.SlateOnly-仅包含广告板的广告插播次数

高级 CloudWatch 日志见解查询

使用以下专业查询进行详细的故障排除：

全面的会话分析

要详细分析特定会话的广告插入行为，请执行以下操作：

```
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
```

寻找创意 ID

要从 FILLED_AVAIL 事件中识别创意 ID，请执行以下操作：

```
fields @timestamp, sessionId, eventType
| filter sessionId like /sessionId/ and eventType!='BEACON_FIRED'
| sort @timestamp desc
```

> **Note**
>
> `sessionId`替换为你正在调查的实际会话 ID。

MediaTailor 广告跳过预防最佳实践

实施这些最佳实践有助于在广告跳过问题发生之前将其预防，从而确保更好的广告插入性能和收入保护。 AWS Elemental MediaTailor这些主动措施可以解决跳过广告的最常见原因，并有助于保持广告投放的稳定性。

主动措施

- 实现广告预取：使用 MediaTailor预取功能确保广告在播放前经过转码。有关实现the section called “预取广告”的详细信息，请参见
- 保持一致的创意 ID：确保您的广告决策服务器在各个会话中对相同的广告内容使用一致的创意 ID
- 确保正确的持续时间格式：使用整数值作为 EXT-X-CUE-OUT 持续时间参数，而不是 ISO 8601 格式
- 配置 VOD 优化：为具有多个广告时段的 VOD 流设置 “最大并发 ADS 请求数” 设置，以减少服务器负载。有关更多信息，请参阅 高级设置。
- 优化 ADS 性能：配置您的广告决策服务器以快速响应并处理峰值流量

实施指南

- 实施正确的错误处理：配置 Slate 内容以在无法插入广告时填充广告间隔
- 彻底测试：验证您在不同设备和网络条件下的广告插入工作流程
- 实施备用策略：配置备用广告来源或默认广告，以备主要广告来源失效时使用

- 监控转码模式：监控 CloudWatch 日志以提高转码效率，如果您发现模式表明转码存在问题，请联系 AWS Support

MediaTailor 跳过广告参考指南

本节提供有关跳过广告的原因的全面参考信息以及相关文档的链接 AWS Elemental MediaTailor。使用本参考指南了解不同跳过原因的具体含义，并查找其他故障排除资源。

完整广告跳过原因参考

MediaTailor 在日志组`FILLED_AVAIL`的事件日志消息中记录跳过广告的`MediaTailor/AdDecisionServerInteractions`具体原因。

完整跳过广告的理由

跳过理由	说明
`NEW_CREATIVE`	该广告尚未进行转码。当 MediaTailor遇到需要在插入前进行转码的新广告素材时，就会出现这种情况。
`PROFILE_NOT_FOUND`	与会话配置相关的 MediaConvert 转码配置文件不存在，因此无法准备广告。
`TRANSCODE_ERROR`	广告转码过程遇到错误并未能完成。
`TRANSCODE_IN_PROGRESS`	广告转码仍在进行中，尚未准备好插入。
`INTERNAL_ERROR`	处理广告时出现内部 MediaTailor 错误，无法插入。
`AVAIL_DURATION_EXCEEDED`	广告无法在广告间隔的剩余时长内播放。
`LEFTOVER_AVAIL_EXCEEDED_THRESHOLD`	本可以插入的所有广告的累计时长未达到为会话配置的个性化阈值。
`VAST_PARSING_ERROR`	来自广告决策服务器的 VAST 响应包含错误或格式不正确。

跳过理由	说明
ADS_TIMEOUT	广告决策服务器未在配置的超时时间内做出响应。
MEDIA_FILE_UNAVAILABLE	无法访问 VAST 响应中指定的广告媒体文件。
SESSION_INITIALIZATION_FAILED	会 MediaTailor 话无法正确初始化，通常是由于会话变量不正确。
EARLY_CUE_IN	由于提早出现提示信号，广告中断结束的时间比预期的要早，导致广告无法完全插入。
NO_VARIANT_MATCH	广告素材没有与内容流的编码参数（比特率、分辨率、编解码器）相匹配的变体。
NO_MODEL_CREATIVE_MATCH	广告素材与当前播放配置的预期型号或格式要求不符。
REJECTED_REPLICA_VAST	由于内容检测策略重复或重复，VAST 响应被拒绝。
INVALID_VAST_WRAPPER_AD	VAST 包装器广告包含无效或格式错误的包装元素，这些元素会阻止广告成功插入。
IMPORT_ERROR	广告导入过程中出现错误，无法处理广告以进行插入。
IMPORT_IN_PROGRESS	广告导入过程目前正在进行中，尚未完成。

