



Add a permission의

AWS Elemental MediaTailor



AWS Elemental MediaTailor: Add a permission의

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

Amazon의 상표 및 트레이드 드레스는 Amazon 외 제품 또는 서비스와 함께, Amazon 브랜드 이미지를 떨어뜨리거나 고객에게 혼동을 일으킬 수 있는 방식으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon과 제휴 관계이거나 관련이 있거나 후원 관계와 관계없이 해당 소유자의 자산입니다.

Table of Contents

AWS Elemental MediaTailor란 무엇인가요?	1
오리진 서버 요구 사항	1
MediaTailor 개념	2
HLS 재생 목록 유형	2
DASH 매니페스트 유형	4
광고 삽입 개념	9
채널 어셈블리 개념	10
광고 삽입 작동 방식	11
광고 삽입 이벤트 흐름	13
관련 서비스	16
MediaTailor 액세스	16
가격 책정	16
리전	16
설정	17
에 가입 AWS 계정	17
MediaTailor 시작하기	18
MediaTailor 광고 삽입 시작하기	18
사전 조건	19
1단계: MediaTailor 액세스	19
2단계: 스트림 준비	19
3단계: ADS 요청 URL 및 쿼리 파라미터 구성	21
4단계: 구성 생성	22
5단계: 구성 테스트	24
6단계: MediaTailor로 재생 요청 전송	25
(선택 사항) 7단계: 활동 모니터링	26
8단계: 정리	27
MediaTailor 채널 어셈블리 시작하기	27
사전 조건	28
1단계: 소스 위치 생성	29
2단계: 소스 위치에 VOD 소스 추가	29
3단계: 채널 생성	31
4단계: 채널 일정에 프로그램 추가	32
5단계(선택 사항): MediaTailor를 사용하여 스트림에 개인화된 광고 삽입	33
6단계: 채널 시작	34

7단계: 채널 테스트	34
8단계: 정리	35
광고 삽입	36
지원되는 오디오 및 비디오 코덱	37
광고 삽입 동작 이해	37
VOD에 대한 광고 스티칭 동작	37
라이브 광고 스티칭 동작	40
서버 가이드 광고 삽입 개요	43
재생 구성에서 활성화	43
서버 가이드 세션 생성	43
SGAI를 사용한 광고 추적	44
매니페스트 하트비트를 사용한 안내 미리 가져오기	45
SGAI 기능 호환성	46
SGAI 라이브 구성	48
SGAI VOD 구성	51
광고 서버 통합 요구 사항	55
VAST 요구 사항	55
VPAID 요구 사항	57
광고 부페	58
VAST 예	58
주요 개념	59
이점	60
활성화	60
작동 방식	63
VAST 래퍼 동작	64
지표 및 모니터링	71
ADS 이벤트 변경	73
다음 사항도 참조하세요.	73
구성 관리	74
구성 생성	74
구성 세부 정보 보기	83
구성 설정 편집	84
구성 삭제	84
콘텐츠 소스 통합	84
입력 소스 요구 사항	85
HLS 소스 통합	85

MPEG-DASH 소스 통합	96
SigV4와의 오리진 상호 작용 보호	118
Google의 광고 플랫폼과 통합	122
서버 측 통합	123
클라이언트 측 통합	124
광고 채우기 속도 최적화	125
광고 중단 억제를 사용하여 광고 중단 동작 사용자 지정	127
광고 중단 억제 구성	128
범퍼 광고 삽입	139
범퍼 구성	139
프리롤 광고 삽입	141
슬레이트 광고 삽입	142
슬레이트 구성	142
개인화 임계값 작동 방식	143
슬레이트 구성 및 VPAID	146
광고 트랜스코딩	146
동적 트랜스코딩	146
사용자 지정 트랜스코딩 프로파일	148
광고 미리 가져오기	151
프리페치 작동 방식	153
미리 가져오기 일정 생성	157
TPS 기반 트래픽 셰이핑	163
미리 가져오기 일정 삭제	164
사전 조건화된 광고	164
사전 조건화된 광고 요구 사항	165
사전 조절된 광고 워크플로	168
수익화 함수	169
함수가 MediaTailor 워크플로에 맞는 방식	169
빠른 시작 설명서	170
수명 주기 후크	174
함수 유형 및 구성	199
Elemental Inference 통합	221
생성 및 관리	233
JSONata 표현식 참조	235
함수 예제	247
문제 해결 및 모니터링	257

한도	268
수율 최적화	273
수율 최적화가 MediaTailor 워크플로에 어떻게 적용되는지	273
사전 조건	274
지원되는 콘텐츠 유형	274
빠른 시작 설명서	275
APS와 통합	278
수율 최적화 작동 방식	279
입찰 구성 및 템플릿 지정	282
ORTB 필드 참조	292
템플릿 예제	302
모범 사례	313
문제 해결 및 모니터링	316
동적 광고 변수	324
세션 변수	326
플레이어 변수	338
도메인 변수	340
서비스 변수	340
구성 별칭	349
ADS 파라미터 전달	356
파라미터 라우팅	360
MediaPackage 통합	361
세션 동작	363
파라미터 참조	365
파라미터 문제 해결	366
별칭 문제 해결	370
MediaTailor 매니페스트 쿼리 파라미터	373
오리진 쿼리 파라미터	374
MediaTailor 문자 제한	375
MediaTailor HLS 암시적 세션	376
MediaTailor DASH 암시적 세션	377
MediaTailor 명시적 세션 초기화	379
MediaTailor 프로토콜별 동작	381
MediaTailor CDN 통합	383
광고 추적 데이터 보고	384
서버 측 추적	385

클라이언트 측 추적	399
오버레이 광고	474
오버레이 광고를 사용하기 위한 사전 조건	475
시작하기	475
로그 및 지표	488
MediaTailor의 오버레이 광고에 대한 결제	491
광고 ID 장식	491
세션 상태	492
매니페스트 및 광고 메타데이터 삽입	493
Ad Decision Server(ADS) 상호 작용	517
클라이언트 측 추적 API	518
선형 어셈블 스트림 생성	520
채널 어셈블리 워크플로 유형	520
VOD-to-live	521
Live-to-live	523
VOD + Live-to-live(혼합)	527
올바른 워크플로 선택	530
다음 단계	531
소스 위치 작업	531
소스 위치 생성	532
소스 위치에 대한 인증 구성	534
VOD 소스 작업	542
라이브 소스 작업	546
패키지 구성 사용	550
매니페스트 캐싱	550
채널 작업	551
채널 생성	551
채널의 출력과 함께 소스 그룹 사용	553
채널 삭제	554
프로그램 추가	554
프로그램 생성	555
대상 그룹 및 대체 콘텐츠 정의	562
대상별 매니페스트 생성	564
광고 및 광고 시간 삽입	565
광고 삽입 설정	565
광고 시간에 대한 SCTE-35 메시지	566

채널 재생 시간 전환	572
CDN 사용	574
CDN 선택	575
CDN 통합 계획	576
트래픽 요구 사항 추정	578
최적화 전략 구성	578
계획 사용자 지정	580
모니터링 및 조정 설정	582
비용 최적화	583
구현 테스트	584
일반적인 문제 해결	586
CDN 통합 설정	588
CDN 통합 구성 요소 및 요구 사항	589
CDN 통합을 위한 사전 조건	19
CDN 통합 설정 단계	589
MediaTailor CDN 통합에 필요한 헤더	590
CDN 라우팅 동작 설정	591
CDN 매핑 설정	595
보안 모범 사례	598
CDN을 사용한 광고 삽입	599
필요한 사항	600
시작하기 전 준비 사항	600
CDN 통합의 이점	601
CDN 아키텍처 이해	602
기본 광고 삽입 설정	608
채널 어셈블리가 포함된 SSAI	611
CDN 성능 최적화	617
CDN 작업 모니터링	619
CDNs 사용한 광고 삽입 문제 해결	625
CDN을 사용한 채널 어셈블리	628
필요한 사항	629
시작하기 전 준비 사항	629
CDN 통합의 이점	630
CDN 아키텍처 이해	631
기본 설정	633
기본 URLs 구성	634

광고 삽입 구현	635
시간 이동 보기 구성	637
CDN 작업 모니터링	638
전체 최적화 가이드	639
MediaPackage CDN 통합	639
MediaPackage 및 CDN 워크플로 이해	640
사전 조건	640
필수 설정 구성	641
통합 확인	643
다음 단계	644
매니페스트 필터링	644
CDN 캐싱	650
CDN 성능	654
CDN 통합 문제 해결	658
CloudFront 통합	662
기본 CloudFront 설정	663
성능 최적화	669
다중 리전 복원력	673
모니터링 및 문제 해결	676
타사 CDN 설정	681
사전 조건	681
CDN 캐싱 규칙 구성	682
CDN 구성 확인	685
CDN 성능 최적화	686
타사 CDN 문제 해결	687
CDN 성능 최적화	688
캐싱 최적화	689
라우팅 최적화 요청	693
성능 벤치마크	693
고급 최적화	694
CDN 모니터링	697
필수 CDN 성능 지표	698
모니터링 도구 설정	700
구현 체크리스트	701
관련 주제	702
CDN 통합 테스트	702

사전 조건 테스트	703
체계적 테스트 방법론	703
배포 전 체크리스트	708
테스트 도구 참조	709
CDN 통합 문제 해결	710
빠른 진단 체크리스트	711
통합 테스트하기	712
매니페스트 404 오류	716
매니페스트 전송 문제	717
세그먼트 전송 문제	718
세션 관리 문제	720
광고 중단 타이밍 문제	722
성능 및 지연 시간 문제	724
디바이스 동작 불일치	730
문제 해결 준비	730
워크플로별 문제 해결 가이드	732
CDN 통합 로그 분석 참조	734
CDN 로그 해석	734
오류 코드 참조	736
로그 분석 도구	738
MediaTailor 로깅 구성 리소스	738
CloudFormation 자동화	740
를 사용하는 이유 AWS CloudFormation	740
배포 준비	741
템플릿 배포	744
배포된 리소스 사용	745
배포 테스트 및 검증	747
배포 관련 문제 해결	748
템플릿 사용자 지정	750
템플릿 참조	753
프로덕션 CloudFront 구성	759
3원 아키텍처	759
캐시 동작 우선 순위 및 구성	761
CDN 통합 지원 받기	766
Support로 에스컬레이션해야 하는 경우	767
지원을 위한 정보 수집	767

지원 리소스	768
지원 사례 모범 사례	769
보안	770
데이터 보호	770
데이터 암호화	771
자격 증명 및 액세스 관리	772
대상	772
ID를 통한 인증	773
정책을 사용하여 액세스 관리	774
AWS Elemental MediaTailor 에서 IAM을 사용하는 방법	775
ID 기반 정책 예시	780
리소스 기반 정책 예제	783
AWS 관리형 정책	783
서비스 연결 역할 사용	785
ID 및 액세스 문제 해결	788
규정 준수 확인	791
복원력	791
인프라 보안	791
교차 서비스 혼동된 대리자 방지	792
로그 및 모니터링	793
CloudWatch 경보	793
CloudTrail 로그	793
AWS Trusted Advisor	793
모니터링 및 태그 지정	794
로그 보기	795
ADS 로그	795
매니페스트 로그	851
로그 트랜스코딩	857
벤딩된 로그 사용	859
CloudWatch Logs에 로그 쓰기	863
광고 삽입 세션 로그의 볼륨 제어	872
로그 및 이벤트 필터링	874
디버그 로그 생성	877
CloudWatch 지표를 사용한 모니터링	882
AWS Elemental MediaTailor CloudWatch 지표	882
AWS Elemental MediaTailor CloudWatch 차원	901

지표를 사용하여 오래된 매니페스트 진단	902
API 호출 기록	904
AWS Elemental MediaTailor CloudTrail의 정보	905
AWS Elemental MediaTailor 로그 파일 항목 이해	906
채널 어셈블리 알림 수신	908
매니페스트 관련 알림 문제 해결	913
알림 보기	914
알림 처리	915
리소스에 태그 지정	915
지원되는 리소스	915
태그 제한 사항	916
태그 관리	916
분석 대시보드	916
분석 대시보드 사용	917
AWS 리전 선택	917
주요 수익 지표	918
추세 그래프	919
구성 테이블별 수익화 성능	919
비컨 실행 성능 테이블	920
대시보드 데이터 내보내기	921
워크플로 모니터	921
워크플로 모니터 구성 요소	923
지원되는 서비스	923
워크플로 모니터 구성	924
워크플로 모니터 사용	942
듀얼 스택 엔드포인트	944
듀얼 스택 엔드포인트란 무엇입니까?	944
지원되는 기능	944
듀얼 스택 엔드포인트 형식	944
듀얼 스택 엔드포인트 사용	945
AWS Management Console	945
AWS Command Line Interface	945
AWS SDK	945
고려 사항	946
할당량	947
광고 삽입에 대한 할당량	947

채널 어셈블리의 할당량	953
문제 해결	957
이벤트 흐름 문제 해결	957
불완전한 흐름	957
타이밍 분석	959
공통 문제	961
모니터링 모범 사례	964
추가 리소스	965
재생 문제 해결	965
클라이언트 오류	966
서버 오류	968
예제	970
비콘 중복 제거	971
파라미터 문제 해결	971
광고 건너뛰기 문제 해결	971
증상	971
사유 건너뛰기	972
NEW_CREATIVE 건너뛰기	973
ADS 제한 시간 건너뛰기	975
VAST 구문 분석 건너뛰기	975
기간 불일치 건너뛰기	976
세션 변수 건너뛰기	978
광고 건너뛰기 모니터링	979
예방 모범 사례	980
참조 자료	981
MediaTailor 리소스	984
문서 기록	985
.....	miv

AWS Elemental MediaTailor란 무엇인가요?

AWS Elemental MediaTailor 는 AWS 클라우드에서 실행되는 확장 가능한 광고 삽입 및 채널 어셈블리 서비스입니다. MediaTailor를 사용하면 대상 광고 콘텐츠를 최종 사용자에게 제공하고 선형 스트림을 생성하는 동시에 OTT(Over-the-top) 비디오 애플리케이션에서 브로드캐스트 품질을 유지할 수 있습니다. MediaTailor 광고 삽입은 온디맨드 비디오(VOD) 및 라이브 워크플로에 대해 Apple HTTP Live Streaming(HLS) 및 MPEG Dynamic Adaptive Streaming over HTTP(DASH)를 지원합니다.

AWS Elemental MediaTailor 광고 삽입은 기존 광고 추적 시스템에 비해 중요한 발전을 제공합니다. 광고는 더 나은 수익화, 비디오 품질 및 해상도의 일관성, 다중 플랫폼 환경 전반에서 더 쉽게 관리할 수 있습니다. MediaTailor는 모든 IP 연결 디바이스가 다른 콘텐츠를 렌더링하는 것과 동일한 방식으로 광고를 렌더링할 수 있도록 하여 광고 워크플로를 간소화합니다. 또한 서비스는 광고 뷰의 고급 추적을 제공하여 콘텐츠 수익성을 증진합니다.

AWS Elemental MediaTailor 채널 어셈블리는 기존 온디맨드 비디오(VOD) 콘텐츠를 사용하여 선형 스트리밍 채널을 생성할 수 있는 매니페스트 전용 서비스입니다. MediaTailor는 오리진 서버에서 직접 제공되는 콘텐츠 세그먼트에 절대 접촉하지 않습니다. 대신 MediaTailor는 오리진에서 매니페스트를 가져오고 이를 사용하여 기본 콘텐츠 세그먼트를 참조하는 라이브 슬라이딩 매니페스트 창을 조합합니다.

MediaTailor 채널 어셈블리를 사용하면 SCTE-35 마커로 조건을 지정할 필요 없이 스트림에 광고 브레이크를 삽입하여 채널을 쉽게 수익화할 수 있습니다. MediaTailor 광고 삽입 또는 다른 서버 측 광고 삽입 서비스와 함께 채널 어셈블리를 사용할 수 있습니다.

오리진 서버 요구 사항

AWS Elemental MediaTailor 에는 오리진 서버 통신에 대한 특정 요구 사항이 있습니다.

- 지원되는 포트 - MediaTailor는 표준 HTTP 및 HTTPS 포트를 사용하는 오리진만 허용합니다.
 - HTTP 연결을 위한 포트 80
 - HTTPS 연결용 포트 443

MediaTailor는 오리진 서버 통신을 위한 사용자 지정 포트를 지원하지 않습니다.

- 프로토콜 요구 사항 - 보안 통신을 위해 MediaTailor는 특정 오리진 유형 및 인증 시나리오에 HTTPS를 요구합니다. 자세한 내용은 [the section called “콘텐츠 소스 통합”](#) 단원을 참조하십시오.

MediaTailor 개념

다음은 AWS Elemental MediaTailor 사용 설명서 전체에서 사용되는 개념의 개요입니다.

HLS 재생 목록 유형

HTTP 라이브 스트리밍(HLS)은 다변량 재생 목록과 미디어 재생 목록이라는 두 가지 기본 유형의 재생 목록을 사용합니다. 이러한 재생 목록 유형의 차이점을 이해하는 것은 MediaTailor 워크플로를 구성하고 문제를 해결하는 데 필수적입니다.

다변량 재생 목록

다변량 재생 목록은 HLS 스트리밍 워크플로에서 사용 가능한 모든 콘텐츠 변환을 나열하는 최상위 인덱스 파일입니다. 여기에는 미디어 재생 목록에 대한 참조가 포함되지만 미디어 세그먼트 자체는 포함되지 않습니다. 다중 변형 재생 목록을 사용하면 플레이어가 네트워크 조건, 디바이스 기능 또는 사용자 기본 설정에 따라 가장 적합한 변환을 선택할 수 있습니다.

이 재생 목록 유형은 다양한 컨텍스트에서 다른 여러 이름으로도 알려져 있습니다.

- 마스터 재생 목록(사용되지 않는 용어)
- 마스터 매니페스트(공통 업계 용어)
- 기본 재생 목록
- 기본 재생 목록
- 인덱스 파일
- 마스터 M3U8

MediaTailor 워크플로에서 다변량 재생 목록은 재생 요청의 진입점이며 광고 개인화가 시작되는 곳입니다.

Example다변량 재생 목록 예제

```
#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
```

미디어 재생 목록

미디어 재생 목록에는 콘텐츠의 특정 변환(품질 수준)에 대한 실제 미디어 세그먼트 정보가 포함됩니다. 여기에는 타이밍 정보, 세그먼트 URLs 및 단일 변환 재생에 필요한 기타 메타데이터가 포함됩니다. 각 미디어 재생 목록은 해상도, 비트레이트 및 기타 특성의 다양한 조합을 나타냅니다.

이 재생 목록 유형은 다른 여러 이름으로도 알려져 있습니다.

- 변형 재생 목록
- 하위 매니페스트(공통 산업 용어)
- 청크리스트
- 미디어 M3U8
- 변환 재생 목록

MediaTailor 워크플로에서 미디어 재생 목록은 콘텐츠 세그먼트와 광고 세그먼트를 모두 적절한 순서로 포함하도록 개인화됩니다.

Example미디어 재생 목록 예제

```
#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

다변량 재생 목록과 미디어 재생 목록은 모두 .m3u8 확장을 사용하는 텍스트 기반 파일입니다. 이러한 파일의 용어는 설명서와 컨텍스트에 따라 다르지만 기본 역할은 HLS 스트리밍 아키텍처에서 동일하게 유지됩니다.

MediaTailor를 구성할 때 일반적으로 콘텐츠 오리진의 다변량 재생 목록에 대한 URL을 제공합니다. 그런 다음 MediaTailor는 다변량 재생 목록과 미디어 재생 목록의 개인화를 모두 처리하여 구성에 따라 광고를 삽입합니다.

HLS 재생 목록 사양에 대한 자세한 내용은 [HTTP 라이브 스트리밍 RFC를 참조하세요](#).

DASH 매니페스트 유형

Dynamic Adaptive Streaming over HTTP(DASH)는 미디어 프레젠테이션 설명(MPD) 매니페스트를 사용하여 스트리밍 콘텐츠를 제공합니다. DASH 매니페스트의 구조와 구성 요소를 이해하는 것은 MediaTailor 워크플로를 구성하고 문제를 해결하는 데 필수적입니다.

MPD(미디어 프레젠테이션 설명)

MPD는 미디어 콘텐츠의 구조와 가용성을 설명하는 DASH 스트리밍의 기본 매니페스트 파일입니다. 여기에는 스트리밍 콘텐츠를 구성하는 기간, 적응 세트, 표현 및 세그먼트에 대한 정보가 포함되어 있습니다.

이 매니페스트 유형은 다양한 컨텍스트에서 여러 다른 이름으로도 알려져 있습니다.

- DASH 매니페스트
- 대시 MPD
- 마스터 매니페스트(HLS와 비교 시)
- 프레젠테이션 매니페스트

MediaTailor 워크플로에서 MPD는 재생 요청의 진입점이며 광고 개인화가 시작되는 곳입니다.

Example MPD 매니페스트 예제

```
<?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>

```

Period

기간은 DASH 프레젠테이션의 임시 섹션입니다. 각 기간은 하나 이상의 적응 세트를 포함하며 미디어 시간 범위를 나타냅니다. 광고 삽입 워크플로에서는 일반적으로 별도의 기간을 사용하여 콘텐츠와 광고를 구분합니다.

이 구성 요소는 다른 여러 이름으로도 알려져 있습니다.

- 콘텐츠 세그먼트
- 타임라인 섹션
- 프로그램 세그먼트

MediaTailor 워크플로에서 기간은 기본 콘텐츠를 광고 콘텐츠와 분리하는 데 사용되며, 각 광고는 일반적으로 자체 기간으로 표시됩니다.

Example기간 예제

```
<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

AdaptationSet는 하나 또는 여러 미디어 콘텐츠 구성 요소의 교체 가능한 인코딩 버전 세트를 그룹화합니다. 예를 들어 한 AdaptationSet에는 여러 오디오 언어 옵션이 포함될 수 있습니다.