相关资源

有关排除广告跳过问题的更多信息，请参阅以下相关主题：

- the section called “动态广告变量”-有关动态广告变量的综合指南 MediaTailor
- the section called “预取广告”-如何实现广告预取以防止与转码相关的跳过
- the section called “使用 CloudWatch 指标进行监控”- MediaTailor 使用 CloudWatch 指标进行监控

- the section called “查看日志”-如何查看和分析 MediaTailor日志
- the section called “事件流疑难解答”-了解广告插入事件流程

AWS Elemental MediaTailor 资源

下表列出了在您使用 AWS Elemental MediaTailor时可提供帮助的相关资源。

资源	说明
SCTE 标准：SCTE 35	的 SCTE 标准文档。 SCTE35
课程和研讨会	链接到基于角色的课程和专业课程以及自定进度的实验室，可帮助您提高 AWS 技能并获得实践经验。
AWS 开发者工具	指向开发者工具 SDKs、IDE 工具包以及用于开发和管理 AWS 应用程序的命令行工具的链接。
AWS 白皮书	链接到由 AWS 解决方案架构师或其他技术 AWS 专家撰写的技术白皮书的完整列表，这些白皮书涵盖架构、安全和经济学等主题。
AWS Support 中心	创建和管理 Support AWS 案例的中心。还包括指向其他有用资源的链接，例如论坛、技术 FAQs、服务运行状况和 T AWS rusted Advisor。
AWS 支持	有关 AWS Support信息的主要网页，Su one-on-one pport是一个快速响应的支持渠道，可帮助您在云中构建和运行应用程序。
联系我们	用于查询 AWS 账单、账户、事件、滥用行为和其他问题的中央联络点。
AWS 网站条款	有关我们的版权和商标、您的账户、许可和网站访问以及其他主题的详细信息。

的文档历史记录 AWS Elemental MediaTailor

下表介绍了此文档的重要更改。

变更	说明	日期
添加了 ADS 超时、个性化时间预算和并发设置	添加了有关可配置广告决策服务器 (ADS) 请求超时、个性化时间预算和播放配置并发性的文档，包括新的预取超时设置。	2026 年 7 月 23 日
Account-level 由于跳过广告的原因而产生的尺寸	增加了指标的账户级（无维度）排放量。SkippedReason	2026 年 7 月 21 日
新的指标和账户级别维度	添加了Avail.Complete 指标。增加了Requests、、、、AdsBilled Avail.Impression 、Avail.Complete 和AdDecisionServer.Ads 指标的账户级（无维度）排放量。Avail.FilledDuration Avail.Duration	2026 年 6 月 30 日
使用 Google 广告平台自动进行服务器端身份验证	更新了本章，以反映 MediaTailor 现在会自动对与谷歌广告管理器 (GAM)、谷歌广告管理器 (GCM) 和谷歌显示和视频 360 (DV360) 的服务器间连接进行身份验证。无需客户操作或支持票。更新了服务器端部分以描述 MediaTailor 适用的 URI-based 规则，并删除了手动支持票证程序。	2026 年 4 月 15 日

添加了 TPS 流量整形	更新了章节，增加了有关根据 TPS 限制对 ADS 的请求的信息。	2025 年 10 月 2 日
新增了 prefetchScheduleName	在 ADS 日志prefetchScheduleName 中添加了有关信息。	2025 年 9 月 24 日
重组后的广告跳过故障排除指南	重新设计了故障排除指南，将每个部分重点放在要完成的特定工作上。	2025 年 8 月 26 日
添加了查询参数处理文档	添加了介绍如何 MediaTailor 处理特定起源功能的查询参数的章节，例如使用时移查看。MediaPackage	2025 年 8 月 26 日
SGAI VOD 配置指南	为服务器引导的带有视频点播内容的广告插入添加了配置指南。	2025 年 8 月 7 日
集成的参数故障排除指南	添加了全面的故障排除部分，涵盖字符限制错误、长度限制问题、配置别名问题、参数处理流程调试和安全最佳实践。	2025 年 8 月 4 日
增强的参数格式化文档	为清单查询参数和 ADS 参数添加了全面的字符限制、 URL-encoding 支持和长度限制。	2025 年 8 月 1 日
带有全面参考的增强型参数文档	添加了全面的参数字符限制、长度限制、 URL-encoding 支持、配置别名文档和基于 AWS 指南分析的会话初始化方法。	2025 年 8 月 1 日