이 구성 요소를 다음과 같이 알려져 있습니다.

- 미디어 구성 요소 그룹
- 스트림 세트
- 트랙 그룹

MediaTailor 워크플로에서는 콘텐츠와 광고 간에 일관된 미디어 유형을 유지하기 위해 광고 삽입 중에 AdaptationSets가 보존됩니다.

표시

표현은 AdaptationSet 내의 미디어 콘텐츠의 특정 인코딩 버전입니다. 각 표현은 일반적으로 비트 전송률, 해상도 또는 기타 인코딩 파라미터가 다르므로 클라이언트가 네트워크 조건 및 디바이스 기능에 따라 가장 적합한 버전을 선택할 수 있습니다.

이 구성 요소를 다음과 같이 알려져 있습니다.

- 변환(HLS와 유사)
- 품질 수준
- 비트레이트 변형
- 스트림 변형

MediaTailor 워크플로에서 광고 기간의 표현은 원활한 시청 경험을 보장하기 위해 콘텐츠 기간의 표현과 최대한 가깝게 매칭됩니다.

세그먼트

세그먼트는 URL에서 개별적으로 참조할 수 있는 미디어 데이터의 단위입니다. 세그먼트에는 실제 미디어 콘텐츠(비디오, 오디오 등)가 포함되며 MPD 내에서 참조됩니다. DASH에는 두 가지 주요 유형의 세그먼트가 있습니다.

- 초기화 세그먼트: 코덱 파라미터 및 타이밍 정보와 같은 표현에 대한 초기화 정보를 포함합니다.
- 미디어 세그먼트: 표현 내의 특정 시간 범위에 대한 실제 미디어 데이터를 포함합니다.

MediaTailor 워크플로에서 세그먼트 URLs은 적절한 콘텐츠 또는 광고 미디어 파일을 가리키도록 수정되는 경우가 많습니다.

Note

DASH 매니페스트는 일반적으로 .mpd 확장자를 사용하는 XML 기반 파일입니다. 이러한 파일 및 해당 구성 요소의 용어는 설명서 및 컨텍스트에 따라 다를 수 있지만 DASH 스트리밍 아키텍처에서는 기본 구조가 동일하게 유지됩니다.

MediaTailor를 구성할 때 콘텐츠 오리진의 MPD 매니페스트에 대한 URL을 제공합니다. 그런 다음 MediaTailor는 일반적으로 구성에 따라 광고에 대한 추가 기간을 삽입하여 매니페스트의 개인화를 처리합니다.

DASH 매니페스트 사양에 대한 자세한 내용은 [DASH Industry Forum 구현 지침](#)을 참조하세요.

고급 DASH 매니페스트 개념

MediaTailor에서 DASH 매니페스트를 사용할 때 다음과 같은 고급 개념을 이해하면 스트리밍 워크플로를 보다 효과적으로 구성하고 문제를 해결하는 데 도움이 될 수 있습니다.

기간 시작 시간 계산

DASH 매니페스트에서 기간 시작 시간은 MPD의 `availabilityStartTime` 속성과 기간의 `start` 속성을 기반으로 계산됩니다. 라이브 스트림의 경우 시작 시간은 가용성 시작 시간을 기준으로 하지만 VOD 콘텐츠의 경우 일반적으로 프레젠테이션 시작을 기준으로 합니다.

MediaTailor가 광고 기간을 삽입하면 콘텐츠와 광고 간의 원활한 전환을 보장하기 위해 시작 시간을 신중하게 계산합니다. 이 계산에서는 다음을 고려합니다.

- 원래 기간의 시작 시간
- 이전 광고 기간의 기간
- 광고 결정 서버 응답에 지정된 시간 오프셋

프리롤 타이밍 계산

DASH 매니페스트의 프리롤 광고는 기본 콘텐츠가 시작되기 전에 표시되므로 특별한 처리가 필요합니다. MediaTailor는 매니페스트 시작 시 적절한 시작 시간 및 기간과 함께 프리롤 광고를 별도의 기간으로 삽입합니다.

프리롤 광고의 경우 MediaTailor:

- 각 프리롤 광고에 대해 새 기간을 생성합니다.
- 첫 번째 프리롤 기간의 시작 시간을 0으로 설정합니다.
- 모든 프리롤 광고의 총 기간을 고려하여 기본 콘텐츠 기간의 시작 시간을 조정합니다.

Live-to-VOD 전환

라이브 스트림이 종료되면 DASH 매니페스트가 라이브에서 VOD(Video on Demand) 형식으로 전환될 수 있습니다. 이 전환에는 MPD의 `type` 속성에서 `dynamic`로 변경 `static`하고 `timeShiftBufferDepth` 및와 같은 다른 속성을 조정하는 작업이 포함됩니다. `minimumUpdatePeriod`.

MediaTailor는 live-to-VOD 전환 중에 매니페스트를 처리할 때 다음을 보장합니다.

- 광고 마커는 VOD 매니페스트에 보존됩니다.
- 적절한 타이밍을 유지하기 위해 기간 시작 시간이 조정됩니다.
- 매니페스트는 VOD 재생 클라이언트와 호환됩니다.

DRM 처리

DASH 매니페스트의 디지털 권한 관리(DRM) 정보는 일반적으로 `AdaptationSets` 또는 `Representations` 내의 `ContentProtection` 요소에 포함됩니다. MediaTailor는 광고 삽입 중에 이러한 요소를 보존하여 콘텐츠 보호가 손상되지 않도록 합니다.

DRM 보호 콘텐츠를 처리할 때 MediaTailor는 다음을 수행합니다.

- 원본 매니페스트에서 모든 ContentProtection 요소를 유지합니다.
- 해당하는 경우 광고 콘텐츠가 호환되는 DRM 체계를 사용하는지 확인합니다.
- 매니페스트 전체에서 DRM 관련 속성 및 요소를 보존합니다.

SCC 플래그

DASH 매니페스트의 보충 콘텐츠 제어(SCC) 플래그는 콘텐츠 특성 및 재생 요구 사항에 대한 추가 정보를 제공합니다. 이러한 플래그는 일반적으로 MPD 구조 내에 속성 또는 요소로 포함됩니다.

MediaTailor가 처리하는 일반적인 SCC 플래그는 다음과 같습니다.

- 콘텐츠 등급 정보
- 접근성 기능(닫힌 캡션, 오디오 설명)
- 콘텐츠 권고 공지
- 재생 제한

MediaTailor는 매니페스트 처리 중에 이러한 플래그를 보존하여 모든 콘텐츠 메타데이터가 개인화된 매니페스트에 유지되도록 합니다.

이러한 고급 개념을 이해하면 최적의 성능을 위해 MediaTailor를 구성하고 DASH 스트리밍 워크플로에서 발생할 수 있는 문제를 해결하는 데 도움이 됩니다.

광고 삽입 개념

다음은 광고 삽입과 관련된 개념의 개요입니다.

광고 가능 여부

광고 시간에 광고자에게 판매할 수 있는 특정 광고 시간 단위입니다. 광고 시간에는 여러 광고 시간이 포함될 수 있습니다. MediaTailor는 광고 결정 서버로부터 VAST 응답을 받으면 이러한 가용 시간을 개인 맞춤형 광고로 채웁니다.

광고 브레이크

프로그래밍 중 광고가 표시되는 기간입니다. 광고 브레이크는 콘텐츠 전(프리롤), 콘텐츠 중(미드롤) 또는 콘텐츠 후(포스트롤)에 발생할 수 있습니다. MediaTailor는 SCTE-35와 같은 마커를 통해 매니페스트의 광고 시간을 식별합니다.

광고 결정 서버(ADS)

현재 광고 캠페인 및 최종 사용자 기본 설정을 포함한 기준을 바탕으로 광고 스팟 사양을 제공하는 서버입니다.

구성

MediaTailor에서 상호 작용하는 객체입니다. 구성에는 오리진 서버 및 광고 결정 서버(ADS)에 대한 위치 정보가 들어 있습니다. 구성에는 MediaTailor 안팎으로 액세스 포인트를 제공하는 엔드포인트도 포함됩니다.

동적 트랜스코딩

콘텐츠 요청 시 광고 품질 및 형식을 기본 비디오 콘텐츠에 일치시키는 프로세스입니다. 동적 트랜스코딩은 스토리지 요구 사항을 줄이고 재생이 광고와 비디오 콘텐츠 간에 원활히 전환되도록 보증합니다.

매니페스트 조작

매니페스트가 적절한 광고 및 내용 조각을 참조하도록 오리진 서버에서 매니페스트를 재작성하는 프로세스입니다. 광고는 광고 결정 서버(ADS)의 VAST 응답에 따라 결정됩니다. 재생이 진행됨에 따라 MediaTailor는 콘텐츠 스트림에 광고 삽입 또는 광고 교체를 수행합니다.

VAST 및 VMAP

Video Ad Serving Template(VAST) 및 Video Multiple Ad Playlist(VMAP)는 광고 결정 서버가 MediaTailor의 광고 요청에 보내는 XML 응답입니다. 응답은 MediaTailor가 매니페스트에 삽입하는 광고를 지정합니다. VMAP에는 광고 시간 및 광고 시간도 포함됩니다. MediaTailor 광고 삽입의 로직에 대한 자세한 내용은 섹션을 참조하세요 [AWS Elemental MediaTailor 광고 삽입 동작 이해](#). MediaTailor가 VAST와 작동하는 방식에 대한 자세한 내용은 섹션을 참조하세요 [the section called “광고 서버 통합 요구 사항”](#).

채널 어셈블리 개념

다음은 채널 어셈블리와 관련된 개념의 개요입니다.

채널

채널은 소스 매니페스트를 선형 스트림으로 어셈블합니다. 각 채널에는 플레이어는 액세스하는 재생 URLs이 포함된 출력이 하나 이상 있습니다. 채널 출력은 VOD 소스에 대해 생성한 패키지 구성 설정에 해당합니다. 채널에는 채널 스트림에서 VOD 소스가 재생되는 시기를 결정하는 일정이 포함되어 있습니다.

패키지 구성

패키지 구성은 특정 패키지 형식 특성을 포함하는 VOD 소스를 나타냅니다. 패키지 구성을 채널 출력과 연결하여 HTTP 라이브 스트리밍(HLS)과 같은 VOD 소스의 패키지 형식에 대한 재생 스트림을 생성합니다.

일정

각 채널은 채널 일정에 정렬된 프로그램으로 구성됩니다. 일정에 따라 채널의 선형 스트림에서 프로그램이 재생되는 시간이 결정됩니다.

소스 위치

소스 위치는 자산이 저장되는 오리진 서버를 나타냅니다. Amazon S3, HTTP 서버, 콘텐츠 전송 네트워크(CDN) 또는 MediaPackage와 같은 패키징 인프라일 수 있습니다.

VOD 소스

VOD 소스는 영화나 TV 프로그램 에피소드와 같은 단일 콘텐츠를 나타냅니다. VOD 소스를 프로그램과 연결하여 채널의 선형 스트림에 추가합니다.

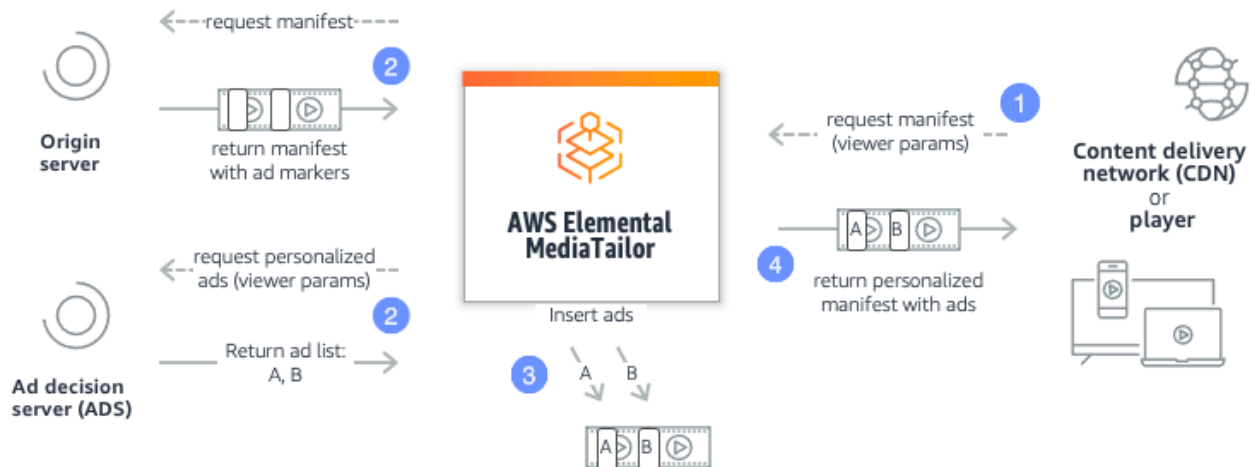
대상

대상은 선택적으로 대체 콘텐츠를 가질 수 있는 최종 사용자 코호트를 정의합니다. 표준 선형 채널에서 대상을 정의할 수 있습니다.

MediaTailor 광고 삽입 작동 방식

AWS Elemental MediaTailor 는 콘텐츠 전송 네트워크(CDN), 오리진 서버 및 광고 결정 서버(ADS) 간에 상호 작용하여 맞춤형 광고를 라이브 및 온디맨드 비디오 콘텐츠 내의 광고 시간에 연결합니다.

다음은 MediaTailor 광고 삽입의 작동 방식에 대한 개요입니다.



1. Amazon CloudFront와 같은 플레이어 또는 CDN은 HLS 또는 DASH 콘텐츠에 대한 요청을 MediaTailor에 보냅니다. 요청에는 광고 개인화에 사용되는 최종 사용자에게 대한 정보가 포함된 플레이어의 파라미터가 포함됩니다.
2. MediaTailor는 최종 사용자 정보가 포함된 요청을 ADS에 보냅니다. ADS는 최종 사용자 정보와 현재 광고 캠페인을 바탕으로 광고를 선택합니다. MediaTailor에 대한 VAST 또는 VMAP 응답에서 광고 크리에이티브에 URLs을 반환합니다.

광고를 사전 조절한 경우 URLs 사전 트랜스코딩된 광고를 가리킵니다. 사전 트랜스코딩된 광고를 사용한 광고 스티칭에 대한 자세한 내용은 섹션을 참조하세요 [사전 조건화된 광고](#).

3. MediaTailor는 ADS에서 반환된 광고 URLs을 포함하도록 매니페스트를 조작하고 오리진 콘텐츠의 인코딩 특성과 일치하도록 트랜스코딩합니다. 사전 조정된 광고를 사용하는 경우 광고가 템플릿 매니페스트와 일치하는지 확인하는 것은 사용자의 책임입니다.

콘텐츠와 일치하도록 광고가 아직 트랜스코딩되지 않은 경우 MediaTailor는 광고 삽입을 건너뛰고 MediaConvert를 사용하여 다음 요청을 준비할 수 있도록 광고를 준비합니다.

4. MediaTailor는 요청 CDN 또는 플레이어에게 완전히 개인화된 매니페스트를 반환합니다.

ADS는 광고 시작, 광고 중간 및 광고 종료와 같은 마일스톤 보기를 기반으로 본 광고를 추적합니다. 재생이 진행되면 플레이어 또는 MediaTailor는 광고 추적 비컨을 ADS 광고 추적 URL로 전송하여 광고가 얼마나 많이 표시되는지 기록합니다. MediaTailor를 사용한 세션 초기화에서 플레이어는 플레이어 또는 MediaTailor가 세션에 대해 이러한 비컨을 전송하는지 여부를 나타냅니다.

광고 삽입을 시작하는 방법에 대한 자세한 내용은 섹션을 참조하세요 [MediaTailor 시작하기](#).

AWS Elemental MediaTailor 광고 삽입 이벤트 흐름

AWS Elemental MediaTailor 는 예측 가능한 이벤트 시퀀스를 통해 서버 측 광고 삽입을 위한 매니페스트 개인화를 처리합니다. 이 이벤트 흐름을 이해하면 MediaTailor가 광고 기회를 개인화된 시청 경험으로 변환하는 방법을 이해하고 문제가 발생할 때 문제를 해결할 수 있는 기반을 제공하는 데 도움이 됩니다.

각 광고 삽입 기회는 MediaTailor가 가시성 및 모니터링을 위해 로깅하는 이벤트의 시간 순서를 따릅니다. 이러한 이벤트는 광고 기회 감지부터 추적 정보 전달에 이르기까지 매니페스트 개인화 프로세스의 주요 마일스톤을 나타냅니다.

표준 이벤트 시퀀스

MediaTailor가 광고 삽입 기회를 성공적으로 처리하면 일반적으로 다음과 같은 이벤트 시퀀스가 발생합니다.

1. 광고 기회 감지 - MediaTailor는 콘텐츠 매니페스트에서 광고 마커(예: SCTE-35)를 감지하여 광고 개인화 기회를 사용할 수 있음을 나타냅니다.
2. 광고 결정 서버 요청 - MediaTailor는 구성된 광고 결정 서버(ADS)에 요청을 보내 최종 사용자 파라미터 및 광고 중단 기간을 포함하여 감지된 기회에 대한 광고를 검색합니다.
3. 광고 응답 처리 - MediaTailor는 광고 크리에이티브 정보, 추적 URLs 및 포함할 광고에 대한 메타데이터가 포함된 ADS로부터 응답을 받습니다. VAST, VMAP 및 VPAID 응답 형식에 대한 자세한 내용은 섹션을 참조하세요 [광고 서버 통합 요구 사항](#).
4. 매니페스트 개인화 - MediaTailor는 광고 응답을 성공적으로 처리하고, 필요한 경우 광고를 트랜스코딩하고, 광고 세그먼트 및 오리진 콘텐츠 모두에 대한 상대 URLs이 포함된 개인화된 매니페스트를 생성합니다. 그런 다음 MediaTailor는 개인화된 매니페스트를 재생 디바이스로 전송합니다.
5. 비컨 활성화 추적 - 추적 비컨이 실행되어 광고 이벤트(예: 노출, 사분위수, 완료)를 광고 서버 또는 기타 측정 시스템에 다시 보고합니다. 서버 측 보고 모드(기본값)에서 MediaTailor는 플레이어 세그먼트 요청을 기반으로 비컨을 실행합니다. 클라이언트 측 보고 모드에서 재생 디바이스는 개인화된 매니페스트에 제공된 URLs 사용하여 비컨을 실행합니다.

일반적인 타이밍: 이 전체 시퀀스는 일반적으로 광고 결정 서버 응답 시간 및 광고 크리에이티브 처리 요구 사항에 따라 2~5초 이내에 완료됩니다.

일반적인 이벤트 변형

모든 광고 삽입 시도가 표준 성공 시퀀스를 따르는 것은 아닙니다. MediaTailor는 이벤트 흐름을 변경할 수 있는 다양한 시나리오를 처리합니다.

광고 응답 시나리오 비우기

광고 결정 서버가 광고를 반환하지 않는 경우:

1. 광고 기회 감지는 일반적으로 발생합니다.
2. 광고 결정 서버 요청이 성공적으로 전송되었습니다.
3. ADS는 광고가 없는 빈 응답을 반환합니다.
4. 이 기회에 대한 맞춤형 매니페스트에는 광고가 포함되지 않습니다.

이는 광고 인벤토리 가용성, 대상 지정 기준 또는 광고 결정 서버에 구성된 비즈니스 규칙으로 인해 발생할 수 있는 일반적인 시나리오입니다. 이 경우 스트림이 라이브 또는 live-to-VOD 스트림이고 슬레이트가 구성되지 않은 한 기본 콘텐츠가 표시됩니다. 자세한 내용은 [MediaTailor 슬레이트 광고 삽입](#) 단원을 참조하십시오.

오류 시나리오

기술적 문제로 인해 광고 삽입이 성공할 수 없는 경우 다음과 같은 다양한 지점에서 이벤트 흐름이 중단될 수 있습니다.

- 광고 결정 서버 제한 시간 - ADS에 대한 요청이 구성된 제한 시간 임계값을 초과하여 광고 검색을 방지합니다.
- 통신 오류 - 네트워크 또는 연결 문제로 인해 MediaTailor가 광고 결정 서버에 도달하지 못합니다.
- 응답 구문 분석 오류 - ADS는 잘못된 VAST 형식 또는 구조로 인해 MediaTailor가 구문 분석할 수 없는 응답을 반환합니다.

오류 시나리오에서 MediaTailor는 일반적으로 광고 없이 콘텐츠 재생을 계속하므로 중단 없는 시청자 경험을 보장합니다. 슬레이트가 구성된 라이브 또는 live-to-VOD 스트림의 경우 슬레이트 콘텐츠가 대신 표시될 수 있습니다. 자세한 내용은 [MediaTailor 슬레이트 광고 삽입](#) 단원을 참조하십시오.

VAST 리디렉션 시나리오

초기 VAST 응답에 다른 광고 서버로의 리디렉션이 포함된 경우:

1. 광고 기회 감지 및 초기 ADS 요청이 정상적으로 수행됨

2. 초기 응답에 리디렉션 명령 포함
3. MediaTailor는 리디렉션을 따라 최종 광고 응답을 검색합니다.
4. 매니페스트 개인 맞춤은 최종 광고 콘텐츠를 진행합니다.
5. 비컨 활성화 추적이 정상적으로 수행됨

VAST 리디렉션은 프로그래밍 방식 광고에서 일반적이며 광고 서버 체인 및 실시간 의사 결정을 허용합니다. MediaTailor는 광고 검색 프로세스 중에 최대 7개의 래퍼 리디렉션과 무제한 HTTP 리디렉션을 허용합니다.

이벤트 타이밍 개요

MediaTailor 이벤트 흐름의 타이밍 특성을 이해하면 광고 삽입 성능에 대한 적절한 기대치를 설정하는데 도움이 됩니다.