增强的清单查询参数文档	根据 AWS 指南的分析，添加了全面的参数放置行为、字符限制、 URL-encoding 支持和协议特定差异。	2025 年 8 月 1 日
增强的域变量文档	添加了全面的配置别名文档、支持的字段和基于 AWS 指南分析的域级参数规则。	2025 年 8 月 1 日
添加了事件流故障排除指南	新章节为使用 MediaTailor 事件流分析诊断和解决广告插入问题提供了实用指导。	2025 年 7 月 31 日
添加了广告插入事件流程文档	新章节解释了清单个性化期间按时间顺序排列 MediaTailor 的事件流程，以帮助用户了解工作流程。	2025 年 7 月 23 日
增强的客户端跟踪工作流程文档	添加了全面的客户端报告工作流程文档，其中包含详细的序列图和分步流程说明。	2025 年 7 月 16 日
CDN 集成文档	添加了有关使用 MediaTailor设置 CDN 的更全面的信息。	2025 年 6 月 30 日
为渠道汇编添加了 CDN 集成	添加了有关将 CDN 与 Channel Assembly 集成以优化内容交付的文档。	2025 年 5 月 16 日
HLS 播放列表类型	添加了 HLS 多变量和媒体播放列表的定义。	2025 年 5 月 9 日
DASH 清单类型	添加了 DASH 清单组件的定义。	2025 年 5 月 9 日
添加了 HLS 播放列表类型信息	添加了有关多变体和变体播放列表的解释，为示例提供背景信息。	2025 年 5 月 9 日

添加了 DASH 清单类型信息	添加了有关 MPD 结构和组件的解释，为示例提供背景信息。	2025 年 5 月 9 日
添加了重复预取	更新了章节，增加了有关定期预取计划的信息。	2025 年 4 月 28 日
添加了每个配置的日志筛选信息	添加了有关筛选为配置 MediaTailor 发出的日志的信息。	2025 年 3 月 31 日
添加了已售日志信息	添加了有关使用已售日志发布发 MediaTailor 出的日志的新章节...	2025 年 2 月 4 日
添加了预处理广告概览	添加了新章节，描述了在使用预处理广告时如何 MediaTailor 管理广告插入。	2025 年 1 月 30 日
增加了广告调节	添加了流媒体文件调节设置。	2025 年 1 月 30 日
SGAI 实时配置指南	为服务器引导的带有实时内容的广告插入添加了配置指南。	2025 年 1 月 15 日
添加了日志类型	添加了列出 MediaTailor 发出的日志类型的新章节。	2025 年 1 月 15 日
添加了日志筛选信息	添加了有关筛选 MediaTailor 发出的日志的新章节。	2025 年 1 月 15 日
添加了支持的查询参数格式	添加了清单查询参数和 ADS 查询参数格式的章节。	2025 年 1 月 2 日
谷歌广告管理器的集成信息	添加了有关从服务器端和客户端 MediaTailor 与 Google 广告管理器集成的章节。	2024 年 11 月 25 日

更新的变量	改breakabilityStartTime 为availabilityStartTime 。	2024 年 5 月 6 日
添加了个性化细节	添加了Insertion Mode个性化细节。	2024 年 5 月 6 日
计划规则	添加了有关计划规则的新内容。	2024 年 4 月 25 日
更新了有关创建频道的内容	添加了有关创建频道时节目规则的信息。	2024 年 4 月 20 日
更新了有关添加程序的内容	添加程序时添加了有关程序规则的信息。	2024 年 4 月 20 日
工作流监视器	分析 AWS 媒体服务，在这些服务之间创建信号图，媒体工作流程的可视化。使用信号映射使用 CloudWatch EventBridge、和生成监控警报和通知 CloudFormation。	2024 年 4 月 11 日
AlternateMedia 和 As Run 日志	添加了有关如何 Alternate Media 影响 As Run Log 的注释。	2024 年 2 月 28 日
更新了清单设置	添加了有关直通标签的信息，具体取决于所选的广告标记类型。	2023 年 11 月 28 日
SCTE-35 广告插播消息	添加了有关Daterange 与Scte35 Enhanced广告 SCTE-35 标记类型对比的插入标签的信息。	2023 年 11 月 28 日