- 광고 기회 감지 - MediaTailor가 플레이어로부터 GetManifest 요청을 수신하고 콘텐츠 매니페스트에서 광고 마커를 발견할 때 발생합니다.
- 광고 결정 서버 상호 작용 - 일반적으로 ADS 응답 시간 및 네트워크 조건에 따라 100~500밀리초가 걸립니다.
- 매니페스트 개인 맞춤 - 일반적으로 ADS 응답을 수신한 후 50~200밀리초 이내에 완료됩니다.
- 비컨 타이밍 추적 - 보고 모드에 따라 다릅니다. 서버 측 비컨의 경우 타이밍은 클라이언트 플레이어의 세그먼트 요청을 기반으로 합니다. 클라이언트 측 비컨은 플레이어 구현을 기반으로 실행됩니다.

성능 고려 사항: 최적의 시청자 경험을 유지하려면 광고 기회 감지부터 매니페스트 전송까지의 총 시간이 일반적으로 5초 미만으로 유지되어야 합니다. 개별 구성 요소는 위에서 설명한 대로 밀리초 단위로 완료됩니다. 그러나 5초 임계값은 전체 광고 삽입 워크플로 중에 발생할 수 있는 잠재적 제한 시간 값, 재시도 횟수 및 네트워크 변동성을 고려합니다. 지연 시간이 길어지면 ADS 성능 문제 또는 네트워크 연결 문제가 발생할 수 있습니다.

이벤트 흐름 지식 사용

MediaTailor 이벤트 흐름을 이해하면 다음과 같은 기초가 제공됩니다.

- 구현 계획 - 이벤트 시퀀스를 알고 있으면 플레이어 통합 및 광고 서버 구성을 설계하는 데 도움이 됩니다.
- 성능 최적화 - 타이밍 기대치를 이해하면 병목 현상과 최적화 기회를 식별할 수 있습니다.
- 문제 해결 준비 - 일반적인 이벤트 흐름에 익숙하면 문제가 발생할 때 쉽게 식별할 수 있습니다.

이벤트 흐름 분석을 사용한 자세한 문제 해결 지침은 섹션을 참조하세요 [문제 해결](#). 이벤트 로깅 및 모니터링에 대한 자세한 내용은 섹션을 참조하세요 [로그 보기](#).

관련 서비스

- Amazon CloudFront는 최종 사용자에게 데이터와 비디오를 안전하게 전송하는 글로벌 콘텐츠 배포 네트워크(CDN) 서비스입니다. CloudFront를 사용하여 최고의 성능으로 콘텐츠가 제공됩니다. CloudFront에 대한 자세한 내용은 [Amazon CloudFront 웹 사이트](#)를 참조하세요.
- AWS Elemental MediaPackage은 요청하는 디바이스와 호환되는 형식으로 배포용 라이브 비디오 자산을 사용자 지정하는 적시 패키징 및 제작 서비스입니다. 를 오리진 서버로 사용하여 AWS Elemental MediaPackage MediaTailor로 스트림을 보내기 전에 콘텐츠를 준비하고 광고 마커를 추가합니다. MediaTailor가 오리진 서버와 작동하는 방식에 대한 자세한 내용은 섹션을 참조하세요 [MediaTailor 광고 삽입 작동 방식](#).
- AWS Identity and Access Management (IAM)는 사용자의 AWS 리소스에 대한 액세스를 안전하게 제어하는 데 도움이 되는 웹 서비스입니다. IAM을 사용하여 AWS 리소스를 사용할 수 있는 사용자(인증)와 어떤 방식으로 사용할 수 있는지(권한 부여)를 제어할 수 있습니다. 자세한 내용은 [설 AWS Elemental MediaTailor](#)정 단원을 참조하십시오.

MediaTailor 액세스

서비스의 콘솔을 사용하여 MediaTailor에 액세스할 수 있습니다.

서비스 사용 권한이 있는지 확인하는 자격 증명을 AWS 계정 제공하여에 액세스합니다.

MediaTailor 콘솔에 로그인하려면 링크를 사용합니다 <https://console.aws.amazon.com/mediatailor/home>.

MediaTailor 요금

다른 AWS 제품과 마찬가지로 MediaTailor 사용에 대한 계약이나 최소 약정은 없습니다. 서비스의 사용에 따라 요금이 청구됩니다. 자세한 내용은 [MediaTailor 요금](#)을 참조하세요.

MediaTailor의 리전

애플리케이션의 데이터 지연 시간을 줄이기 위해 MediaTailor는 요청을 위한 리전 엔드포인트를 제공합니다. MediaTailor를 사용할 수 있는 리전 목록을 보려면 [리전 엔드포인트](#)를 참조하세요.

설 AWS Elemental MediaTailor정

이 단원에서는 사용자를 구성하여 AWS Elemental MediaTailor에 액세스하는 데 필요한 단계를 안내합니다. MediaTailor의 자격 증명 및 액세스 관리에 대한 배경 정보 및 추가 정보는 섹션을 참조하세요 [에 대한 자격 증명 및 액세스 관리 AWS Elemental MediaTailor](#).

사용을 시작하려면 다음 단계를 AWS Elemental MediaTailor완료하세요.

주제

- [에 가입 AWS 계정](#)

에 가입 AWS 계정

시작하려면이 AWS필요합니다 AWS 계정. 생성에 대한 자세한 AWS 계정내용은 AWS Account Management 참조 안내서의 [시작하기 AWS 계정](#)를 참조하세요.

MediaTailor 시작하기

MediaTailor를 시작하려면 광고 삽입 설정에 대한 자습서와 채널 어셈블리에 대한 자습서 중 하나를 선택할 수 있습니다. 라이브 이벤트를 위한 실습 워크숍을 탐색할 수도 있습니다. 광고 삽입 자습서에서는 맞춤형 광고를 콘텐츠 스트림에 통합하기 위해 MediaTailor에 재생 요청을 보내는 방법을 안내합니다. 채널 어셈블리 자습서에서는 재생 URL을 사용하여 브라우저에서 직접 개인화된 광고를 포함한 채널의 스트림을 보는 방법을 보여줍니다.

[라이브 이벤트 프레임워크 워크숍](#)을 탐색할 수도 있습니다. 이 실습 워크숍은 초기 검색 및 배포부터 최적화 및 규모 조정까지 AWS Elemental MediaTailor 라이브 이벤트 설정의 전체 수명 주기를 안내합니다. 워크숍을 따라 모든 이벤트 및 채널에 대해 하나의 MediaTailor 설정을 사용하는 간단한 구성 또는 이벤트 또는 채널당 전용 구성을 사용하는 고급 아키텍처를 사용할 수 있습니다.

주제

- [MediaTailor 광고 삽입 시작하기](#)
- [MediaTailor 채널 어셈블리 시작하기](#)

MediaTailor 광고 삽입 시작하기

를 사용하려면 MediaTailor 구성에 액세스하고, 보고, 편집할 수 있는 AWS 계정 및 권한이 AWS Elemental MediaTailor필요합니다. 이렇게 하는 방법에 대한 정보는 [설 AWS Elemental MediaTailor정](#) 단원을 참조하십시오.

이 시작하기 자습서는 다음 작업의 수행 방법을 보여 줍니다.

- HLS 또는 DASH 콘텐츠 스트림 준비
- 광고 결정 서버(ADS) 템플릿 URL 구성
- 재생 엔드포인트가 포함된 MediaTailor 구성 생성
- 플레이어 또는 콘텐츠 전송 네트워크(CDN)를 사용하여 MediaTailor에 재생 요청

동적 변수 및 플레이어 파라미터를 사용한 고급 ADS URL 구성은 [섹션을 참조하세요](#)[ADS 요청에 대한 MediaTailor 동적 광고 변수](#). CDNs[MediaTailor 매니페스트 쿼리 파라미터](#).

완료되면 스트림의 개인화된 광고 콘텐츠에 대한 재생 요청을 MediaTailor로 보낼 수 있습니다.

주제

- [사전 조건](#)
- [1단계: 액세스 AWS Elemental MediaTailor](#)
- [2단계: 스트림 준비](#)
- [3단계: ADS 요청 URL 및 쿼리 파라미터 구성](#)
- [4단계: 구성 생성](#)
- [5단계: 구성 테스트](#)
- [6단계:에 재생 요청 전송 AWS Elemental MediaTailor](#)
- [7단계\(선택 사항\): AWS Elemental MediaTailor 활동 모니터링](#)
- [8단계: 정리](#)

사전 조건

시작하기 전에 먼저 [설 AWS Elemental MediaTailor](#)정의 단계를 완료해야 합니다.

1단계: 액세스 AWS Elemental MediaTailor

IAM 자격 증명을 사용하여에서 MediaTailor 콘솔에 로그인합니다<https://console.aws.amazon.com/mediatailor/home>.

2단계: 스트림 준비

오리진 서버를 구성하여 AWS Elemental MediaTailor과 호환되는 HLS 또는 DASH에 대한 매니페스트를 생성합니다. 오리진 서버는 표준 HTTP 포트(포트 80) 또는 HTTPS 포트(포트 443)를 사용해야 합니다. MediaTailor는 오리진 서버 통신을 위한 사용자 지정 포트를 지원하지 않습니다.

HLS 스트림 준비

HLS 매니페스트는 다음 요구 사항을 충족해야 합니다.

- 매니페스트는 퍼블릭 인터넷에서 액세스할 수 있어야 합니다.
- 매니페스트는 라이브 또는 비디오 온디맨드(VOD)이어야 합니다.
- 매니페스트는 EXT-X-VERSION이 3 이상이어야 합니다.
- 라이브 콘텐츠의 경우 매니페스트는 광고 시간을 규정하기 위해 마커를 포함해야 합니다. 이는 VMAP timeoffsets를 대신 사용할 수 있는 VOD 콘텐츠의 옵션 사항입니다.

매니페스트 파일에는 다음 중 하나로 표시된 광고 슬롯이 있어야 합니다.

- #EXT-X-CUE-OUT / #EXT-X-CUE-IN (일반적임) 기간은 다음 예제에 표시된 바와 같습니다.

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

- #EXT-X-DATERANGE (일반적이지 않음) 기간은 다음 예제에 표시된 바와 같습니다.

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

#EXT-X-DATERANGE에 대해 표시된 모든 필드가 필요합니다.

매니페스트에서 광고 마커를 구성하는 방법은 광고가 스트림에 삽입되는지 또는 스트림에서 다른 조각을 대체하는지 여부에 영향을 줍니다. 자세한 내용은 [the section called “광고 삽입 동작 이해”](#) 단원을 참조하십시오.

- HLS 다변량 재생 목록은 [HTTP 라이브 스트리밍: 마스터 재생 목록 태그](#)에 설명된 HLS 사양을 따라야 합니다. 특히 #EXT-X-STREAM-INF에는 RESOLUTION, BANDWIDTH 및 CODEC 필드가 포함되어야 합니다.

스트림을 구성한 후 다변량 재생 목록의 콘텐츠 오리진 URL 접두사를 기록해 둡니다. 이는 본 자습서에서 나중에 AWS Elemental MediaTailor에서 구성을 생성하는 데 필요합니다.

DASH 스트림 준비

DASH 매니페스트는 다음 요구 사항을 충족해야 합니다.

- 매니페스트는 퍼블릭 인터넷에서 액세스할 수 있어야 합니다.
- 매니페스트는 라이브 또는 비디오 온디맨드(VOD)이어야 합니다.
- 매니페스트는 스폴라이스 삽입 마커 또는 시간 신호 마커를 사용하여 이벤트를 광고 시간으로 표시해야 합니다. 광고 마커를 XML 또는 base64로 인코딩된 이진수로 제공할 수 있습니다. 스폴라이스 삽입의 경우 네트워크 외부 표시기를 활성화해야 합니다. 시간 신호 마커의 경우 분할 UPID 내에 있는 분할 유형 ID에서 인식하는 신호 출력 값이어야 합니다 AWS Elemental MediaTailor. 광고 시간은 이벤트 시작에서 시작하여, 이벤트가 지정된 경우 이벤트 지속 시간 동안, 또는 다음 이벤트가 시작될 때까지 지속됩니다.

다음 예제는 스폴라이스 삽입 마커를 사용하여 광고 시간으로 지정된 이벤트를 표시합니다. 이 광고 시간의 지속 시간은 이벤트의 기간입니다.

```

<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>

```

- 광고 시간에는 콘텐츠 스트림으로 동일한 AdaptationSet 및 Representation 설정이 있어야 합니다. AWS Elemental MediaTailor은 이 둘 사이의 원활한 전환을 위해 이러한 설정을 사용하여 광고를 트랜스코딩하고 콘텐츠 스트림을 일치시킵니다.

스트림을 구성한 후 DAHS 매니페스트에 대한 콘텐츠 오리진 URL 접두사를 기록해 둡니다. 이 자습서 AWS Elemental MediaTailor 뒷부분의에서 구성을 생성하는 데 필요합니다.

3단계: ADS 요청 URL 및 쿼리 파라미터 구성

ADS가 요구하는 쿼리 파라미터를 확인하려면 ADS에서 광고 태그 URL을 생성합니다. 이 URL은 ADS로의 요청에 대한 템플릿의 역할을 하며 다음으로 구성됩니다.

- 정적 값
- 에서 생성된 값 AWS Elemental MediaTailor (session 또는 avail 쿼리 파라미터로 표시됨)
- 플레이어에서 생성하고 클라이언트 애플리케이션에서 가져온 값(player_params. 쿼리 파라미터로 표시됨)

Example ADS에서 광고 태그 URL

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

위치:

- 출력 및 content_id는 정적 값입니다.
- playerSession=[session.id]는에서 제공하는 동적 값입니다 AWS Elemental MediaTailor. [session.id]의 값은 각 플레이어 세션에 대해 변경되고 각 세션의 VAST 요청에 대해 다른 URL을 야 기합니다.
- cust_params는 플레이어가 제공한 동적 값입니다.

플레이어의 다변량 재생 목록 요청은 ADS 요청 URL의 player_params. 쿼리 파라미터에 해당하는 카-값 페어를 제공해야 합니다. 에 대한 요청에서 카-값 페어를 구성하는 방법에 대한 자세한 내용은 섹션을 AWS Elemental MediaTailor참조하세요 [ADS 요청에 대한 MediaTailor 동적 광고 변수](#).

의 MediaTailor에서 오리진 서버/ADS 매핑을 생성할 때 구성된 "템플릿" URL을 입력합니다 [4단계: 구성 생성](#).

테스트

테스트 목적으로 ADS에서 정적 VAST 응답을 사용할 수 있습니다. VAST 응답이 AWS Elemental MediaTailor 에서 트랜스코딩할 수 있는 메자닌 품질 MP4 변환을 반환하는 것이 가장 바람직합니다. ADS의 응답에 여러 재생 변환이 포함된 경우 MediaTailor는 최고 품질의 해상도 MP4 변환을 선택하여 트랜스코더로 전송합니다.

4단계: 구성 생성

AWS Elemental MediaTailor 구성에는 오리진 서버 및 ADS에 대한 매핑 정보가 들어 있습니다.

구성을 만들려면(콘솔)

- <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.
- Configurations(구성) 페이지에서 Create configuration(구성 생성)을 선택합니다.
- 페이지 하단의 구성 단원에서 구성 이름에 구성을 설명하는 고유한 이름을 입력합니다. 이름은 구성의 기본 식별자입니다. 허용된 최대 길이는 512자입니다.

4. 비디오 콘텐츠 소스에이 스트림의 HLS 다변량 재생 목록 또는 DASH 매니페스트의 URL 접두사에서 자산 ID를 뺀 값을 입력합니다. 예를 들어 다변량 재생 목록 URL이 인 경우를 `http://origin-server.com/a/master.m3u8` 입력합니다 `http://origin-server.com/a/`. 또는 `http://origin-server.com`과 같은 더 짧은 접두사를 입력할 수 있지만, 콘텐츠에 대한 플레이어 요청에서 `/a/`를 자산 ID에 포함시켜야 합니다. 최대 길이는 512자입니다.

Note

콘텐츠 오리진이 HTTPS를 사용하는 경우 인증은 잘 알려진 인증 기관에서 받아야 합니다. (자체 서명된 인증서는 사용할 수 없습니다.) 그렇지 않으면 콘텐츠 오리진에 연결 AWS Elemental MediaTailor 하지 못하고 플레이어 요청에 대한 응답으로 매니페스트를 제공할 수 없습니다.

5. 광고 결정 서버에 ADS의 URL을 입력합니다. 이는 [3단계: ADS 요청 URL 및 쿼리 파라미터 구성](#)에 설명된 변수가 있는 URL이거나 테스트 목적으로 사용하는 정적 VAST URL입니다. 최대 길이는 25,000자입니다.

Note

ADS가 HTTPS를 사용하는 경우 인증은 잘 알려진 인증 기관에서 받아야 합니다. (자체 서명된 인증서는 사용할 수 없습니다.) ADS에서 반환되는 메자닌 광고 URL에서도 마찬가지로 MediaTailor가 콘텐츠 오리진에서 매니페스트에 광고를 검색하고 연결하지 못합니다.

6. (DASH의 경우 필요에 따라 선택 사항) MediaTailor 매니페스트에 액세스하기 위해 CDN 라우팅 규칙이 설정되어 있고 클라이언트 측 보고를 사용하거나 플레이어가 고정 HTTP 리디렉션을 지원하는 경우 위치에서 비활성화를 선택합니다.

위치 기능에 대한 자세한 내용은 [the section called “위치 기능”](#) 단원을 참조하십시오.

7. (선택 사항) 오리진 서버에서 단일 기간 DASH 매니페스트를 생성하는 경우 DASH mpd manifest origin type(DASH mpd 매니페스트 오리진 유형)을 선택한 다음 SINGLE_PERIOD를 선택합니다. 기본적으로 MediaTailor는 DASH 매니페스트를 다중 기간 매니페스트로 처리합니다. 자세한 내용은 [the section called “MPEG-DASH 소스 통합”](#) 단원을 참조하십시오.
8. 구성 생성을 선택합니다.

AWS Elemental MediaTailor 는 구성 페이지에 새 구성을 표시합니다.

5단계: 구성 테스트

구성을 저장한 후 스트리밍 프로토콜에 대해 적합한 형식으로 URL을 사용하여 스트림을 테스트합니다.

- 예: HLS

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

- 예: DASH

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

위치:

- playback-endpoint는 구성 생성 시 AWS Elemental MediaTailor 에서 생성한 고유한 재생 엔드 포인트입니다.

예제

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

- hashed-account-id는 사용자의 AWS 계정 ID입니다.

예제

```
777788889999
```

- origin-id는 구성 생성 시 부여한 이름입니다.

예제

```
myOrigin
```

- master.m3u8 또는 manifest.mpd는 테스트 스트림과 파일 확장명에서의 매니페스트 이름입니다. [the section called “4단계: 구성 생성”](#)에서 구성한 비디오 콘텐츠 소스에 이를 추가할 때 전체 식별된 매니페스트를 가져오도록 이를 정의합니다.

이전 예제의 값을 사용하는 전체 URL은 다음과 같습니다.

- 예: HLS

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

- 예: DASH

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

다음 방법 중 하나를 사용하여 스트림을 테스트할 수 있습니다.

- 이전 예제에 표시된 바와 같이 URL을 독립 실행형 플레이어에 입력합니다.
- 자체 플레이어 환경에서 스트림을 테스트합니다.

6단계:에 재생 요청 전송 AWS Elemental MediaTailor

다운스트림 플레이어 또는 CDN을 구성하여 AWS Elemental MediaTailor에서 제공된 구성의 재생 엔드포인트에 재생 요청을 전송합니다. [3단계: ADS 요청 URL 및 쿼리 파라미터 구성](#)에 있는 ADS 요청 URL에서 사용한 플레이어에서 정의된 동적 변수는 플레이어에서의 매니페스트 요청에 정의되어야 합니다.

Example

템플릿 ADS URL이 다음과 같다고 가정합니다.

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

그런 다음 ADS 대신 앞에 없는 파라미터를 ads. 오리진 서버에 ads.. AWS Elemental MediaTailor passs 파라미터로 키-값 페어 앞에 붙여 [player_params.cust_params] 플레이어 요청에를 정의합니다.

플레이어 요청 URL은 다음 HLS 및 DASH 예제의 일부 변형입니다.

```
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
```

가 플레이어 요청을 AWS Elemental MediaTailor 수신하면 요청의 정보를 기반으로 플레이어 변수를 정의합니다. 결과로 초래된 ADS 요청 URL은 이 변수의 일부 변형입니다.

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

ADS로 전달하기 위해 카값 페어를 구성하는 방법에 대한 자세한 내용은 [ADS 요청에 대한 MediaTailor 동적 광고 변수](#) 단원을 참조하십시오.

7단계(선택 사항): AWS Elemental MediaTailor 활동 모니터링

Amazon CloudWatch 및 Amazon CloudWatch Logs를 사용하여 채워진 요청, 오류 및 광고 시간 수와 같은 AWS Elemental MediaTailor 활동을 추적할 수 있습니다.

와 함께 CloudWatch를 처음 사용하는 경우 서비스 간 통신을 허용하는 AWS Identity and Access Management (IAM) 역할을 AWS Elemental MediaTailor 생성합니다.