Key-value Enhanced Scte35广告标记类型的配对	添加了有关如何 MediaTailor 处理已提交的Enhanced Scte35广告标记类型键值对的信息。	2023 年 11 月 28 日
VOD 来源广告机会	MediaTailor 现在可以自动检测 VOD 来源上的广告机会。	2023 年 10 月 6 日
新的自动检测 SigV4 身份验证类型	MediaTailor 现在支持AUTODETECT_SIGV4 访问类型。	2023 年 8 月 18 日
更新了客户端跟踪内容	更新了客户端跟踪内容以包含其他信息。	2023 年 8 月 12 日
使用实时资源的设置 MediaTailor 和 MediaPackage 工作流程	添加了有关设置、一般要求 MediaTailor以及 AWS Elemental MediaPackage 工作流程使用实时源时的行为的信息。	2023 年 5 月 24 日
清单查询参数文档	添加了描述清单查询参数的部分。	2023 年 4 月 26 日
叠加广告文档	添加了描述叠加广告的章节。	2023 年 4 月 24 日
清单文档中的广告 ID 装饰	添加了描述清单中广告 ID 装饰的章节。	2023 年 4 月 24 日
增加了AFTER_LIVE_EDGE抑制模式	AFTER_LIVE_EDGE 除模式外，现在还提供广告抑制BEFORE_LIVE_EDGE 模式。	2023 年 2 月 21 日
全新 As Run 日志	As Run 日志上的新主题。	2023 年 1 月 19 日
IAM 最佳实践更新	更新了指南，使其符合 IAM 最佳实践。有关更多信息，请参阅 IAM 安全最佳实践。	2022 年 12 月 27 日

IAM 最佳实践更新	更新了指南，使其符合 IAM 最佳实践。有关更多信息，请参阅 IAM 安全最佳实践。	2022 年 12 月 27 日
新的分段 UPID 私有数据变量	MediaTailor 现在支持额外的会话变量`scte.segmentation_upid.private_data.{index}` ，用于从分段 UPID 类型 12 (MPU) 中拆分以冒号分隔的私有数据。	2022 年 12 月 22 日
更新的配额内容	更新和重组了配额信息。	2022 年 9 月 13 日
增加并更正了频道组装配额	为频道汇编服务中的直播来源、区段交付配置、清单请求和渠 MediaTailor 道交易增加了配额。	2022 年 9 月 13 日
增加并更正了频道组装配额	增加了频道汇编服务中的日志、频道、直播来源、节目、来源位置和 MediaTailor 频道政策的配额。	2022年9月11日
新的频道集合警报表	现在，您可以看到一个解释频道集合警报的表格。	2022 年 9 月 1 日
新的亚马逊 CloudWatch 指标	添加了新 CloudWatch 指标。	2022年6月26日
广告通话主题	添加了预取支持 VAST 响应的链接。	2022 年 5 月 25 日

新的 ADS 请求变量	MediaTailor 现在在 ADS 请求中支持以下其他 SCTE-35 变量：scte.segmentation_type_id scte.availls_expected scte.deliivery_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 和scte.segmentation_event_id 。	2022 年 4 月 27 日
新的 IAM 管理策略策略主题	为添加了两个新的托管策略 MediaTailor。	2021 年 11 月 24 日
新 AWSElementalMediaTailorReadOnly 托管策略	添加了新的 AWS 托管策略，该策略授予允许对 MediaTailor 资源进行只读访问的权限。	2021 年 11 月 10 日
新 AWSElementalMediaTailorFullAccess 托管策略	添加了允许完全访问 MediaTailor资源的新 AWS 托管策略。	2021 年 11 月 10 日
新混乱的副手话题	添加了一个主题来解释如何防止出现混乱的副手问题。	2021 年 11 月 4 日

预取广告主题	MediaTailor 现在可以在广告插播时间出现之前预取广告。	2021 年 10 月 12 日
为播放配置添加了日志配置设置	使用日志配置设置来控制与播放配置日志相关的设置。	2021 年 9 月 28 日
SCTE-35 广告插播消息	添加了有关将消息与 `segmentation_descriptor time_signal` 消息一起使用的信息。	2021 年 9 月 1 日
新的线性播放模式	添加了新的线性播放模式。	2021 年 9 月 1 日
新的绝对过渡类型	增加了对绝对过渡类型的支持，您可以使用绝对过渡类型为线性通道上的节目设置挂钟开始时间。	2021 年 9 月 1 日
新的频道集合提醒主题	现在，您可以使用 MediaTailor 警报监控您的频道集合资源。当您的频道汇编资源出现问题或潜在问题时， MediaTailor会生成警报。	2021 年 7 月 14 日
更正了频道出口请求的频道组装配额	更正了频道汇编服务中 MediaTailor 频道出口请求的配额。	2021 年 6 月 29 日
新的源位置身份验证类型	MediaTailor 现在支持密钥管理器访问令牌身份验证。	2021 年 6 月 16 日
新等级信息	添加了有关每个层级支持的模式和源类型的信息。	2021 年 6 月 13 日
新的来源类型信息	对于标准频道，添加了有关该程序播放的源类型的信息。	2021 年 6 月 13 日

新的 MediaTailor 直播源文档	直播源代表您添加到来源位置的单个直播。创建频道后，您可以将直播源添加到您的来源位置，并将每个直播源与节目相关联。	2021 年 6 月 13 日
支持其他 UPID 类型	MediaTailor 现在支持 ADS 信息 (0xE) 和用户定义 (0x1) 分段 UPID 类型。	2021 年 4 月 15 日
新的分段 UPID 动态变量	有三个新的动态变量：`scte.segmentation_upid.assetId` `scte.segmentation_upid.cueData.key` 、和`scte.segmentation_upid.cueData.value` 。这些变量与播客工作流程的 MPU 分段 UPID 类型 (0xC) 结合使用。	2021 年 4 月 15 日
新渠道组装服务描述	添加了有关新频道汇编服务的信息。	2021 年 3 月 11 日
新的 MediaTailor 渠道组装服务文档	频道汇编是一项新的仅限清单的服务，允许您使用现有的视频点播 (VOD) 内容创建线性流媒体频道。	2021 年 3 月 11 日
增加了频道组装配额	为新的 MediaTailor 频道组装服务增加了配额。	2021 年 3 月 11 日
新的渠道组装条款	增加了与新渠道汇编服务相对应的条款。	2021 年 3 月 10 日

信道汇编标记支持	添加了对中标记频道汇编资源的 AWS Elemental MediaTailor支持。频道 SourceLocations 和 VodSources 支持标记。	2021 年 3 月 9 日
新的动态变量主题	MediaTailor 现在支持动态域变量。	2021 年 2 月 25 日
添加了可选的配置别名设置	在会话初始化期间，使用配置别名和域变量来动态配置域。	2021 年 2 月 25 日
新的scte.segmentation_upid 动态广告变量	增加了对会scte.segmentation_upid 话数据动态广告变量的支持。	2020 年 12 月 5 日
新的广告标记直通主题	广告标记直通现在可用于 HLS 清单。	2020 年 10 月 29 日
更新了配置高级设置	广告标记直通是一项新的播放配置高级设置。	2020 年 10 月 14 日
新的调试日志模式	关于调试日志模式的新主题。	2020 年 8 月 14 日
对保险杠的 EXT-X-CUE-OUT 持续时间属性的澄清	更新了保险杠要求，因此对于 HLS，每个标签都必须具有该duration属性。EXT-X-CUE-OUT	2020 年 8 月 5 日
会话初始化报告模式参数	增加了对会话初始化请求中reportingMode 参数的支持。现在，您可以指定在初始化会话时是 MediaTailor 执行客户端还是服务器端广告跟踪。	2020 年 7 月 30 日
保险杠的新话题	添加了新的 bumpers 主题	2020 年 7 月 27 日

广告抑制功能适用于 DASH	广告禁止现在可用于 DASH。从广告抑制主题中移除HLS-only了 “” 限制。	2020 年 6 月 3 日
更新控制台专属名称	更新了控制台特定的名称，反映控制台 UI 的较新版本。	2020 年 5 月 1 日
新的avail.index 动态广告变量	添加了对新的 avail.index 会话数据动态广告变量的支持。	2020 年 3 月 13 日
新Extensions 内容AdVerifications 和元素	对于客户端报告，支持 AdVerifications 和 Extensions 元素。	2020 年 3 月 10 日
个性化阈值配置	添加了对个性化阈值可选配置的支持。	2020 年 2 月 14 日
DASH VOD 清单	增加了对源服务器的视频点播 (VOD) DASH 清单的支持，具有多周期清单输出。	2019 年 12 月 23 日
控制台支持转码配置文件名称	在配置中增加了转码配置文件名称的描述。	2019 年 12 月 23 日
更新了限额表	更新了 ADS 重定向和 ADS 超时的限制。	2019 年 12 月 18 日
记录片前直播行为	添加了Pre-Roll 广告插入部分，以描述直播前贴广告的使用 AWS Elemental MediaTailor方式。	2019 年 11 月 26 日
支持直播前贴广告	添加了对在实时流开始时插入前置式广告的支持。	2019 年 9 月 11 日