CloudWatch에 대한 AWS Elemental MediaTailor 액세스를 허용하려면(콘솔)

1. IAM 콘솔(<https://console.aws.amazon.com/iam/>)을 엽니다.
2. IAM 콘솔의 탐색 창에서 역할을 선택하고 역할 생성을 선택합니다.
3. 다른 AWS 계정 역할 유형을 선택합니다.
4. 계정 ID에 AWS 계정 ID를 입력합니다.
5. Require external ID(외부 ID 필요)를 선택하고 **midas**를 입력합니다. 이 옵션은 요청에 올바른 sts:ExternalID가 포함된 경우에만 서비스가 역할을 맡을 수 있도록 허용하는 조건을 신뢰 정책에 자동으로 추가합니다.
6. 다음: 권한을 선택합니다.
7. 역할이 완료할 수 있는 작업을 지정하는 권한 정책을 추가합니다. 다음 옵션 중 하나를 선택한 다음 Next: Review(다음: 검토)를 선택합니다.
 - Amazon CloudWatch Logs에 대한 전체 액세스를 제공하는 CloudWatchLogsFullAccess Amazon CloudWatch
 - Amazon CloudWatch에 대한 전체 액세스를 제공하는 CloudWatchFullAccess Amazon CloudWatch

8. 역할 이름에 **MediaTailorLogger**를 입력한 다음 역할 생성을 선택합니다.
9. 역할 페이지에서 방금 생성한 역할을 선택합니다.
10. 보안 주체를 업데이트하도록 신뢰 관계를 편집합니다.
 1. 역할의 요약 페이지에서 신뢰 관계 탭을 선택합니다.
 2. 신뢰 관계 편집을 선택합니다.
 3. 정책 문서에서 보안 주체부터 AWS Elemental MediaTailor 서비스까지 바꾸어주십시오. 형식은 다음과 같아야 합니다.

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

전체 정책은 다음과 같습니다.

4. 신뢰 정책 업데이트를 선택합니다.

8단계: 정리

관련 없는 요금을 방지하려면 모든 불필요한 구성을 삭제합니다.

구성을 삭제하려면(콘솔)

1. AWS Elemental MediaTailor 구성 페이지에서 다음 중 하나를 수행합니다.
 - 삭제할 구성의 구성 이름을 선택합니다.
 - 구성 이름 옆에서 라디오 버튼을 선택한 다음 삭제를 선택합니다.
2. 구성 삭제 확인 상자에 **Delete**을 입력한 다음 다시 삭제를 선택합니다.

AWS Elemental MediaTailor 는 구성을 제거합니다.

MediaTailor 채널 어셈블리 시작하기

이 시작하기 자습서는 다음 작업의 수행 방법을 보여 줍니다.

- 소스 위치를 생성하고 소스 콘텐츠를 추가합니다.
- 채널 생성

- 일정에 따라 채널의 콘텐츠를 재생할 프로그램 목록 생성
- 광고 삽입을 사용하여 채널 스트림에 개인화된 AWS Elemental MediaTailor 광고 추가

완료되면 브라우저를 열고 채널의 재생 URL을 입력한 다음 개인화된 광고가 포함된 채널의 스트림을 볼 수 있습니다.

이 자습서에서는 MediaTailor 채널 어셈블리를 시작하는 기본 단계를 안내합니다. 자세한 내용은 단원을 참조하십시오 [AWS Elemental MediaTailor 를 사용하여 선형 어셈블 스트림 생성](#).

추정 비용

- 활성 채널에 대한 요금은 시간당 0.10 USD입니다. 비활성 상태인 채널에 대해서는 요금이 부과되지 않습니다.

주제

- [사전 조건](#)
- [1단계: 소스 위치 생성](#)
- [2단계: 소스 위치에 VOD 소스 추가](#)
- [3단계: 채널 생성](#)
- [4단계: 채널 일정에 프로그램 추가](#)
- [5단계\(선택 사항\): MediaTailor를 사용하여 스트림에 개인화된 광고 삽입](#)
- [6단계: 채널 시작](#)
- [7단계: 채널 테스트](#)
- [8단계: 정리](#)

사전 조건

이 자습서를 시작하기 전에 다음 요구 사항을 완료해야 합니다.

- 의 단계를 완료했는지 확인합니다 [설 AWS Elemental MediaTailor정](#).
- VOD 소스 콘텐츠와 광고 슬레이트 모두에 사용할 수 있는 자산이 있어야 합니다. 자산의 매니페스트 경로를 알아야 합니다.

Note

자동 적응형 비트레이트(ABR) 또는 제목당 인코딩을 사용하는 경우 모든 변형의 길이가 동일하고 하위 트랙 수가 동일하도록 자산을 인코딩해야 합니다. 최소 세그먼트 길이가 1초인 인코딩 템플릿을 사용하는 것이 좋습니다.

1단계: 소스 위치 생성

소스 위치는 콘텐츠가 저장되는 오리진 서버를 나타냅니다. Amazon S3, 표준 웹 서버, 콘텐츠 전송 네트워크(CDN) 또는와 같은 패키징 오리진일 수 있습니다 AWS Elemental MediaPackage.

MediaTailor는 소스 위치에서 콘텐츠 매니페스트를 가져와 기본 콘텐츠 세그먼트를 참조하는 라이브 슬라이딩 매니페스트 창을 조합하는 데 사용합니다.

소스 위치를 생성하려면 다음 절차를 수행합니다.

소스 위치를 생성하려면

1. <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.
2. 탐색 창에서 채널 어셈블리 > 소스 위치를 선택합니다.
3. 탐색 모음에서 소스 위치 생성을 선택합니다.
4. 소스 위치 구성에서 소스 콘텐츠의 식별자와 위치를 입력합니다.
 - 이름: my-origin과 같은 소스 위치의 식별자입니다.
 - 기본 URL: <https://111111111111.cloudfront.net> 같이 콘텐츠가 호스팅되는 오리진 서버의 기본 URL입니다. URL은 http:// 또는 https:// 접두사가 붙은 표준 HTTP URL 형식이어야 합니다.
5. 소스 위치 생성을 선택합니다.

2단계: 소스 위치에 VOD 소스 추가

채널에 대해 하나 이상의 소스 위치를 정의했으므로 이제 하나 이상의 VOD 소스를 추가할 수 있습니다. 각 VOD 소스는 단일 영화, TV 프로그램 에피소드 또는 하이라이트 클립과 같은 단일 콘텐츠를 나타냅니다.

VOD 소스에 대해 하나 이상의 패키지 구성을 생성해야 합니다. 각 패키지 구성에는 VOD 소스에 대한 패키지 형식 및 매니페스트 설정이 포함되어 있습니다. 그런 다음 채널에 패키지 구성을 추가하여 출력을 생성합니다.

여러 패키지 구성을 사용하여 다른 채널 출력을 생성할 수 있습니다. 예를 들어 VOD 소스가 HLS와 DASH로 모두 패키징된 경우 각 형식에 대해 두 개의 패키지 구성을 생성할 수 있습니다. 그런 다음 패키지 구성의 소스 그룹을 사용하여 두 개의 채널 출력을 생성할 수 있습니다. 하나는 HLS용이고 다른 하나는 DASH용입니다.

VOD 소스를 추가하고 패키지 구성을 생성하려면

1. <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.
2. 탐색 창에서 채널 어셈블리 > 소스 위치를 선택합니다.
3. 소스 위치 창에서 [소스 위치를 생성하려면](#) 절차에서 생성한 소스 위치를 선택합니다.
4. VOD 소스 추가를 선택합니다.
5. VOD 소스 세부 정보에서 my-example-video와 같은 VOD 소스의 이름을 입력합니다.
6. 패키지 구성 > *source-group-name*에서 패키지 구성에 대한 정보를 입력합니다.

Note

소스의 패키지 구성은 모두 소스 매니페스트에 따라 기간이 동일해야 합니다. 또한 패키지 구성 내의 모든 소스에는 동일한 수의 하위 스트림이 있어야 합니다. 이러한 요구 사항을 충족하려면 자산에 인코딩 템플릿을 사용하는 것이 좋습니다. 최소 세그먼트 길이가 1초 인 인코딩 템플릿을 사용하는 것이 좋습니다. MediaTailor는 제목당 또는 자동 적응형 비트레이트 스트리밍(ABR)을 지원하지 않습니다. 이러한 인코딩 방법은 이러한 요구 사항을 위반하기 때문입니다.

- 소스 그룹: HLS-4k와 같이이 패키지 구성을 설명하는 소스 그룹 이름을 입력합니다. 이 이름을 기록해 둡니다. 채널의 출력을 생성할 때 참조합니다. 자세한 내용은 [채널의 출력과 함께 소스 그룹 사용](#) 단원을 참조하십시오.
 - 유형:이 구성의 패키지 형식을 선택합니다. MediaTailor는 HLS 및 DASH를 지원합니다.
 - 상대 경로: 소스 위치의 기본 HTTP URL에서 매니페스트까지의 상대 경로입니다. 예: /my/path/index.m3u8.
7. 소스 추가를 선택합니다.
 8. 이 절차의 4~7단계를 반복하여 광고 슬레이트의 VOD 소스를 추가합니다.

3단계: 채널 생성

채널은 소스를 라이브 선형 스트림으로 어셈블합니다. 각 채널에는 VOD 소스의 패키지 구성에 해당하는 하나 이상의 출력이 포함되어 있습니다.

먼저 채널을 생성한 다음 프로그램을 생성하여 채널 일정에 VOD 소스를 추가합니다.

채널 생성

1. <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.
 2. 탐색 창에서 채널 어셈블리 > 채널을 선택합니다.
 3. 탐색 모음에서 채널 생성을 선택합니다.
 4. 채널 세부 정보에서 채널에 대한 세부 정보를 입력합니다.
 - 이름: 채널의 이름을 입력합니다.
 - 재생 모드: 허용되는 프로그램 전환 유형과 프로그램이 완료된 후 프로그램에 어떤 일이 발생하는지 결정합니다. 기본 루프 모드를 사용합니다.
 5. 다음을 선택합니다.
 6. 출력 세부 정보에서이 출력에 대한 설정을 정의합니다.
 - 매니페스트 이름: **###**와 같은 매니페스트 이름을 입력합니다. MediaTailor는 HLS용 .m3u8과 같은 형식 확장자를 추가합니다.
-  **Note**

채널 출력당 고유한 매니페스트 이름을 입력해야 합니다.
- 형식 유형: 채널의 스트리밍 형식을 선택합니다. DASH 및 HLS가 지원됩니다. 에서 생성한 패키지 구성에 해당하는 형식을 선택합니다 [1단계: 소스 위치 생성](#).
 - 소스 그룹: 에서 생성한 소스 그룹의 이름을 입력합니다 [1단계: 소스 위치 생성](#).
 7. 매니페스트 설정에서 매니페스트 설정에 대한 추가 정보를 입력합니다.
 - 매니페스트 기간(초): 각 매니페스트에 포함된 기간(초)입니다. 최소값은 30초이고 최대값은 3600초입니다.
 8. 다음을 선택합니다.
 9. 채널 정책에서 채널 정책 연결 안 함을 선택합니다. 이 옵션은 자격 AWS 계정 증명에 액세스할 수 있는 사용자로만 재생을 제한합니다.

10. 다음을 선택합니다.
11. 검토 및 생성 창에서 설정을 검토합니다.
12. Create funnel(퍼널 생성)을 선택합니다.

Note

채널은 중지된 상태로 생성됩니다. 채널을 시작할 때까지 채널이 활성화되지 않습니다.

4단계: 채널 일정에 프로그램 추가

이제 채널이 있으므로 채널 일정에 프로그램을 추가합니다. 각 프로그램에는 계정의 소스 위치에서 온 VOD 소스가 포함되어 있습니다. 채널 일정에 따라 채널 스트림에서 프로그램이 재생되는 순서가 결정됩니다.

각 프로그램에는 하나 이상의 광고 시간이 있을 수 있습니다. 광고 슬레이트로 사용할 VOD 소스를 지정하여 광고 시간을 삽입합니다. 광고 중단 기간은 슬레이트 기간에 따라 결정됩니다. 선택적으로 MediaTailor 광고 삽입과 같은 서버 측 광고 삽입 서버를 사용하여 광고 시간을 개인화할 수 있습니다.

채널을 일정에 추가하려면

1. <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.
2. 탐색 창에서 채널 어셈블리 > 채널을 선택합니다.
3. 채널 창에서 [3단계: 채널 생성](#) 절차에서 생성한 채널을 선택합니다.
4. 프로그램 세부 정보에서 프로그램에 대한 세부 정보를 입력합니다.
 - 이름: 채널 일정에 추가할 프로그램의 이름입니다.
 - 소스 위치 이름: 기존 소스 위치 선택을 선택하고 소스 위치 선택 드롭다운 메뉴에서에서 생성한 [1단계: 소스 위치 생성](#) 소스 위치를 선택합니다.
 - VOD 소스 이름: 기존 VOD 소스 선택을 선택하고이 자습서의 앞부분에서 생성한 VOD 소스를 선택합니다.
5. 재생 구성에서 채널 일정에 프로그램이 삽입되는 방식과 시기를 정의합니다.
 - 전환 유형:이 값은 상대로 고정됩니다. 상대 전환 유형은이 프로그램이 프로그램 목록 내의 다른 프로그램에 비해 발생함을 나타냅니다.

- 상대 위치: 채널 일정의 첫 번째 프로그램인 경우이 설정을 건너뛸 수 있습니다. 채널 일정의 첫 번째 프로그램이 아닌 경우 프로그램 목록에서 프로그램을 추가할 위치를 선택합니다. 프로그램 전 또는 프로그램 후를 선택할 수 있습니다.
- 상대 프로그램: 일정의 첫 번째 프로그램인 경우이 설정을 건너뛸 수 있습니다. 채널 일정의 첫 번째 프로그램이 아닌 경우 기존 프로그램 사용을 선택하고에서 생성한 프로그램 이름을 선택합니다 [채널을 일정에 추가하려면](#).

6.

광고 시간 추가를 선택합니다. 광고 시간에서 광고 시간에 대한 설정을 구성합니다.

- 슬레이트 소스 위치 이름: 기존 소스 위치 선택을 선택하고이 자습서의 앞부분에서 생성한 슬레이트가 저장되는 소스 위치를 선택합니다.
- VOD 소스 이름: 기존 VOD 소스 선택을 선택하고이 자습서의 앞부분에서 추가한 슬레이트에 사용 중인 VOD 소스를 선택합니다. 슬레이트 지속 시간에 따라 광고 중단 지속 시간이 결정됩니다.
- 밀리초 단위 오프셋의 경우:이 값은 프로그램 시작을 기준으로 오프셋으로 광고 시간 시작 시간을 밀리초 단위로 결정합니다. VOD 소스 기간보다 짧고 프로그램의 VOD 소스 내 모든 트랙(모든 오디오, 비디오 및 닫힌 캡션 트랙)의 세그먼트 경계와 일치하는 값을 입력합니다. 그렇지 않으면 광고 시간을 건너뛵니다. 예를 들어 0을 입력하면 프로그램이 시작되기 전에 재생되는 롤링 전 광고 시간이 생성됩니다. 참고: .

7. 프로그램 추가를 선택합니다.

프로그램에 대한 자세한 내용은 단원을 참조하십시오 [Configuring ad breaks for your program](#).

선형 스트림에서 광고를 사용하는 방법에 대한 자세한 내용은 섹션을 참조하세요 [선택적 구성 설정](#).

5단계(선택 사항): MediaTailor를 사용하여 스트림에 개인화된 광고 삽입

이제 프로그램이 포함된 채널이 생겼습니다. 원하는 경우 MediaTailor를 사용하여 채널 스트림의 프로그램 광고 시간에 개인화된 광고를 삽입할 수 있습니다.

사전 조건

계속하기 전에 다음 요구 사항을 충족해야 합니다.

- 광고 결정 서버(ADS)가 있어야 합니다.
- [채널 일정에 프로그램 추가](#) 프로시저에서 광고 중단 설정을 구성해야 합니다.

MediaTailor를 사용하여 채널 스트림에 개인화된 광고를 추가하려면

1. <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.
2. 탐색 창에서 구성을 선택합니다.
3. 필수 설정에서 구성에 대한 기본 필수 정보를 입력합니다.
 - 이름: 구성의 이름입니다.
 - 콘텐츠 소스: 채널 출력의 재생 URL에서 파일 이름과 확장명을 뺀 값을 입력합니다.
MediaTailor 구성에 대한 자세한 내용은 섹션을 참조하세요 [필수 설정](#).
 - 결정 서버 추가: ADS의 URL을 입력합니다.
4. 선택적으로 구성 별칭, 개인화 세부 정보 및 고급 설정을 구성할 수 있습니다. 이러한 설정에 대한 자세한 내용은 단원을 참조하십시오 [선택적 구성 설정](#).
5. 탐색 모음에서 구성 생성을 선택합니다.

MediaTailor 광고 삽입 사용에 대한 자세한 내용은 섹션을 참조하세요 [AWS Elemental MediaTailor 를 사용하여 광고 삽입](#).

6단계: 채널 시작

이제 채널이 생겼습니다. 하지만 채널의 스트림에 액세스하려면 먼저 채널을 시작해야 합니다. 채널이 활성화되기 전에 액세스하려고 하면 MediaTailor는 HTTP 4xx 오류 코드를 반환합니다.

채널 시작

1. <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.
2. 탐색 창에서 채널 어셈블리 > 채널을 선택합니다.
3. 탐색 모음에서 시작을 선택합니다.

7단계: 채널 테스트

채널이 올바르게 작동하는지 확인하려면 웹 브라우저를 열고 채널 출력의 URL을 입력합니다. 채널의 스트림이 표시되어야 합니다.

예상 동작을 확인하기 위해 캐시를 지워야 하는 경우도 있습니다.

8단계: 정리

이 자습서를 위해 생성한 채널을 완료한 후에는 채널을 삭제하여 정리해야 합니다.

채널 상태가 중지됨으로 변경되는 즉시 해당 채널에 대한 요금 발생을 중지합니다. 나중에 채널을 유지할 수 있지만 요금이 발생하지 않도록 채널을 지금 중지한 다음 나중에 다시 시작할 수 있습니다.

채널을 삭제하려면

1. <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.
2. 탐색 창에서 채널 어셈블리 > 채널을 선택합니다.
3. 삭제할 채널을 선택합니다.
4. 채널이 실행 중인 경우 작업 드롭다운 메뉴에서 중지를 선택합니다. 채널을 삭제하려면 먼저 채널을 중지해야 합니다.
5. 채널이 중지되면 작업 드롭다운 메뉴에서 삭제를 선택합니다.

AWS Elemental MediaTailor 를 사용하여 광고 삽입

구성은 AWS Elemental MediaTailor에서 상호 작용하는 객체입니다. 구성에는 오리진 서버 및 광고 결정 서버(ADS)에 대한 매핑 정보가 들어 있습니다. 또한 광고를 사용할 수 없거나 전체 광고 시간을 채우지 못할 때 MediaTailor에서 사용할 기본 재생을 정의할 수 있습니다.

MediaTailor에서 콘텐츠 전송 네트워크(CDN)를 사용하는 경우 구성에 CDN 정보를 추가하기 전에 CDN에서 동작 규칙을 설정해야 합니다. CDN 설정에 대한 자세한 내용은 [CDN을 사용하여 MediaTailor 광고 개인화 및 콘텐츠 전송 최적화](#) 단원을 참조하십시오.

주제

- [지원되는 오디오 및 비디오 코덱](#)
- [AWS Elemental MediaTailor 광고 삽입 동작 이해](#)
- [MediaTailor 서버 가이드 광고 삽입 개요 및 구현](#)
- [MediaTailor 광고 서버 통합 요구 사항](#)
- [광고 부패](#)
- [MediaTailor 재생 구성 관리](#)
- [MediaTailor 광고 삽입을 위한 콘텐츠 소스 통합](#)
- [Google의 광고 플랫폼 AWS Elemental MediaTailor 과 통합](#)
- [광고 중단 억제를 사용하여 광고 중단 동작 사용자 지정](#)
- [MediaTailor 범퍼 광고 삽입](#)
- [MediaTailor 프리롤 광고 삽입](#)
- [MediaTailor 슬레이트 광고 삽입](#)
- [의 광고 트랜스코딩 AWS Elemental MediaTailor](#)
- [광고 미리 가져오기](#)
- [에서 사전 조정된 광고 사용 AWS Elemental MediaTailor](#)
- [MediaTailor 수익화 함수 작업](#)
- [MediaTailor 수율 최적화 작업](#)
- [ADS 요청에 대한 MediaTailor 동적 광고 변수](#)
- [MediaTailor 매니페스트 쿼리 파라미터](#)
- [광고 추적 데이터 보고](#)
- [오버레이 광고](#)

- [광고 ID 장식](#)

지원되는 오디오 및 비디오 코덱

MediaTailor는 다음 코덱을 지원합니다.

- 오디오 코덱: mp4a, ac-3 및 ec-3
- 비디오 코덱: h.264 (AVC), h.265 (HEVC), av01 (AV1)

AWS Elemental MediaTailor 광고 삽입 동작 이해

AWS Elemental MediaTailor 는 오리진 매니페스트에 광고를 교체하거나 삽입하여 라이브 또는 온디맨드 비디오(VOD) 콘텐츠에 광고를 연결합니다. 광고 삽입 또는 교체 여부는 오리진 매니페스트에서 광고 중단이 구성되는 방식과 콘텐츠가 VOD인지 라이브인지에 따라 달라집니다. 광고 시간은 프로그래밍 중 광고가 표시되는 기간이며, 광고 시간은 광고로 채워질 수 있는 광고 시간 내의 특정 광고 시간 단위입니다.

- MediaTailor는 광고 대체를 통해 콘텐츠 세그먼트를 광고로 대체합니다.
- 광고 삽입을 사용하면 MediaTailor는 세그먼트가 없는 광고 콘텐츠를 삽입합니다.

MediaTailor가 광고를 라이브 및 VOD 콘텐츠에 연결하는 방법에 대한 자세한 내용은 해당 주제를 선택합니다.

주제

- [VOD에 대한 광고 스티칭 동작](#)
- [라이브 광고 스티칭 동작](#)

VOD에 대한 광고 스티칭 동작

MediaTailor는 오리진 매니페스트에서 광고 마커가 구성되는 방식과 광고 결정 서버(ADS)가 VMAP 응답을 전송하는지 여부에 따라 VOD 콘텐츠에 광고를 삽입하거나 교체합니다.