分析 Amazon Logs 见解中的 ADS CloudWatch 日志	添加了有关使用 AWS Elemental MediaTailor ADS 日志和 CloudWatch 日志见解来分析 MediaTailor 会话的信息。	2019 年 8 月 13 日
新增安全性章节	添加了安全性章节以丰富涵盖的内容并实现标准化。	2019 年 5 月 23 日
DASH 单周期清单	添加了对源服务器的单周期 DASH 清单的支持，具有多周期清单输出。	2019 年 4 月 4 日
支持 AD SCTE-35 S 网址中的 UPID	添加了在广告决策服务器 (ADS) URL 中包含唯一程序 ID (UPID) 的支持。这允许 ADS 在直播线性流中提供程序级广告定向。	2019 年 3 月 28 日
Client-side 报告支持配套广告	对于客户端报告， AWS Elemental MediaTailor 跟踪网址响应现在包括配套广告元数据。	2019 年 3 月 28 日
HLS 广告标记文档	新增一节，介绍受支持的 HLS 广告标记。	2019 年 3 月 1 日
标记支持	添加了对 AWS Elemental MediaTailor中配置资源标记的支持。使用标记，您可以识别和组织 AWS 资源，控制对资源的访问权限并跟踪 AWS 成本。	2019 年 2 月 14 日
添加了 AWS CloudTrail 日志信息	添加了有关使用 CloudTrail 在 AWS Elemental MediaTailor API 中记录操作的主题。	2019 年 2 月 11 日

添加了关于播放错误的章节	添加了有关 MediaTailor 在播放期间可能返回的错误的信息，以响应来自播放器或内容分发网络 (CDN) 的请求。	2019 年 2 月 4 日
DASH base64 编码二进	添加了对在base64编码的二进制中（<scte35:Signal> <scte35:Binary> 标记内）的清单中提供拼接信息的支持。	2019 年 1 月 4 日
DASH 时间信号	添加了对在 <scte35:TimeSignal> 标记内的清单中提供拼接信息的支持。	2018 年 12 月 5 日
DASH 位置支持	增加了对 MPEG-DASH <Location> 标签的支持。	2018 年 12 月 4 日
DASH 支持	增加了对 MPEG-DASH 清单的支持。	2018 年 11 月 14 日
更新了限额表	更新了配置和清单大小的限制。	2018 年 10 月 13 日
新的和更新的指标	添加了广告决策服务器 (ADS) 和源超时的指标，并更新了 ADS 和源错误定义以包含超时响应。	2018 年 10 月 13 日
为服务器端和客户端广告插入用例提供更好的文档覆盖率	对说明和示例进行了扩展，以覆盖针对服务器端广告插入和客户端广告插入的动态广告变量使用。	2018 年 10 月 1 日
新区域	增加了对 PDX 和 FRA 区域的支持。	2018 年 7 月 18 日
VAST/VPAID	增加了有关 VAST 和 VPAID 的信息。	2018 年 3 月 16 日

CloudWatch	添加了有关可用 CloudWatch 指标、命名空间和维度的信息。	2018 年 3 月 16 日
新区域	增加了对亚太地区（新加坡）、亚太地区（悉尼）和亚太地区（东京）区域的支持。	2018 年 2 月 8 日
IAM 政策信息	添加了特定于的 IAM 策略信息 AWS Elemental MediaTailor。添加了有关创建具有有限权限的非管理员角色的说明。	2018 年 1 月 3 日
首次发布	本文档的第一版。	2017 年 11 月 27 日

Note

- AWS 媒体服务并非设计或设计用于应用程序或需要故障安全性能的情况，例如生命安全操作、导航或通信系统、空中交通管制或生命支持机器，在这些情况下，服务的不可用、中断或故障可能导致死亡、人身伤害、财产损失或环境损害。

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。