마커 구성에 의한 광고 방식은 다음 단원을 참조하십시오.

광고 마커가 있는 경우

AWS Elemental MediaTailor 는 오리진 매니페스트에 SCTE-35 광고 마커가 있는 광고를 삽입합니다. 0 지속 시간 EXT-X-CUE-OUT 값이 있는 광고 마커는 광고 삽입을 나타냅니다.

HLS 광고 마커 지침

롤 후 및 광고 포드 SCTE 신호 전송에 대한 다음 지침을 따릅니다.

롤링 전 광고

HLS 포스트롤의 경우 CUE-OUT/IN 마커는 마지막 콘텐츠 세그먼트에 선행해야 합니다. 이는 HLS 사양에 따라 태그 데코레이터가 세그먼트 앞에 명시적으로 선언되어야 하기 때문입니다.

예를 들어 다음 선언을 고려해 보십시오.

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

AWS Elemental MediaTailor 는 다음과 같이 사후 롤을 삽입합니다.

```
#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: 광고 포드

CUE-OUT/IN 태그는 세그먼트에 명시적으로 연결되어야 합니다. 여러 CUE-OUT/IN 태그를 연속으로 사용하여 Ad Pod 방식을 모방할 수 없습니다.

예를 들어 다음 선언은를 사용하여 광고 포드CUE-OUT/IN를 표현하는 데 유효합니다.

```
#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
```

앞의 선언을 사용하면 다음과 같은 출력이 나옵니다.

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

다음 선언은 유효하지 않습니다.

```
#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
```

광고 마커가 없는 경우

광고 마커는 매니페스트에서 광고 시간을 알리는 데 권장되는 방법입니다. 그러나 광고 마커는 필요하지 않습니다. 매니페스트에 DASH 또는 HLS에 대한 광고 마커가 포함되어 있지 않은 경우 MediaTailor는 ADS에 대한 단일 호출을 수행하고 응답을 기반으로 광고 시간을 생성합니다.

- ADS가 VAST 응답을 전송하는 경우 MediaTailor는 매니페스트 시작 시 광고 시간에 응답의 모든 광고를 삽입합니다. 프리롤 광고입니다.

- ADS가 VMAP 응답을 전송하는 경우 MediaTailor는 광고 중단 시간 오프셋을 사용하여 중단 시간을 생성하고 지정된 시간(프리롤, 미드롤 또는 포스트롤)에 매니페스트 전체에 삽입합니다. MediaTailor는 매니페스트의 각 광고 시간에 대해 VMAP 응답의 각 광고 시간의 모든 광고를 사용합니다.

Note

세그먼트가 VOD 콘텐츠에 대한 VMAP와 삽입 지점을 겹치면 MediaTailor는 가장 가까운 삽입 지점으로 내림합니다.

Tip

미드롤 광고 시간을 생성하려는 경우 ADS가 VMAP를 지원하지 않는 경우 매니페스트에 광고 마커가 있는지 확인합니다. MediaTailor는 다음 섹션에 설명된 대로 마커에 광고를 삽입합니다.

Note

서버 가이드 광고 삽입 방법의 경우 MediaTailor는 매니페스트 상단에 프리롤 광고를 삽입하고 플레이어는 다른 광고 유형보다 먼저 광고를 재생합니다.

라이브 광고 스티칭 동작

라이브 스트림에서는 AWS Elemental MediaTailor 항상 광고 교체를 수행하여 광고 마커 사이의 총 시간을 최대한 가깝게 유지합니다. 광고 마커에 DURATION 속성이 포함된 경우 MediaTailor는 값을 사용하여 광고 중단 기간을 결정합니다. 모든 CUE-OUT 표시기에는 라이브 워크플로에 지속 시간 또는 일치하는 CUE-IN 표시기가 있어야 합니다.

MediaTailor는 HLS 및 DASH 라이브 콘텐츠에 대한 광고 교체를 수행합니다. MediaTailor가 광고 시간 배치 및 타이밍을 계산하는 방법에 대한 자세한 내용은 [the section called “광고 마커”](#) 및 섹션을 참조하세요 [the section called “광고 마커”](#).

광고 선택 및 교체

AWS Elemental MediaTailor에는 다음과 같이 광고 결정 서버(ADS) VAST 응답의 광고가 포함됩니다.

- 기간을 지정하면 MediaTailor는 기간에 맞는 광고 세트를 선택하고 이를 포함합니다.

- 기간을 지정하지 않으면 MediaTailor는 기본 콘텐츠로 돌아가는 광고 마커가 나타날 때까지 최대한 많은 광고를 재생합니다.

AWS Elemental MediaTailor 는 라이브 광고 교체 중에 다음 지침을 준수합니다.

- MediaTailor는 클리핑이나 잘림 없이 완전한 광고를 재생하려고 시도합니다.
- MediaTailor가 광고 시간의 끝을 나타내는 광고 마커를 발견하면 기본 콘텐츠로 돌아갑니다. 이는 현재 재생 중인 광고가 잘리는 것을 의미합니다.
- 기간이 끝나면 MediaTailor는 기본 콘텐츠로 돌아갑니다.
- MediaTailor가 광고 중단 기간 동안 재생할 광고가 부족한 경우 구성된 경우 슬레이트를 재생하거나 기본 콘텐츠 스트림의 재생을 재개합니다. 이는 대개 트랜스코딩된 광고가 광고 지속 시간을 채우기에 충분치 않은 경우에 발생합니다.

Tip

개인화 임계값 구성 설정을 사용하여 광고 시간에 허용되는 채워지지 않은 광고 시간 제한을 정의할 수 있습니다. 자세한 내용은 [PlaybackConfiguration 참조](#)를 참조하십시오.

서버 가이드 광고 삽입을 위한 라이브 프리롤

라이브 프리롤은 서버 측 광고 삽입과 비교하여 서버 안내 광고 삽입 방법에서 다르게 작동합니다.

서버 측 광고 삽입(스티치 모드)

프리롤 광고는 각 시청자 세션 시작 시 라이브 콘텐츠의 일부를 대체합니다. 각 뷰어는 스트림에 조인하는 시점에 따라 서로 다른 시간에 프리롤을 볼 수 있습니다.

서버 가이드 광고 삽입 방법

MediaTailor는 CUE="PRE,ONCE" 속성이 있는 모든 미디어 매니페스트 상단에 프리롤 날짜 범위 태그를 배치합니다. 이로 인해 플레이어는 개인화되지 않은 동일한 매니페스트를 공유하더라도 재생 시작 시 프리롤 광고를 한 번 요청하고 재생합니다.

구성 요구 사항:

- 라이브 프리롤 광고 결정 서버: 프리롤 광고를 위한 VAST 엔드포인트 구성(미드롤 광고와 다를 수 있음)

- 라이브 프리롤 최대 허용 기간: 프리롤 광고의 최대 기간을 설정합니다(선택 사항 - 생략한 경우 반환된 모든 광고가 사용됨).

기술 구현:

- 프리롤 날짜 범위 태그 사용 `START-DATE="1970-01-01T00:00:00.000Z"` (Unix epoch)
- 프리롤에 대한 자산 목록 요청은 일반 ADS 대신 구성된 프리롤 광고 결정 서버를 사용합니다.
- 플레이어는 자산 목록 데이터의 `availId="aws-mediatailor-preroll-1"`를 통해 프리롤 요청을 식별합니다.

Important

라이브 스트림의 경우 프리롤 광고는 콘텐츠를 지연시키는 대신 콘텐츠를 포함합니다. 향후 버전은 추가 구성 옵션을 통해 콘텐츠 지연 모드를 지원할 수 있습니다.

Note

프리롤 동작은 서버 가이드 광고 삽입을 위한 라이브 콘텐츠와 VOD 콘텐츠에 따라 다릅니다. 라이브 콘텐츠에는 명시적 프리롤 구성이 필요한 반면, VOD 콘텐츠에는 기본적으로 일반 광고 결정 서버를 사용한 프리롤이 포함됩니다.

예제

- 광고 시간의 지속 시간이 70초로 설정되어 있고 ADS 응답에 2개의 40초 광고가 포함된 경우는 40초 광고 중 하나를 AWS Elemental MediaTailor 재생합니다. 남은 시간에는 구성된 슬레이트나 기본 콘텐츠로 전환됩니다. 이 프로세스 중 언제든지 MediaTailor에서 신호 표시기가 발생하면 기본 콘텐츠로 즉시 잘립니다.
- 광고 시간의 지속 시간이 30초로 설정되어 있고 ADS 응답에서 제공하는 가장 짧은 광고가 40초인 경우 MediaTailor는 광고를 재생하지 않습니다. 광고 슬레이트가 구성된 경우 MediaTailor는 30초 동안 또는 신호 표시기가 나타날 때까지 이를 재생합니다. 그렇지 않으면 MediaTailor가 기본 콘텐츠를 재생합니다.

MediaTailor 서버 가이드 광고 삽입 개요 및 구현

AWS Elemental MediaTailor 서버 가이드 광고 삽입(SGAI)은 광고를 미디어 재생 목록에 직접 스티칭하는 대신 별도의 재생 목록으로 참조하여 서버 측 광고 삽입에 대한 대안을 제공합니다. 이 접근 방식은 캐시 가능한 매니페스트를 통해 성능을 개선하고 확장성을 높입니다.

MediaTailor에서 서버 가이드 광고 삽입을 사용하는 방법에 대한 자세한 내용은 다음 목록에서 해당 주제를 선택합니다.

재생 구성에서 활성화

플레이어가 서버 가이드 광고 삽입을 사용하도록 허용하려면 MediaTailor 재생 구성PLAYER_SELECT에서 Insertion Mode로 설정해야 합니다. 이렇게 하면 플레이어가 세션 초기화 시 스티칭된 광고 삽입 또는 안내된 광고 삽입을 선택할 수 있습니다.

서버 가이드 세션 생성

재생 세션을 생성할 때 안내 모드를 선택합니다. 이를 수행하는 방법은 플레이어가 암시적 세션을 사용하는지 명시적 세션을 사용하는지에 따라 달라집니다.

암시적으로 생성된 서버 가이드 세션

HLS 다변량 재생 목록 요청에 `aws.insertionMode=GUIDED`를 추가합니다. 예제:

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

위치:

- `playback-endpoint`는 구성 생성 시 AWS Elemental MediaTailor 에서 생성한 고유한 재생 엔드 포인트입니다.

예제

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

- `hashed-account-id`는 사용자의 AWS 계정 ID입니다.

예제

```
777788889999
```

- origin-id는 구성 생성 시 부여한 이름입니다.

예제

```
myOrigin
```

- index.m3u8 또는는 테스트 스트림의 매니페스트 이름과 파일 확장명입니다. [the section called “4 단계: 구성 생성”](#)에서 구성한 비디오 콘텐츠 소스에 이를 추가할 때 전체 식별된 매니페스트를 가져 오도록 이를 정의합니다.

이전 예제의 값을 사용하는 전체 URL은 다음과 같습니다.

- 예제:

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

명시적으로 생성된 서버 가이드 세션

플레이어가 HTTP에서 MediaTailor 구성의 세션 초기화 접두사 엔드포인트POST로 보내는 insertionMode=GUIDED JSON 메타데이터에 추가합니다.

다음 예제에서는 JSON 메타데이터의 구조를 보여줍니다.

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

이 초기화 메타데이터를 사용하면 재생 세션은 server-guided 광고 삽입을 사용합니다.

SGAI를 사용한 광고 추적

SGAI는 서버 측 및 클라이언트 측 광고 추적을 모두 지원합니다. 세션 초기화 시 보고 모드를 설정합니다. 세션 중에는 모드를 변경할 수 없습니다.

서버 측 추적(기본값)

MediaTailor는 플레이어가 광고 세그먼트를 요청할 때 VAST 비컨을 자동으로 실행합니다. 자산 목록의 광고 URIs에는 암호화된 비컨 메타데이터(awsBeaconData, awsBeaconDomain, awsConfigurationName)가 포함됩니다. 플레이어는 HLS #EXT-X-DEFINE:QUERYPARAM 변수 대체를 지원해야 합니다. 자산 목록 응답에는 TRACKING 섹션이 포함되지 않습니다.

SGAI 서버 측 비컨 작동 방식에 대한 자세한 내용은 섹션을 참조하세요 [서버 가이드 광고 삽입 \(SGAI\)을 사용한 서버 측 추적](#).

클라이언트 측 추적

세션 초기화 요청에 aws.reportingMode=CLIENT를 추가합니다. 자산 목록 응답에는 광고 재생 중에 플레이어가 실행하는 비컨 URLs이 있는 TRACKING 섹션이 포함됩니다. GetTracking API 엔드포인트는 SGAi 세션에 사용되지 않습니다. 대신 각 자산 목록 응답에는 직접 데이터 추적이 포함됩니다. 추적 데이터는 서버 측 광고 삽입(SSAI) 추적 응답과 동일한 JSON 스키마를 사용합니다.

자세한 내용은 [서버 가이드 광고 삽입](#)을 참조하세요.

매니페스트 하트비트를 사용한 안내 미리 가져오기

라이브 SGAi 스트림의 경우 세션 초기화 요청에를 추가하여 매니페스트 기반 광고 미리 가져오기를 활성화aws.guidedPrefetchMode=MANIFEST할 수 있습니다.

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

활성화되면 MediaTailor는 다중 변형 재생 목록의 각 중간 미디어 매니페스트(?aws.sessionId=<id>) URL에 세션 식별자(/v1/i-media)를 쿼리 파라미터로 추가합니다. 플레이어가 i-media 매니페스트를 새로 고칠 때마다 요청은 세션 ID로 MediaTailor에 도달하며, MediaTailor는 이를 사용하여 예정된 광고 시간에 대한 미리 가져오기 요청을 대기열에 넣습니다.

- aws.guidedPrefetchMode 파라미터는 MANIFEST (활성화됨) 및 OFF (비활성화됨, 기본값)의 두 값을 허용합니다.
- 안내 미리 가져오기 모드는 SGAi 세션에만 유효합니다. 연결된 세션과 함께 사용하면 오류가 반환됩니다.
- DASH는 아직 안내 미리 가져오기 모드를 지원하지 않습니다.
- 안내 미리 가져오기는 보고 모드와 독립적입니다. 비컨은 미리 가져오기 시간이 아니라 재생 시간에 실행됩니다.

- 안내 미리 가져오기를 사용할 때 CDN에 i-media 매니페스트를 캐싱하지 마세요. 미리 가져오기 메커니즘은 MediaTailor에 직접 도달하는 플레이어의 매니페스트 새로 고침 요청에 따라 달라집니다. CDN이 /v1/i-media 응답을 캐싱하는 경우 MediaTailor는 하트비트 요청을 수신하지 않으며 미리 가져오기를 트리거할 수 없습니다.

MediaTailor 서버 가이드 광고 삽입 기능 호환성 매트릭스

AWS Elemental MediaTailor 는 기능 호환성이 서로 다른 두 가지 광고 삽입 방법을 제공합니다. 서버 가이드 광고 삽입은 서버 측 광고 삽입과 다르게 작동하여 일부 MediaTailor 기능과의 호환성에 영향을 미칩니다. 이 표를 사용하여 각 광고 삽입 방법에서 작동하는 기능을 이해할 수 있습니다.

광고 삽입 방법별 기능 호환성

기능	서버 측 광고 삽입 (SSAI)	서버 가이드 광고 삽입 (SGAI)
광고 미리 가져오기	✓ 지원	아직 지원되지 않음
광고 억제	✓ 지원	해당 사항 없음
프리롤 광고 동작	MediaTailor 구성에서 제어	MediaTailor 구성에서 제어
클라이언트 측 광고 추적	GetTracking API 사용	자산 목록의 사용 TRACKING 섹션 (GetTracking API 는 사용되지 않음)
서버 측 광고 추적	✓ 지원 - 세션 ID를 사용한 /v1/segment 요청에 따라 비컨이 실행됩니다.	✓ 지원(HLS만 해당) -를 통해 광고 URIs에 포함된 암호화된 비컨 데이터와 함께 세션 없는 비컨을 사용합니다#EXT-X-DEFINE:QUERYPARAM . HLS v11 이상이 필요합니다.

기능	서버 측 광고 삽입 (SSAI)	서버 가이드 광고 삽입 (SGAI)
		DASH는 아직 지원되지 않습니다.
광고 ID 장식	✓ 지원	" 호환되지 않음

호환성 세부 정보

광고 미리 가져오기

광고 미리 가져오기는 현재 지원되지 않습니다.

광고 억제

플레이어는 재생할 광고만 가져오기 때문에 서버 가이드 광고 삽입 방법에서는 광고 억제가 지원되지 않습니다.

프리롤 광고 동작

프리롤 광고 타이밍은 삽입 방법 간에 다르게 작동합니다.

- 서버 측 광고 삽입: MediaTailor는 구성 설정에 따라 프리롤 광고가 재생되는 시기를 제어합니다.
- 서버 안내 광고 삽입: MediaTailor는 매니페스트 상단에 프리롤 광고를 삽입합니다. 플레이어가 이러한 광고를 먼저 표시한 다음 콘텐츠를 시작합니다.

광고 추적

클라이언트 측 추적은 광고 삽입 방법에 따라 다양한 메커니즘을 사용합니다.

- 서버 측 광고 삽입(SSAI): GetTracking API 엔드포인트를 사용합니다.
- 서버 가이드 광고 삽입(SGAI): MediaTailor는 각 자산 목록 응답의 TRACKING 섹션에 추적 정보를 제공합니다. GetTracking API 엔드포인트는 사용되지 않습니다. 세션 초기화 응답에는가 포함되지 않습니다trackingUrl.

서버 측 추적도 메서드마다 다릅니다.

- 서버 측 광고 삽입(SSAI): MediaTailor는 플레이어가 세션 ID를 /v1/segment/ 사용하여 스티칭된 광고 세그먼트를 가져올 때 비컨을 실행합니다.
- 서버 안내 광고 삽입(SGAI): MediaTailor는 세션 없는 비컨을 사용합니다. MediaTailor는 자산 목록에서 반환하는 광고 매니페스트 URIs에 암호화된 비컨 데이터(awsBeaconData, awsBeaconDomain, awsConfigurationName)를 포함합니다. 광고 매니페스트는 #EXT-X-DEFINE:QUERYPARAM 태그를 사용하므로 플레이어가 이러한 값을 세그먼트 URLs. 플레이어가 각 광고 세그먼트를 요청하면 MediaTailor는 데이터를 해독하고 적절한 비컨을 실행하며 콘텐츠 세그먼트로 리디렉션합니다. 서버 측 보고가 활성화되면 MediaTailor는 자산 목록 응답에서 TRACKING 섹션을 생략합니다. 자세한 내용은 [서버 가이드 광고 삽입\(SGAI\)을 사용한 서버 측 추적을 참조하세요](#).

광고 ID 방식

X-AD-CREATIVE-SIGNALING 헤더를 채우는 필드는 매니페스트가 작성될 때가 아니라 자산 목록을 가져올 때만 알려져 있기 때문에 Ad-ID 방식은 서버 가이드 광고 삽입과 호환되지 않습니다.

라이브 스트림에 대한 MediaTailor 서버 가이드 광고 삽입 구성

AWS Elemental MediaTailor 라이브 콘텐츠에 대한 서버 가이드 광고 삽입은 캐시 가능한 매니페스트를 통해 상당한 성능 이점을 제공합니다. 라이브 콘텐츠에 대한 SGAI 구성은 라이브 스트림 특성 및 실시간 처리에 대한 특정 고려 사항과 함께 VOD와 동일한 코어 파라미터를 사용합니다.

라이브 SGAI 요구 사항

라이브 콘텐츠에 대해 SGAI를 활성화하기 전에 다음이 있는지 확인합니다.

- 라이브 스트림에 올바른 형식의 DATERANGE 마커 포함
- 광고 중단 기간은 일관되고 예측 가능합니다.
- CDN이 SGAI 매니페스트를 적절하게 캐싱하도록 구성됨
- 플레이어가 서버 가이드 광고 삽입 워크플로 지원
- 광고 결정 서버가 라이브 콘텐츠에 대한 실시간 요청을 처리할 수 있음

플레이어 요구 사항

SGAI 라이브 매니페스트를 올바르게 처리하도록 플레이어를 구성해야 합니다.

- 서버 가이드 광고 삽입 워크플로 지원
- 매니페스트에서 광고 삽입 지침을 처리하는 기능

- 라이브 스트림 타이밍 및 동기화의 적절한 처리
- HLS 콘텐츠의 경우: CLASS 속성이 있는 HLS 버전 8 및 EXT-X-DATERANGE를 지원합니다. 서버 측 비컨을 위한 버전 11.
- HLS 콘텐츠의 경우: EXT-X-DEFINE 변수 대체 지원

라이브 재생 구성

라이브 콘텐츠에 대해 SGAI를 활성화하려면 다음 설정이 있는 재생 구성을 생성합니다.

Example SGAI 라이브 재생 구성

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

다음은 라이브 SGAI 구성의 주요 고려 사항입니다.

VideoContentSourceUrl

올바른 형식의 SCTE-35 DATERANGE 마커가 있는 라이브 HLS 스트림을 가리켜야 합니다. 스트림은 일관된 세그먼트 기간과 비트 전송률 변형을 유지해야 합니다.

SGAI 라이브 매니페스트 요청

SGAI 라이브 매니페스트는 기존 광고 삽입과 동일한 URL 패턴을 사용합니다.

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

라이브 SGAI에 대한 매니페스트 기반 미리 가져오기

라이브 SGAI 워크플로의 경우 매니페스트 기반 미리 가져오기 하트비트를 활성화하여 광고 채우기 지연 시간을 줄일 수 있습니다. 매니페스트 요청에 `aws.guidedPrefetchMode=MANIFEST`를 추가합니다.

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

활성화되면 MediaTailor는 다중 변형 재생 목록의 각 중간 미디어 매니페스트(?
`aws.sessionId=<id>`) URL에 세션 식별자(/v1/i-media)를 쿼리 파라미터로 추가합니다. 플레이

어가 i-media 매니페스트를 새로 고칠 때마다 요청은 세션 ID로 MediaTailor에 도달하며, MediaTailor는 이를 사용하여 세션을 식별하고 예정된 광고 시간에 대한 미리 가져오기 요청을 대기열에 넣습니다.

Important

안내 미리 가져오기를 사용할 때 CDN에 i-media 매니페스트를 캐싱하지 마세요. 미리 가져오기 하트비트 메커니즘은 MediaTailor에 직접 도달하는 플레이어의 매니페스트 새로 고침 요청에 따라 달라집니다. CDN이 캐시하고 /v1/i-media 응답을 제공하는 경우 MediaTailor는 하트비트 요청을 수신하지 않으며 미리 가져오기를 트리거할 수 없습니다. 가 사용 중일 때 MediaTailor에 /v1/i-media/* 요청을 전달하도록 `CDNaws.guidedPrefetchMode=MANIFEST`을 구성합니다.

안내 미리 가져오기는 보고 모드와 독립적입니다. 서버 측(기본값) 또는 클라이언트 측(`aws.reportingMode=CLIENT`) 추적을 사용하든, 비컨은 광고를 미리 가져올 때가 아니라 재생 시 실행됩니다. MediaTailor에서 광고 미리 가져오기가 작동하는 방식에 대한 일반적인 내용은 섹션을 참조하세요 [광고 미리 가져오기](#).

SGAI 라이브 구성 테스트

다음 검증 단계를 통해 SGAI 라이브 설정을 확인합니다.

1. 매니페스트 생성 테스트

SGAI 라이브 매니페스트 URL을 요청하고 적절한 광고 삽입 지침에 따라 캐시 가능한 콘텐츠를 반환하는지 확인합니다.

2. CDN 캐싱 확인

CDN이 구성된 TTL 값에 따라 SGAI 매니페스트를 캐싱하고 있는지 확인합니다.

3. 광고 삽입 테스트

플레이어가 SGAI 매니페스트에 제공된 지침에 따라 광고를 성공적으로 삽입할 수 있는지 확인합니다.

4. 성능 모니터링

CloudWatch 지표를 사용하여 오리진 로드 감소 및 캐시 적중률 개선을 확인합니다.

VOD 콘텐츠에 대한 MediaTailor 서버 가이드 광고 삽입 구성

AWS Elemental MediaTailor VOD 콘텐츠에 대한 서버 가이드 광고 삽입은 캐시 가능성이 높은 매니페스트와 서버 처리 감소를 통해 상당한 성능 이점을 제공합니다. VOD 콘텐츠용 SGAI를 구성하면 video-on-demand 자산의 정적 특성을 활용하여 캐싱 효율성을 극대화하고 오리진 요청을 최소화하므로 반복되는 보기 패턴이 있는 대규모 콘텐츠 라이브러리에 적합합니다.

VOD SGAI 요구 사항

VOD 콘텐츠에 대해 SGAI를 활성화하기 전에 다음이 있는지 확인합니다.

- VOD 콘텐츠에는 적절한 형식의 광고 마커(SCTE-35 또는 시간 지정 메타데이터)가 포함됩니다.
- 콘텐츠는 일관된 가용성으로 안정적인 오리진에 저장됩니다.
- CDN이 적절한 TTL 값으로 SGAI 매니페스트를 캐싱하도록 구성됨
- 플레이어가 서버 가이드 광고 삽입 워크플로 지원
- 광고 결정 서버는 VOD별 메타데이터 및 대상 지정을 처리할 수 있습니다.

플레이어 요구 사항

플레이어는 SGAI VOD 매니페스트 및 광고 삽입을 처리하도록 구성되어야 합니다.

- 서버 가이드 광고 삽입 워크플로 지원
- VOD 매니페스트의 광고 삽입 지침을 처리하는 기능
- VOD 재생 중 클라이언트 측 광고 삽입 지원
- 광고 시간 동안 탐색 작업의 적절한 처리
- 콘텐츠 기간 및 위치 추적 지원

VOD 재생 구성

VOD 콘텐츠에 대해 SGAI를 활성화하려면 다음 설정이 있는 재생 구성을 생성합니다.

Example SGAI VOD 재생 구성

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

다음은 VOD SGAI 구성의 주요 고려 사항입니다.

VideoContentSourceUrl

일관된 URL 패턴으로 VOD 콘텐츠 라이브러리를 가리켜야 합니다. 오리진이 예상 요청 볼륨을 처리하고 안정적인 콘텐츠 전송을 제공할 수 있는지 확인합니다.

ConfigurationAliases

매니페스트 캐시 가능성에 영향을 주지 않고 광고 대상 지정에 사용할 수 있는 콘텐츠 기간, 장르 또는 시리즈 정보와 같은 VOD 관련 파라미터를 포함합니다.

ManifestProcessingRules

광고 마커 패스스루를 활성화하여 원래 콘텐츠 타이밍 정보를 보존합니다. 이는 광고 시간 위치가 미리 정의된 VOD 콘텐츠에 특히 중요합니다.

SGAI VOD 매니페스트 요청

SGAI VOD 매니페스트는 동일한 URL 패턴의 기존 VOD 광고 삽입을 사용합니다.

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

VOD별 광고 타겟팅

VOD 콘텐츠는 다음과 같은 고유한 광고 타겟팅 기회를 제공합니다.

콘텐츠 메타데이터 대상 지정

VOD 콘텐츠 메타데이터를 활용하여 광고 타겟팅을 개선합니다.

- 장르 및 범주: 콘텐츠 유형(드라마, 코미디, 다큐멘터리)에 따른 대상 광고
- 콘텐츠 등급: 연령에 적합한 광고 콘텐츠 보장(G, PG, R 등급)
- 시리즈 및 시즌: 시리즈 연속성 또는 관련 콘텐츠에 대한 대상 광고
- 릴리스 날짜: 콘텐츠 수명을 기준으로 한 대상(새 릴리스 대 카탈로그 콘텐츠)
- 콘텐츠 기간: 총 콘텐츠 길이를 기준으로 광고 로드 조정

컨텍스트 타겟팅 보기

VOD 보기 패턴은 컨텍스트 광고 타겟팅을 활성화합니다.

- 시간대: 콘텐츠를 시청하는 시점을 기준으로 광고 대상 지정
- 폭주 시청: 여러 에피소드를 시청하는 사용자의 광고 빈도 조정
- 완료율: 사용자의 과거 콘텐츠 완료 패턴을 기반으로 하는 대상
- 디바이스 유형: 디바이스 보기를 위한 광고 형식 최적화(TV, 모바일, 태블릿)

SGAI VOD 구성 테스트

다음 검증 단계를 사용하여 SGAI VOD 설정을 확인합니다.

1. 매니페스트 생성 테스트

다양한 콘텐츠 유형에 대해 SGAI VOD 매니페스트 URLs 요청하고 적절한 광고 삽입 지침에 따라 캐시 가능한 콘텐츠를 반환하는지 확인합니다.

2. CDN 캐싱 확인

CDN이 구성된 TTL 값에 따라 SGAI 매니페스트를 캐싱하고 높은 캐시 적중률을 달성하고 있는지 확인합니다.

3. 광고 삽입 테스트

플레이어가 다양한 VOD 콘텐츠에 대해 SGAI 매니페스트에 제공된 지침을 사용하여 광고를 성공적으로 삽입할 수 있는지 확인합니다.

4. 검색 작업 테스트

VOD 콘텐츠 내에서 검색이 광고 시간 동안 올바르게 작동하고 적절한 재생 위치를 유지하는지 확인합니다.

5. 성능 모니터링

CloudWatch 지표를 사용하여 오리진 로드 감소, 캐시 적중률 개선, 광고 삽입률 성공을 확인할 수 있습니다.

주요 테스트 시나리오

다음과 같은 특정 VOD 시나리오를 테스트합니다.

- 인기 콘텐츠: 자주 액세스하는 VOD 자산의 높은 캐시 적중률 확인
- 롱폼 콘텐츠: 여러 광고 시간이 있는 영화 또는 긴 에피소드에서 광고 삽입 테스트
- 시리즈 콘텐츠: 시리즈의 에피소드 간에 일관된 광고 타겟팅 확인
- 다양한 장르: 콘텐츠 메타데이터 및 장르를 기반으로 광고 타겟팅 테스트

VOD SGAI 최적화 모범 사례

최대 성능을 위해 SGAI VOD 구현을 최적화합니다.

캐시 최적화

- TTL 값 최대화: 콘텐츠가 변경되지 않으므로 VOD 매니페스트에 더 긴 캐시 기간 사용
- 캐시 키 최소화: 캐시 키 변형을 줄여 적중률 개선
- 인기 콘텐츠 사전 워밍: 추세 또는 주요 VOD 콘텐츠에 대한 캐시 매니페스트
- 캐시 성능 모니터링: 캐시 적중률을 추적하고 사용 패턴에 따라 최적화

콘텐츠 전송 최적화

- 일관된 URL 패턴: 더 나은 캐싱을 위해 예측 가능한 URL 구조 사용
- 메타데이터 표준화: 안정적인 광고 타겟팅을 위한 일관된 콘텐츠 메타데이터 보장

- 광고 시간 위치 지정: 자연스러운 콘텐츠 전환을 위한 광고 시간 배치 최적화
- 품질 변형: SGAI가 VOD 콘텐츠의 모든 비트레이트 변형에서 작동하는지 확인합니다.

MediaTailor 광고 서버 통합 요구 사항

광고 서버를와 통합하려면 AWS Elemental MediaTailor 광고 서버가 지원되는 VAST 및 VMAP 버전의 IAB 사양을 준수하는 XML을 전송해야 합니다. 퍼블릭 VAST 검사기를 사용하여 태그 형식이 올바른지 확인합니다.

AWS Elemental MediaTailor 는 광고 결정 서버의 VAST 및 VMAP 응답을 지원합니다. AWS Elemental MediaTailor 또한 클라이언트 측 광고 삽입을 위한 클라이언트 측 보고 API를 통해 VPAID 메타데이터의 프록시를 지원합니다. 클라이언트 측 보고에 대한 자세한 내용은 [클라이언트 측 광고 추적](#) 단원을 참조하십시오.

MediaTailor는 다음 버전의 VAST, VMAP 및 VPAID를 지원합니다.

- 최대 [VAST 4.3](#)

MediaTailor는 VAST 4.3을 통해 응답 버전을 수락하지만 VAST 4.0 이상의 일부 고급 기능은 지원되지 않습니다.

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

VAST 요구 사항

광고 서버의 VAST 응답에는 IAB 호환 TrackingEvents 요소 및 표준 이벤트 유형(예: impression)이 포함되어야 합니다. 비표준 추적 이벤트를 포함하는 경우는 VAST 응답을 AWS Elemental MediaTailor 거부하고 가용성에 대한 광고를 제공하지 않습니다.

VAST 3.0은 순차적 선형 광고 세트를 전송하는 Ad Pods에 대한 지원 기능을 도입했습니다. 광고 포드의 특정 광고를 사용할 수 없는 경우는 ADS의 상호 작용 로그에 CloudWatch에 AWS Elemental MediaTailor 오류를 기록합니다. 그런 다음 Ad Pod에 다음 광고를 삽입하려고 시도합니다. 이러한 방식으로 MediaTailor는 사용할 수 있는 광고를 찾을 때까지 포드의 광고를 반복합니다.

Targeting

특정 플레이어의 광고 타겟을 설정하기 위해 광고 태그 및 URL의 템플릿을 생성할 수 있습니다. 자세한 내용은 [ADS 요청에 대한 MediaTailor 동적 광고 변수](#) 단원을 참조하십시오.

AWS Elemental MediaTailor 는 광고 서버 VAST 요청을 보낼 때와 서버 측 추적 호출을 수행할 때 플레이어의 user-agent 및 x-forwarded-for 헤더를 프록시합니다. 광고 서버가 이러한 헤더를 처리할 수 있는지 확인하십시오. 그렇지 않으면 [session.user_agent] 또는 [session.client_ip]를 사용하여 광고 태그 및 광고 URL의 쿼리 문자열에 이 값을 전달할 수 있습니다. 자세한 내용은 [ADS 요청에 대한 MediaTailor 세션 변수](#) 단원을 참조하십시오.

광고 호출

AWS Elemental MediaTailor 는 구성에 정의된 대로 VAST 광고 URL을 호출합니다. 광고 호출 시 플레이어별 또는 세션별 파라미터를 대체합니다. MediaTailor는 VAST 응답에서 최대 7개의 VAST 래퍼 및 리디렉션 수준을 따릅니다. 라이브 스트리밍 시나리오에서 MediaTailor는 연결된 플레이어의 광고 가능 시간 시작 시 동시에 광고 호출을 수행합니다. 실제로 이러한 광고 호출은 지터 때문에 몇 초에 걸쳐 분산될 수 있습니다. 광고 서버가 이 유형의 호출에 필요한 동시 연결 수를 처리할 수 있는지 확인합니다. MediaTailor는 라이브 워크플로에 대한 VAST 응답 미리 가져오기를 지원합니다. 자세한 내용은 [광고 미리 가져오기](#) 단원을 참조하십시오.

크리에이티브 처리

가 ADS VAST 응답을 AWS Elemental MediaTailor 수신하면 각 크리에이티브에 대해 트랜스코딩에 MediaFile 가장 높은 비트레이트를 식별하고 이를 소스로 사용합니다. 플레이어의 다변량 재생 목록 비트레이트 및 해상도에 맞는 변환으로 변환하기 위해 파일을 on-the-fly 트랜스코더로 전송합니다. 최상의 결과를 얻으려면 최고 비트레이트 미디어 파일이 유효한 매니페스트 프리셋을 갖춘 고품질 MP4 자산인지 확인하십시오. 매니페스트 프리셋이 유효하지 않을 경우 트랜스코드 작업이 실패하여 광고가 표시되지 않습니다. 유효하지 않은 프리셋의 예에는 지원되지 않는 입력 파일 형식(예: ProRes)과 특정한 변환 사양(예: 해상도 855X481)이 포함됩니다.

미디어 파일 입력에 지원되는 형식 목록은 AWS Elemental MediaConvert 사용 설명서의 [지원되는 입력 형식](#)의 MP4 행을 참조하세요.

크리에이티브 인덱싱

AWS Elemental MediaTailor 는 <Creative> 요소에 제공된 id 속성의 값을 기준으로 각 크리에이티브를 고유하게 인덱싱합니다. 크리에이티브의 ID가 지정되지 않은 경우 MediaTailor는 인덱스에 미디어 파일 URL을 사용합니다.

다음 예제 선언에서는 크리에이티브 ID를 보여줍니다.

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

고유한 크리에이티브 ID를 정의할 경우 각 크리에이티브에 대해 고유한 새 ID를 사용합니다. 크리에이티브 IDs 재사용하지 마세요.는 크리에이티브 콘텐츠를 반복적으로 사용하기 위해 AWS Elemental MediaTailor 저장하고 인덱싱된 ID로 각각 찾습니다. 새 크리에이티브가 들어오면 서비스는 먼저 ID를 인덱스와 대조합니다. ID가 있는 경우 MediaTailor는 수신 콘텐츠를 재처리하는 대신 저장된 콘텐츠를 사용합니다. 크리에이티브 ID를 재사용하면 MediaTailor는 오래된 저장된 광고를 사용하며 새 광고를 재생하지 않습니다.

광고 서비스 파트너가 제공하는 VAST 확장

크리에이티브 IDs와의 충돌을 방지하기 위해 광고 서비스 파트너가 VAST 응답에 제공하는 확장을 사용할 수 있습니다. MediaTailor는 SpringServe, Publica 및 FreeWheel의 확장을 지원합니다. VAST 확장 재정의의 활성화하면 MediaTailor가 기본 크리에이티브 ID를 확장 값으로 바꿉니다.

이 기능을 활성화하려면 [AWS 지원 티켓을 제출하여](#) VAST 확장 기반 크리에이티브 IDs 활성화를 요청합니다. 지원 티켓에 다음 정보를 포함합니다.

- AWS 리전
- AWS 계정 ID
- MediaTailor 재생 구성 이름

계정에서 VAST 확장 기반 크리에이티브 IDs 활성화되어 있는지 확인하려면 스테이징 또는 테스트 재생 구성에서도 RAW_ADS_RESPONSE 로깅을 활성화하도록 요청하는 것이 좋습니다. 로깅을 사용하면 ADS가 수신하는 원래 VAST 응답을 보고 올바른 크리에이티브 IDs가 사용되는지 확인할 수 있습니다.

VPAID 요구 사항

게시자는 VPAID를 통해 고도의 대화형 비디오 광고를 제공하고 수익 흐름에 대한 가시적인 지표를 제공할 수 있습니다. VPAID에 대한 자세한 내용은 [VPAID 사양](#)을 참조하세요.

AWS Elemental MediaTailor 는 server-side-stitched VAST MP4 선형 광고와 client-side-inserted VPAID 대화형 크리에이티브를 동일한 광고 공간에서 혼합하여 지원합니다. VAST 응답에 나타나는 순서를 유지합니다. MediaTailor는 최대 7개 수준의 래퍼를 통해 VPAID 리디렉션을 따릅니다. 클라이언트 측 보고 응답에는 언래핑된 VPAID 메타데이터가 포함됩니다.

VPAID를 사용하려면 다음 지침을 따릅니다.

- VPAID 크리에이티브에 대해 MP4 슬레이트를 구성합니다.는 구성된 슬레이트로 VPAID 광고 슬롯을 AWS Elemental MediaTailor 채우고 클라이언트 플레이어는 대화형 광고를 실행하는 데 사

용할 VPAID 광고 메타데이터를 제공합니다. 슬레이트가 구성되지 않은 경우 VPAID 광고가 나타나면 MediaTailor는 평소와 같이 클라이언트 측 보고를 통해 광고 메타데이터를 제공합니다. 또한 CloudWatch에 누락된 슬레이트에 대한 오류를 기록합니다. 자세한 내용은 [MediaTailor 슬레이트 광고 삽입](#) 및 [MediaTailor 재생 구성 생성](#) 섹션을 참조하세요.

- 클라이언트 측 보고를 사용합니다.는 클라이언트 측 보고 API를 통해 VPAID를 AWS Elemental MediaTailor 지원합니다. 자세한 내용은 [클라이언트 측 광고 추적](#) 단원을 참조하십시오.

이론적으로는 기본 서버 측 보고 모드를 VPAID와 함께 사용할 수 있습니다. 그러나 서버 측 보고를 사용할 경우 VPAID 광고의 존재와 주변의 메타데이터에 대한 정보가 손실됩니다. 그 이유는 클라이언트 측 API를 통해서만 사용할 수 있기 때문입니다.

- 라이브 시나리오에서는 로 표시된 광고 시간이 VPAID에서 사용자 상호 작용을 수용할 수 있을 만큼 충분히 긴EXT-X-CUE-OUT: Duration지 확인합니다. 예를 들어 VAST XML이 30초 길이의 VPAID 광고를 지정하는 경우 광고 시간을 30초 이상으로 구성하는 것이 좋습니다. 이 추가 시간은 사용자에게 광고와 상호 작용할 수 있는 더 많은 기회를 제공합니다. 시간을 추가하지 않으면 광고 시간의 남은 지속 시간이 VPAID 광고를 수용할 만큼 충분히 길지 않기 때문에 VPAID 메타데이터가 손실될 수 있습니다.

광고 부페

Interactive Advertising Bureau (IAB) Video Ad Serving Template (VAST) 사양은 광고 포드와 광고 부페를 정의합니다. 이 메커니즘은 광고 삽입 실패 시 자동 대체와 함께 정렬된 광고 삽입을 지원합니다. 광고 포드에서 시퀀스화된 광고는 순서대로 재생됩니다. 독립 실행형 광고는 시퀀스 광고가 실패할 때 MediaTailor가 대체 광고를 선택할 수 있는 부페 역할을 합니다.

AWS Elemental MediaTailor 는 AdDecisionServerConfiguration의에 있는 AdSequencingMode 설정을 통해이 사양을 지원합니다PlaybackConfiguration. 로 설정하면 FOLLOW_AD_SEQUENCE MediaTailor는 순서대로 시퀀스화된 광고를 삽입하고 시퀀스화된 광고가 삽입되지 않을 때만 독립 실행형 광고를 대체물로 사용합니다.

VAST 응답 예

다음 예제는 시퀀스 광고와 독립 실행형 광고가 포함된 VAST 응답을 보여줍니다.

```
<VAST>
  <Ad sequence="1" id="sequencedAd1">...</Ad>      <!-- Sequenced Ad 1 -->
  <Ad sequence="2" id="sequencedAd2">...</Ad>      <!-- Sequenced Ad 2 -->
```

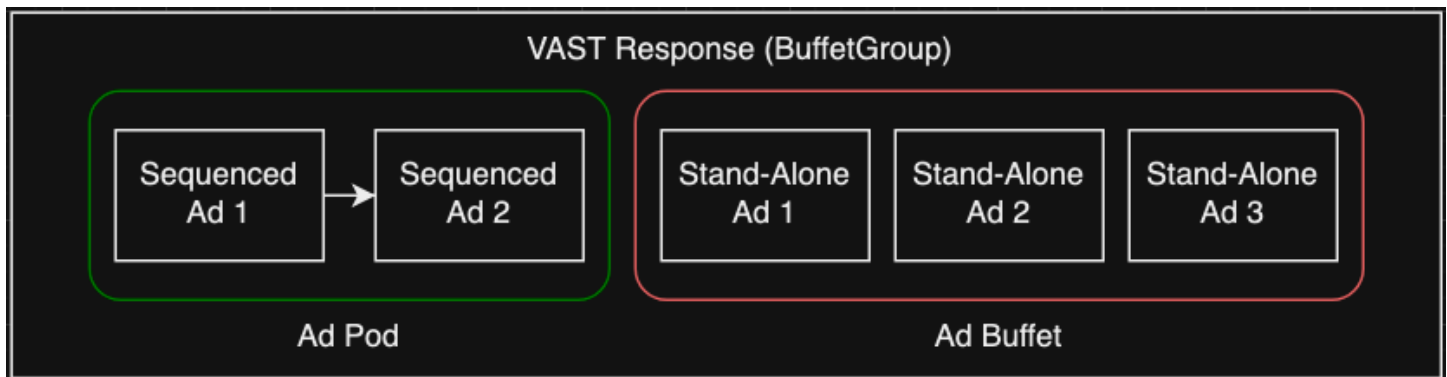
```

<Ad id="standAloneAd1">...</Ad>
<Ad id="standAloneAd2">...</Ad>
<Ad id="standAloneAd3">...</Ad>
</VAST>

```

이 예제에서는 두 개의 시퀀싱된 광고가 광고 포드를 형성합니다. 세 개의 독립 실행형 광고는 광고 부페를 형성합니다. 이들은 함께 하나의 부페 그룹을 구성합니다. sequencedAd1 실패하면 MediaTailor는 먼저 이를 동일한 부페 그룹의 적격 독립 실행형 광고로 교체하려고 시도합니다. 해당 그룹에서 사용할 수 있는 적격 독립 실행형 광고가 없는 경우 MediaTailor는 다른 부페 그룹의 독립 실행형 광고 중에서 선택합니다.

다음 다이어그램은 광고 포드(시퀀스 광고)와 함께 부페 그룹을 구성하는 광고 부페(독립형 광고)가 있는 VAST 응답을 보여줍니다.



주요 개념

다음 용어는 광고 부페의 핵심 개념을 설명합니다.

- 시퀀스 광고 - <Ad> 요소에 sequence 속성이 있는 광고입니다. 이는 정렬된 광고 포드를 형성하며 증가하는 시퀀스 순서로 삽입됩니다.
- 독립 실행형(버펫) 광고 - sequence 속성이 없는 광고입니다. 이는 시퀀스 광고가 삽입되지 않을 때만 대체물로 사용됩니다.
- Vacuum Group - MediaTailor는 단일 VAST 응답 내에서 광고 포드를 독립 실행형 광고와 페어링합니다. 시퀀싱된 광고가 래퍼인 경우 리디렉션된 VAST는 자체 독립 부페 그룹을 구성합니다. 실패한 시퀀스 광고를 대체할 때 MediaTailor는 동일한 부페 그룹의 독립 실행형 광고의 우선 순위를 지정합니다. 동일한 부페 그룹에 적격 독립 실행형 광고가 없는 경우 MediaTailor는 다른 부페 그룹의 독립 실행형 광고로 돌아갑니다.

Note

sequence는 <Ad> 요소에 있는 속성이 아닌 <Creative> 요소 수준 sequence 속성을 참조합니다.

광고 부페의 이점

광고 부페가 없으면 MediaTailor는 모든 광고를 동일하게 취급하므로 다음과 같은 문제가 발생할 수 있습니다.

- VOD - 장애가 발생하지 않더라도 독립 실행형 광고가 삽입되어 합의된 광고 중단 위치가 중단되고 광고 중단이 과도하게 채워집니다.
- 라이브 - 낮은 수익의 독립 실행형 광고는 시퀀스 광고 전에 삽입할 수 있으며 가용 시간을 차지하므로 수익이 높은 시퀀스 광고를 위한 공간이 없습니다.

광고 부페가 활성화된 경우 MediaTailor는 시퀀스화된 광고만 삽입하고 시퀀스화된 광고가 삽입되지 않을 때 독립 실행형 광고를 대체물로만 사용합니다. 이렇게 하면 수익이 가장 높은 광고가 먼저 재생됩니다.

Warning

광고 결정 서버(ADS) 응답에 시퀀스가 지정된 광고가 포함되지 않은 경우(sequence속성이 있는 <Ad> 요소가 없음) 광고 부페를 활성화하면 광고 삽입이 발생하지 않습니다.

광고 부페 활성화

광고 부페는 PutPlaybackConfiguration API를 사용하여 각 재생 구성에 대해 구성하는 옵션인 기능입니다.

AdSequencingMode 값

다음 표에서는 사용 가능한 AdSequencingMode 값을 설명합니다.

값	설명
IGNORE_AD_SEQUENCE (기본값)	MediaTailor는 시퀀스 속성에 관계없이 VAST 응답에 표시되는 순서대로 광고를 삽입합니다.
FOLLOW_AD_SEQUENCE	MediaTailor는 라이브 및 VOD 워크플로 모두에 대해 시퀀스 광고를 삽입합니다. 실패한 시퀀스 광고는 독립 실행형 광고로 대체됩니다.
FOLLOW_AD_SEQUENCE _ONLY_LIVE	광고 부페 동작은 라이브 워크플로에서만 활성화됩니다.
FOLLOW_AD_SEQUENCE _ONLY_VOD	광고 부페 동작은 VOD 워크플로에서만 활성화됩니다.

Note

광고 부페를 활성화하면 MediaTailor는 `Wrapper.fallbackOnNoAd` 및 `Wrapper.allowMultipleAds` 속성을 준수합니다.

미드롤 구성

다음 예제에서는 라이브 워크플로와 VOD 워크플로 모두에 대한 광고 부페를 활성화합니다.

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

다음 예제에서는 라이브 워크플로에 대해서만 광고 부페를 활성화합니다.

```
{
  "PlaybackConfiguration": {
```

```

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

```

다음 예제에서는 VOD 워크플로에 대해서만 광고 부폐를 활성화합니다.

```

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

```

라이브 프리롤 구성

다음 예시에서는 라이브 프리롤 광고에 대한 광고 부폐를 활성화합니다.

```

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

```

Note

LivePrerollConfiguration는 FOLLOW_AD_SEQUENCE 및 만 지원하지
다 IGNORE_AD_SEQUENCE. 미드를 구성을 통해 VOD 프리롤에 대한 광고 부폐를 구성합니다.

광고 부패 작동 방식

광고 대체 조건

시퀀스 광고에 다음 실패 사례 중 하나가 발생하면 MediaTailor는 시퀀스 광고를 동일한 부패 그룹의 적격 독립 실행형 광고로 교체하려고 시도합니다. 동일한 부패 그룹에서 독립 실행형 광고를 사용할 수 없는 경우 MediaTailor는 가능한 경우 다른 부패 그룹의 독립 실행형 광고를 사용합니다.

실패 사례	설명
VAST 구문 분석 실패	시퀀스 광고는 필수 VAST 요소가 누락되어 구문 분석에 실패합니다 (예: , Creatives 또는 MediaFiles Impression 가 누락되었거나 MediaFile 또는 VASTAdTagURI 가 유효하지 않음).
VAST 래퍼 해상도 실패	리디렉션 시간이 초과되거나 실패하거나 빈 응답을 반환합니다.
VPAID 광고 삭제됨	시퀀스 광고는 VPAID이고 세션은 서버 측 보고를 사용하거나 슬레이트가 구성되지 않았습니다.
광고 트랜스코딩 준비되지 않음	시퀀스 광고의 트랜스코딩 상태는 성공이 아닙니다(예: IN_PROGRESS 또는 ERROR).
가용 기간 초과	시퀀스 광고의 지속 시간이 나머지 가용 지속 시간(실시간 워크플로)을 초과합니다.

독립 실행형 광고 자격

교체 자격을 갖추려면 독립 실행형 광고가 다음 기준을 충족해야 합니다.

- VPAID 광고가 아님
- 트랜스코딩 완료
- 남은 가용 기간 내에 맞는 기간이 있어야 합니다.

독립 실행형 광고 선택

시퀀스 광고가 실패하면 MediaTailor는 다음 우선 순위를 사용하여 적격 독립 실행형 광고에서 대체를 선택합니다.

1. 동일한 부패 그룹 먼저 - MediaTailor는 실패한 광고의 자체 부패 그룹의 독립 실행형 광고를 선호합니다.
2. 기타 부패 그룹 - 동일한 부패 그룹에 적격 광고가 없는 경우에만 사용됩니다.

적합한 독립 실행형 광고를 찾을 수 없는 경우 해당 위치가 채워지지 않고 적절한 SkippedReason이 보내집니다.

VAST 래퍼 동작

MediaTailor의 래퍼 리디렉션 응답 처리는 두 가지 래퍼 변수인 `Wrapper.fallbackOnNoAd` 및의 값에 따라 달라집니다 `Wrapper.allowMultipleAds`.

- `fallbackOnNoAd`는 리디렉션 응답이 비어 있거나, 시간이 초과되었거나, ADS 요청 중에 오류가 발생할 때 적용됩니다.
- `allowMultipleAds`는 ADS 응답에 하나 이상의 광고가 포함된 경우 적용됩니다.

VAST 비우기, 요청 오류 또는 요청 제한 시간

다음 표에서는 리디렉션 응답이 비어 있거나, 시간이 초과되었거나, 오류가 발생할 때의 MediaTailor 동작을 설명합니다.

fallbackOnNoAd	MediaTailor 동작
true(기본값)	MediaTailor는 실패한 래퍼를 대체할 독립 실행형 광고를 선택하며, 동일한 부패 그룹을 선호합니다. 사용 가능한 동일한 그룹 광고가 없는 경우 MediaTailor는 다른 부패 그룹 중에서 선택합니다.
false	MediaTailor는 광고 위치를 삭제하고 포드의 다음 광고로 이동합니다. 대체가 발생하지 않습니다.

Example `fallbackOnNoAd=true`(리디렉션 실패)

상위 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>

```

래퍼 리디렉션 응답: VAST 비우기, 시간 초과 또는 오류.

해결된 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>

```

재생 순서:

1. seqAd1
2. standAloneAd1

Example fallbackOnNoAd=false(리디렉션 실패)

상위 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>

```

래퍼 리디렉션 응답: VAST 비우기, 시간 초과 또는 오류.

해결된 VAST:

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

재생 순서:

1. seqAd1

응답에 하나 이상의 광고 포함

다음 표에서는 리디렉션 응답에 하나 이상의 광고가 포함된 경우 MediaTailor 동작을 설명합니다.

allowMultipleAds	MediaTailor 동작
false(기본값)	MediaTailor는 리디렉션 응답에서 하나의 광고를 선택하고 나머지 광고를 삭제합니다. 선택한 광고는 상위 래퍼 광고에서 시퀀스 값을 상속합니다. 선택 전략: MediaTailor는 리디렉션 응답에서 가장 낮은 순서의 광고를 선택합니다. 사용 가능한 순차적 광고가 없는 경우 MediaTailor는 가능한 경우 독립 실행형 광고를 선택합니다.
true	래퍼에 시퀀스 속성이 있는 경우 MediaTailor는 래퍼의 위치에 리디렉션 광고를 삽입합니다. 래퍼에 시퀀스 속성이 없는 경우 MediaTailor는 상위의 시퀀스 광고 뒤에 리디렉션 광고를 삽입합니다. 해결된 광고는 자체 부패 그룹을 형성합니다.

Example allowMultipleAds=false, 래퍼의 시퀀스=2

상위 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>
```

래퍼 리디렉션 응답:

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

해결된 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>
```

재생 순서:

1. seqAd1
2. redirectSeqAd1
3. seqAd3

Example allowMultipleAds=false, 래퍼에 시퀀스 없음

상위 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>
```

래퍼 리디렉션 응답:

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

해결된 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>
```

재생 순서:

1. seqAd1
2. seqAd3

Note

redirectSeqAd1에는 시퀀스가 정의되지 않았으므로 재생되지 않습니다.

Example allowMultipleAds=true, 래퍼의 시퀀스=2

상위 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>

```

래퍼 리디렉션 응답:

```

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

```

해결된 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>

```

재생 순서:

1. seqAd1
2. redirectSeqAd1
3. redirectSeqAd2
4. seqAd3

Example allowMultipleAds=true, 래퍼에 시퀀스가 없음

상위 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>
```

래퍼 리디렉션 응답:

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

해결된 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>
```

재생 순서:

1. seqAd1
2. seqAd3
3. redirectSeqAd1
4. redirectSeqAd2

지표 및 로그 모니터링

기존 지표에 대한 변경 사항

광고 부폐가 활성화되면 다음과 같은 기존 지표의 동작이 변경됩니다.

지표	광고 부폐를 사용한 동작
AdDecisionServer.Ads	ADS 응답에서 구문 분석된 시퀀스 광고 수입니다.
AdDecisionServer.Duration	순서가 지정된 광고의 총 기간입니다.
AdDecisionServer.FillRate	시퀀스 광고만 사용하여 계산된 채우기 속도입니다.

새로운 지표

광고 부폐가 활성화되면 다음과 같은 새 지표를 사용할 수 있습니다.

지표	설명
AdNotReady.SequencedAd	삽입할 준비가 되지 않은 시퀀스 광고 수입니다.
AdNotReady.StandaloneAd	삽입할 준비가 되지 않은 독립 실행형 광고 수(로 건너뛴 광고 제외StandaloneAdNotNeeded).
Avail.FillRate.SequencedAd	시퀀스 광고로 채워진 가용성의 백분율입니다.
Avail.FillRate.StandaloneAd	독립 실행형 광고로 채워진 가용성의 백분율입니다.
Avail.FilledDuration.SequencedAd	시퀀스 광고로 채워진 가용 기간입니다.

지표	설명
Avail.FilledDuration.StandaloneAd	독립 실행형 광고로 채워진 가용 기간입니다.
AdsBilled.SequenceAd	순서가 지정된 청구된 광고 수입입니다.
AdsBilled.StandaloneAd	독립 실행형인 청구된 광고 수입입니다.
SkippedReason.StandaloneAdNotNeeded	순서가 지정된 모든 광고가 성공적으로 삽입되었기 때문에 독립 실행형 광고가 필요하지 않을 때 발생합니다.

새로 건너뛴 광고 이유

광고 부폐가 활성화된 경우 다음과 같은 새로운 건너뛴 광고 이유를 사용할 수 있습니다.

이유	설명
STANDALONE_AD_NOT_NEEDED	독립 실행형 광고는 적격했지만 시퀀스가 지정된 모든 광고가 성공적으로 삽입되었기 때문에 필요하지 않았습니다. 오류 비컨이 실행되지 않습니다.

삽입할 수 없는 독립 실행형 광고(예: VPAID 또는 트랜스코딩 준비되지 않음)의 경우 실제 실패 이유(예: TRANSCODE_IN_PROGRESS 또는 VPAID_AD)가 로 내보내집니다SkippedReason.

오류 비컨

MediaTailor는 다음과 같이 광고 부폐에 대한 오류 비컨을 처리합니다.

- 실패한 순차적 광고 - 오류 비컨이 실행됩니다(기존 동작).
- 준비되지 않은 독립 실행형 광고(광고 트랜스코딩 상태가 성공하지 않음) - 오류 비컨이 실행됩니다(시퀀싱된 광고와 동일).
- 삽입할 수 있지만 필요하지 않은 미사용 독립 실행형 광고 - 오류 비컨이 실행되지 않습니다.

ADS 이벤트 변경

광고 부페가 활성화되면 MediaTailor는 다음 필드를 두 개의 ADS 이벤트에 추가합니다.

VAST_RESPONSE 이벤트

VAST_RESPONSE 이벤트 VastAd의 각 에는 다음 필드가 포함됩니다.

Field	유형	설명
buffetGroup	문자열	MediaTailor에서 생성한 문자열로, 광고가 속한 부페 그룹을 정의합니다.
sequence	정수	Ad 요소의 시퀀스 값입니다.

FILLED_AVAIL 이벤트

creativeAds 및 FILLED_AVAIL 이벤트skippedAds의 각 항목에는 다음 필드가 포함됩니다.

Field	유형	설명
buffetGroup	문자열	MediaTailor에서 생성한 문자열로, 광고가 속한 부페 그룹을 정의합니다.
sequence	정수	Ad 요소의 시퀀스 값입니다.

이러한 필드를 통해 다음을 확인할 수 있습니다.

- 순서와 독립 실행형으로 구문 분석된 광고
- 삽입된 광고 및 유형
- 순서가 지정된 광고를 건너뛰는 이유와 독립 실행형 광고가 이를 대체했는지 여부
- 독립 실행형 광고를 사용할 수 없는 이유

다음 사항도 참조하세요.

PutPlaybackConfiguration API 및 IAB VAST 사양에 대한 자세한 내용은 다음을 참조하세요.

- [PutPlaybackConfiguration API 참조](#) - 구성 AdSequencingMode 및 관련 설정을 위한 전체 파라미터 참조입니다.
- [IAB VAST 4.x Specification](#) IAB Tech Lab 웹 사이트 - 광고 포드, 시퀀스 광고 및 독립 실행형 광고를 정의하는 업계 표준입니다.
- [Amazon CloudWatch 지표 AWS Elemental MediaTailor 를 사용한 모니터링](#) - 새 광고 부페 지표를 포함하여 모든 MediaTailor CloudWatch 지표에 대한 전체 참조입니다.

MediaTailor 재생 구성 관리

이 섹션에서는 AWS Elemental MediaTailor 재생 구성을 관리하기 위한 주요 작업을 다룹니다. 새 구성을 생성하여 콘텐츠 스트림을 설정하고 재생 디바이스에 대한 액세스 권한을 제공하고, 기존 구성의 세부 정보를 보고, 구성을 편집하여 오리진 서버 및 광고 결정 서버와 같은 설정을 업데이트하고, 더 이상 필요하지 않은 구성을 삭제하는 방법을 배울 수 있습니다.

주제

- [MediaTailor 재생 구성 생성](#)
- [MediaTailor 구성 세부 정보 보기](#)
- [MediaTailor 구성 설정 편집](#)
- [MediaTailor 구성 삭제](#)

MediaTailor 재생 구성 생성

이 주제에서는 콘텐츠 스트림 수신을 시작하는 구성을 생성하는 방법을 보여줍니다 AWS Elemental MediaTailor. 또한 다운스트림 재생 디바이스가 콘텐츠를 요청할 수 있는 액세스 포인트를 제공하는 방법도 보여줍니다.

AWS Elemental MediaTailor 콘솔, AWS Command Line Interface (AWS CLI)> 또는 MediaTailor API를 사용하여 구성을 생성할 수 있습니다. AWS CLI 또는 MediaTailor API를 통해 구성을 생성하는 방법에 대한 자세한 내용은 [AWS Elemental MediaTailor API 참조](#)를 참조하세요.

구성을 생성할 때 구성 이름 필드와 같은 자유 형식 필드에 민감한 식별 정보를 입력하지 마십시오. 식별 정보에는 고객 계정 번호와 같은 것이 포함될 수 있습니다. 또한 MediaTailor 콘솔, REST API, AWS CLI 또는 AWS SDKs에서 작업할 때 식별 정보를 사용하지 마십시오. MediaTailor에 입력하는 모든 데이터는 진단 로그 또는 Amazon CloudWatch Events에 포함되도록 선택될 수 있습니다.

구성(콘솔)을 추가하려면

1. <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.
2. Configurations(구성) 페이지에서 Create configuration(구성 생성)을 선택합니다.
3. 다음 주제에 설명한 대로 구성 및 추가 구성 필드를 완료하십시오.

- [필수 설정](#)
- [선택적 구성 설정](#)

4. 구성 생성을 선택합니다.

AWS Elemental MediaTailor 는 구성 페이지의 테이블에 새 구성을 표시합니다.

5. (권장) 매니페스트 및 보고 요청에 AWS Elemental MediaTailor 대해 사용하여 CDN을 설정합니다. 구성 재생 URLs CDN 설정에 사용할 수 있습니다. 매니페스트 및 보고 요청을 위한 CDN 설정에 대한 자세한 내용은 [CDN을 사용하여 MediaTailor 광고 개인화 및 콘텐츠 전송 최적화](#) 단원을 참조하십시오.

필수 설정

구성을 생성할 때 다음과 같은 필수 설정을 포함해야 합니다.

이름

구성을 설명하는 고유한 이름을 입력합니다. 이름은 구성의 기본 식별자입니다. 허용된 최대 길이는 512자입니다.

콘텐츠 소스

자산 ID를 제외하고 이 스트림에 대한 매니페스트의 URL 접두사를 입력합니다. 최대 길이는 512자입니다.

예를 들어 URL 접두사는의 HLS 다변량 재생 목록 URL `http://origin-server.com/a/main.m3u8`과의 DASH MPD URL에 `http://origin-server.com/a/` 유효합니다 `http://origin-server.com/a/dash.mpd`. 또는 `http://origin-server.com`과 같은 더 짧은 접두사를 입력할 수 있지만, 콘텐츠에 대한 플레이어 요청에서 `/a/`이 자산 ID에 포함되어야 합니다.

Note

콘텐츠 오리진이 HTTPS를 사용하는 경우 인증은 잘 알려진 인증 기관에서 받아야 합니다. 자체 서명된 인증서는 될 수 없습니다. 자체 서명된 인증서를 사용하는 경우는 콘텐츠 오리

진에 연결 AWS Elemental MediaTailor 하지 못하며 플레이어 요청에 대한 응답으로 매니페스트를 제공할 수 없습니다.

광고 의사결정 서버

광고 의사결정 서버(ADS)의 URL을 입력합니다. 이는 [3단계: ADS 요청 URL 및 쿼리 파라미터 구성](#)에 설명된 변수가 있는 URL이거나 테스트 목적으로 사용하는 정적 VAST URL입니다. 최대 길이는 25,000자입니다.

Note

ADS가 HTTPS를 사용하는 경우 인증은 잘 알려진 인증 기관에서 받아야 합니다. 자체 서명된 인증서는 될 수 없습니다. 동일한 규정이 ADS에서 반환되는 메자닌 광고 URL에도 적용됩니다. 자체 서명된 인증서를 사용하는 경우는 콘텐츠 오리진에서 매니페스트에 광고를 검색하고 연결할 AWS Elemental MediaTailor 수 없습니다.

선택적 구성 설정

선택적으로 MediaTailor 콘솔, MediaTailor API 또는 AWS Command Line Interface ()에서 구성 별칭, 개인화 세부 정보 및 고급 설정을 구성할 수 있습니다AWS CLI.

구성 별칭

다음은 MediaTailor 콘솔 또는 MediaTailor API를 사용하여 구성할 수 있는 선택적 구성 별칭입니다.

플레이어 파라미터 변수

세션 초기화 중 동적 도메인 구성의 경우 플레이어 파라미터 변수를 하나 이상 추가합니다.

플레이어 파라미터 변수를 사용하여 도메인을 동적으로 구성하는 방법에 대한 자세한 내용은 섹션을 참조하세요[여러 콘텐츠 소스에 대한 MediaTailor 도메인 변수](#).

로그 구성

다음은 로그 구성 설정입니다.

활성화 비율

MediaTailor가 CloudWatch Logs에 쓰는 재생 구성 세션 로그의 비율을 설정합니다. 예를 들어 재생 구성에 1,000개의 세션이 있고 백분율 활성화를 60으로 설정하면 MediaTailor는 600개의 세션 로그를 CloudWatch Logs에 기록합니다.

이 옵션을 활성화하면 MediaTailor가 CloudWatch Logs 계정에서 세션 로그 MediaTailor를 작성하고 관리할 수 있는 서비스 연결 역할을 자동으로 생성합니다. 자세한 내용은 [MediaTailor에 서비스 연결 역할 사용](#) 단원을 참조하십시오.

로깅 전략

MediaTailor가 내보내는 로그를 수집하는 데 사용되는 방법을 나타냅니다. CloudWatch Logs로 로그를 직접 보내려면 선택합니다 LEGACY_CLOUDWATCH. CloudWatch Logs로 로그를 전송한 다음 원하는 대상으로 로그를 제공하려면 선택합니다 VENDED_LOGS. 지원되는 대상은 CloudWatch Logs 로그 그룹, Amazon S3 버킷 및 Amazon Data Firehose 스트림입니다.

판매 로그에는 추가 설정이 필요합니다. 설정은 단원을 참조하십시오 [벤딩된 로그 사용](#).

ADS 상호 작용 로그 옵트인 이벤트

MediaTailor가 이 구성으로 초기화된 세션에 대한 RAW_ADS_RESPONSE 로그를 내보내는 것을 나타냅니다.

RAW_ADS_RESPONSE 로그 이벤트에는 ADS의 전체 VAST 또는 VMAP 응답이 포함됩니다. 따라서 로그는 광범위할 수 있으며 로깅 비용이 증가할 수 있습니다.

ADS 상호 작용 로그 제외 이벤트

MediaTailor가 ADS와의 상호 작용을 설명하는 로그에서 선택한 이벤트를 내보내지 않음을 나타냅니다.

ADS 로그 이벤트에 대한 설명은 섹션을 참조하세요 [ADS 로그](#).

매니페스트 서비스 상호 작용 로그 제외 이벤트

MediaTailor가 매니페스트 서비스와의 상호 작용을 설명하는 로그에서 선택한 이벤트를 내보내지 않음을 나타냅니다.

매니페스트 서비스 로그 이벤트에 대한 설명은 섹션을 참조하세요 [매니페스트 로그](#).

광고 컨디셔닝

다음은 MediaTailor가 광고를 콘텐츠 스트림에 스티칭하기 전에 조건을 지정하기 위해 수행하는 작업을 결정합니다.

스트리밍 미디어 파일 컨디셔닝

연결할 광고를 결정할 때 MediaTailor가 사용하는 로직을 결정합니다.

- 스트리밍 미디어 파일 컨디셔닝이 트랜스코딩으로 설정된 경우 MediaTailor는 progressive 전송으로 미디어 파일을 트랜스코딩하고 매니페스트에 연결합니다. progressive 전송 미디어 파일이 있는 광고가 공간을 채우기에 충분하지 않은 경우 MediaTailor는 이를 트랜스코딩하고 streaming 전송에 사용합니다.
- 스트리밍 미디어 파일 컨디셔닝이 없음으로 설정된 경우 MediaTailor는 streaming 전송 미디어 파일이 포함된 광고를 트랜스코딩하지 않고 매니페스트에 연결합니다. streaming 전송 미디어 파일이 있는 광고가 공간을 채우기에 충분하지 않은 경우 MediaTailor는 이를 트랜스코딩하고 progressive 전송에 사용합니다.

개인화 세부 정보

다음은 MediaTailor 콘솔 또는 MediaTailor API를 사용하여 구성할 수 있는 개인화 세부 정보입니다.

슬레이트 광고

MediaTailor가 채워지지 않은 광고 시간을 채우는 데 사용하는 고품질 MP4 자산의 URL을 입력합니다. 슬레이트 구성은 대부분의 워크플로에서는 선택 사항이지만 VPAID 광고에서는 필수입니다. 슬레이트 동작, 구성 요구 사항 및 슬레이트 표시 시기에 대한 자세한 내용은 [섹션을 참조하세요](#) [MediaTailor 슬레이트 광고 삽입](#).

범퍼 시작

시작 범퍼 자산 위치의 URL입니다. 범퍼는 광고 시간의 시작 또는 끝에서 재생되는 짧은 비디오 또는 오디오 클립입니다. Amazon의 S3 또는 다른 스토리지 서비스에 저장할 수 있습니다. 범퍼에 대한 자세한 내용은 [섹션을 참조하세요](#) [MediaTailor 범퍼 광고 삽입](#).

엔드 범퍼

엔드 범퍼 자산 위치의 URL입니다. 범퍼는 광고 시간의 시작 또는 끝에서 재생되는 짧은 비디오 또는 오디오 클립입니다. Amazon의 S3 또는 다른 스토리지 서비스에 저장할 수 있습니다. 범퍼에 대한 자세한 내용은 [섹션을 참조하세요](#) [MediaTailor 범퍼 광고 삽입](#).

개인화 임계값

슬레이트 광고와 함께 사용하는 경우에만 적용됩니다. MediaTailor가 개인화를 중단하고 기본 콘텐츠를 표시하기 전에 광고 시간에 허용되는 과소 채워진 광고 시간(초)의 최대 지속 시간을 정의합니다. 이 기능은 기본 콘텐츠 스트림에 의존하기 때문에 광고 삽입이 아닌 라이브 및 VOD 스트림에서 광고 교체를 수행하는 데 적용됩니다. 자세한 동작 시나리오 및 포괄적인 정보는 섹션을 참조하세요 [요개인화 임계값 작동 방식](#).

라이브 프리 롤 광고 의사결정 서버

기본 콘텐츠 재생이 시작되기 전에 라이브 스트림 시작 부분에 광고를 삽입하려면 광고 결정 서버(ADS)에서 광고 프리 롤의 URL을 입력하십시오. 이는 [3단계: ADS 요청 URL 및 쿼리 파라미터 구성](#)에 설명된 변수가 있는 URL이거나 테스트 목적으로 사용하는 정적 VAST URL입니다. 최대 길이는 25,000자입니다.

Note

ADS가 HTTPS를 사용하는 경우 인증은 잘 알려진 인증 기관에서 받아야 합니다. 자체 서명된 인증서는 될 수 없습니다. 동일한 규정이 ADS에서 반환되는 메자닌 광고 URL에도 적용됩니다. 자체 서명된 인증서를 사용하는 경우는 콘텐츠 오리진에서 매니페스트에 광고를 검색하고 연결할 AWS Elemental MediaTailor 수 없습니다.

프리롤의 작동 방식에 대한 자세한 내용은 [MediaTailor 프리롤 광고 삽입](#) 단원을 참조하십시오.

라이브 프리 롤 최대 허용 시간

라이브 스트림 시작 시 광고를 삽입할 때 롤링 전 광고 시간에 허용되는 최대 기간을 입력합니다. MediaTailor는 광고를 삽입할 때 이 기간을 초과하지 않습니다. ADS의 응답에 이 기간에 맞는 것보다 많은 광고가 포함된 경우 MediaTailor는 기간을 초과하지 않고 가능한 한 많은 광고로 시간을 채웁니다. MediaTailor가 가용 시간을 채우는 방법에 대한 자세한 내용은 [라이브 광고 스티칭 동작](#) 섹션을 참조하세요.

가용성 억제 모드

광고 억제라고도 하는 가능 구간 억제를 설정합니다. 기본적으로 광고 억제는 꺼져 있으며 MediaTailor는 모든를 광고 또는 슬레이트로 채웁니다. 모드가 ON로 설정되면 BEHIND_LIVE_EDGE 광고 억제가 활성화되고 MediaTailor가 매니페스트 록백 창의 가용 억제 값 시간 또는 그 뒤에서 광고 시간을 채우지 않습니다. 모드를 OFF로 설정하면 AFTER_LIVE_EDGE 광고 억제가 활성화됩니다. MediaTailor는 라이브 엣지와 가용 억제 값 및 버퍼 시간인 가용 억제 기간 안팎에서 광고 시간을 채우지 않습니다.

가용성 억제 값

가능 구간 억제 값은 라이브 끝 오프셋 시간(HH:MM:SS 형식)입니다. MediaTailor는 매니페스트 록백 윈도우에서 이 시간 또는 뒤의 광고 시간을 채우지 않습니다.

삽입 모드

삽입 모드는 플레이어에 스티칭 또는 안내 광고 삽입을 사용할 수 있는지 여부를 제어합니다. 기본 값인 모든 플레이어 세션이 연결된(서버 측) 광고 삽입을 사용하도록 STITCHED_ONLY 강제합니다. InsertionMode를 로 설정PLAYER_SELECT하면 플레이어가 세션 초기화 시 스티칭되거나 안내된 광고 삽입을 선택할 수 있습니다. 삽입 모드를 지정하지 않은 플레이어의 기본값은 스티칭됩니다.

고급 설정

다음은 선택적 고급 설정입니다. MediaTailor 콘솔, AWS Command Line Interface (AWS CLI) 또는 MediaTailor API를 사용하여 이를 구성할 수 있습니다.

CDN 콘텐츠 세그먼트 접두사

콘텐츠 세그먼트 AWS Elemental MediaTailor 의 CDN 경로에 대한 URLs을 사용하여 매니페스트를 생성할 수 있습니다. 이 단계를 수행하기 전에 오리진 서버에서 세그먼트를 가져올 수 있도록 CDN에서 규칙을 설정합니다. CDN 콘텐츠 세그먼트 접두사에 CDN 접두사 경로를 입력합니다.

MediaTailor를 CDN과 통합하는 방법에 대한 자세한 내용은 섹션을 참조하세요 [CDN을 사용하여 MediaTailor 광고 개인화 및 콘텐츠 전송 최적화](#).

CDN 광고 세그먼트 접두사

광고 세그먼트 AWS Elemental MediaTailor 의 자체 CDN 경로에 대한 URLs을 사용하여 매니페스트를 생성할 수 있습니다. 기본적으로 MediaTailor는 기본 캐시 설정을 사용하여 내부 Amazon CloudFront 배포의 광고 세그먼트를 제공합니다. 다음 예제처럼 CDN ad segment prefix(광고 세그먼트 접두사) 필드를 완료하기 전에 다음 오리진에서 광고 세그먼트를 가져올 수 있도록 CDN에서 규칙을 설정해야 합니다.

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

CDN ad segment prefix(광고 세그먼트 접두사)에서 구성에 CDN 접두사의 이름을 입력합니다.

MediaTailor를 CDN과 통합하는 방법에 대한 자세한 내용은 섹션을 참조하세요 [CDN을 사용하여 MediaTailor 광고 개인화 및 콘텐츠 전송 최적화](#).

DASH 오리진 매니페스트 유형

오리진 서버가 단일 기간 DASH 매니페스트를 생성하는 경우 드롭다운 목록을 열고 SINGLE_PERIOD를 선택합니다. 기본적으로 MediaTailor는 DASH 매니페스트를 다중 기간 매니페스트로 처리합니다. 자세한 내용은 [the section called “MPEG-DASH 소스 통합”](#) 단원을 참조하십시오.

DASH mpd 위치

(DASH의 경우 필요에 따라 선택 사항) 미디어 프레젠테이션 설명(mpd) 위치입니다. 다음과 같은 상황에서는 비활성화됨을 선택합니다.

- MediaTailor 매니페스트에 액세스하기 위한 CDN 라우팅 규칙을 설정합니다.
- 클라이언트 측 보고를 사용하거나 플레이어에 고정 HTTP 리디렉션을 지원합니다.

위치 기능에 대한 자세한 내용은 [the section called “위치 기능”](#) 단원을 참조하십시오.

트랜스코딩 프로파일 이름

이 구성을 사용자 지정 트랜스코드 프로필에 연결하는 이름입니다. 이 이름은 MediaTailor의 동적 트랜스코딩 기본값을 재정의합니다. AWS Support를 사용하여 사용자 지정 프로필을 이미 설정한 경우에만 이 필드를 작성하십시오.

광고 마커 패스스루

HLS의 경우 광고 마커 패스스루를 활성화하거나 비활성화합니다. 광고 마커 패스스루가 활성화되면 MediaTailor는 오리진 매니페스트에서 MediaTailor 개인화된 매니페스트로 EXT-X-CUE-INEXT-X-CUE-OUT, 및 EXT-X-SPLICEPOINT-SCTE35 광고 마커를 전달합니다. 광고 마커 값에는 로직이 적용되지 않으며 오리진 매니페스트에서 개인화된 매니페스트로 있는 그대로 전달됩니다. 예를 들어 EXT-X-CUE-OUT의 값이 오리진 매니페스트60에 있지만 광고가 배치되지 않은 경우 MediaTailor는 개인화된 매니페스트0에서 값을 로 변경하지 않습니다.

개별 ADS 요청 제한 시간

라이브 또는 VOD 재생 중에 광고 결정을 해결할 때 MediaTailor가 단일 HTTP 응답을 기다리는 밀리초 단위의 최대 시간입니다. 기본값: 3000. 유효한 범위: 250-6000.

MediaTailor가 라이브 및 VOD 워크플로에서 광고 스티칭을 처리하는 방법에 대한 자세한 내용은 [라이브 광고 스티칭 동작](#) 및 섹션을 참조하세요 [VOD에 대한 광고 스티칭 동작](#).

이 설정은 [PutPlaybackConfiguration](#) AdsPersonalizationTimeouts 파라미터의 AdsRequestTimeoutMilliseconds 멤버에 해당합니다.

최대 동시 ADS 요청 수

MediaTailor가 ADS에 수행할 수 있는 동시 요청 수입니다. 기본값: 1. 유효한 범위: 1-100.

이 설정은 [PutPlaybackConfiguration](#) AdPersonalizationConcurrency 파라미터의 MaxConcurrentAdsRequests 멤버에 해당합니다.

라이브 최대 ADS 개인화 시간

라이브 매니페스트 개인화 중 모든 ADS 활동에 대한 밀리초 단위의 총 시간 예산입니다. 이 예산은 단일 매니페스트 요청의 모든 광고 시간에 공유됩니다. 기본값: 10000. 유효한 범위: 250-10000.

라이브 광고 스티칭 동작에 대한 자세한 내용은 섹션을 참조하세요 [라이브 광고 스티칭 동작](#).

이 설정은 [PutPlaybackConfiguration](#) AdPersonalizationTimeouts 파라미터의 LiveMaximumAdPersonalizationTimeMilliseconds 멤버에 해당합니다.

VOD 최대 ADS 개인화 시간

VOD 매니페스트 개인화 중 모든 ADS 활동에 대한 밀리초 단위의 총 시간 예산입니다. 이 예산은 단일 매니페스트 요청의 모든 광고 시간에 공유됩니다. 기본값: 10000. 유효한 범위: 250-10000.

VOD 광고 스티칭 동작에 대한 자세한 내용은 섹션을 참조하세요 [VOD에 대한 광고 스티칭 동작](#).

이 설정은 [PutPlaybackConfiguration](#) AdPersonalizationTimeouts 파라미터의 VodMaximumAdPersonalizationTimeMilliseconds 멤버에 해당합니다.

VOD VAST 병렬화 활성화

이 설정을 켜면 MediaTailor는 VAST 응답을 반환하는 VOD 워크플로에 대해 ADS 요청을 병렬로 처리합니다. 이 설정은 VOD 워크플로에만 적용됩니다. MediaTailor는 기본적으로 이 설정을 끄니다. 적용하려면 최대 동시 ADS 요청을 보다 큰 값으로 설정합니다¹.

동시 요청이 있는 빈도 제한

VAST 병렬화를 활성화하면 일부 광고 서버는 동시 광고 요청에 대해 빈도 제한 제한을 적용하지 않을 수 있습니다. 광고 서버 공급자에게 문의하여 병렬 요청 조건에서 주파수 제한이 지원되는지 확인합니다. 광고 서버가 동시 요청으로 빈도 제한을 지원하지 않는 경우 병렬 광고 요청에 대해 빈도 제한을 올바르게 처리하는 VMAP로 전환하는 것이 좋습니다.

이 설정은 [PutPlaybackConfiguration](#) AdPersonalizationConcurrency 파라미터의 EnableVodVastParallelization 멤버에 해당합니다.

ADS 요청 제한 시간 미리 가져오기

미리 가져오기 검색 중에 광고 결정을 해결할 때 MediaTailor가 단일 HTTP 응답을 기다리는 밀리초 단위의 최대 시간입니다. 기본값: 개별 ADS 요청 제한 시간 값입니다. 유효한 범위: 250-10000.

미리 가져오기에 대한 자세한 내용은 섹션을 참조하세요 [광고 미리 가져오기](#).

이 설정은 [PutPlaybackConfiguration](#) AdPersonalizationTimeouts 파라미터의 PrefetchAdsRequestTimeoutMilliseconds 멤버에 해당합니다.

최대 ADS 개인화 시간 미리 가져오기

미리 가져오기 검색 중 모든 ADS 활동에 대한 밀리초 단위의 총 시간 예산입니다. 기본적으로이 제한 시간에는 상한이 없습니다. 유효한 범위: 250-10000.

미리 가져오기에 대한 자세한 내용은 섹션을 참조하세요 [광고 미리 가져오기](#).

이 설정은 [PutPlaybackConfiguration](#) AdPersonalizationTimeouts 파라미터의 PrefetchMaximumAdPersonalizationTimeMilliseconds 멤버에 해당합니다.

제한 시간 함께 설정

각 워크플로에 대해 개별 제한 시간과 총 시간 예산을 함께 설정해야 합니다. 개별 제한 시간은 총 예산을 초과할 수 없습니다. 라이브 및 VOD 워크플로는 단일 개별 ADS 요청 제한 시간을 공유합니다. 라이브 및 VOD 총 시간 예산과 함께 해당 제한 시간을 설정합니다. 마찬가지로 사전 가져오기 ADS 요청 제한 시간과 사전 가져오기 최대 ADS 개인 설정 시간을 함께 설정합니다. 한 ADS를 다른 ADS 요청 없이는 설정할 수 없습니다.

MediaTailor 구성 세부 정보 보기

구성의 생성 시 제공된 값 이외에 AWS Elemental MediaTailor 은 구성의 이름, 재생 엔드포인트 및 관련 액세스 URL을 표시합니다. 구성을 보려면 다음 절차를 사용합니다.

구성을 보려면

1. <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.
2. 구성 페이지에서 조회할 구성의 구성 이름을 선택합니다.

MediaTailor 구성 설정 편집

구성을 편집하여 오리진 서버 및 광고 결정 서버(ADS) 매핑을 업데이트하거나 콘텐츠 배포 네트워크(CDN)와 AWS Elemental MediaTailor 상호 작용하는 방식을 변경할 수 있습니다.

구성을 편집하려면

1. <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.
2. 구성 페이지에서 편집할 구성의 이름을 선택합니다.
3. 구성 세부 정보 페이지에서 편집을 선택한 다음 필요에 따라 구성 설정을 수정합니다. 구성 이름은 편집할 수 없습니다. 구성 속성에 대한 자세한 내용은 [MediaTailor 재생 구성 생성](#) 단원을 참조하십시오.
4. 저장을 선택합니다.

MediaTailor 구성 삭제

구성을 삭제하여에서 재생할 수 없도록 할 수 있습니다 AWS Elemental MediaTailor.

구성을 삭제하려면

1. <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.
2. 구성 페이지에서 다음 중 하나를 수행합니다.
 - 삭제할 구성의 이름을 선택합니다.
 - 구성 이름 옆에서 이름 옆의 옵션을 선택한 다음 삭제를 선택합니다.
3. 삭제 확인 상자에 **Delete**을 입력한 다음 삭제를 선택합니다.

MediaTailor 광고 삽입을 위한 콘텐츠 소스 통합

이 주제에서는 다양한 유형의 비디오 콘텐츠 소스를 MediaTailor와 통합하는 방법을 다룹니다.

MediaTailor는 라이브 및 온디맨드 콘텐츠에 대해 HLS 및 DASH 스트리밍 프로토콜을 모두 지원합니다. 서비스는 지정된 광고 시간 동안 광고 삽입 또는 교체를 수행할 수 있으며, 이러한 기능을 활성화하기 위해 입력 비디오 매니페스트의 구조 및 형식 지정에 대한 특정 요구 사항이 있습니다. 다음 주제에서는 입력 소스 요구 사항에 대한 세부 정보와 HLS 및 DASH 콘텐츠를 MediaTailor와 통합하여 맞춤형 광고 경험을 가능하게 하는 단계를 제공합니다.

주제

- [MediaTailor 광고 삽입에 대한 입력 소스 요구 사항](#)
- [HLS 소스 통합](#)
- [MPEG-DASH 소스 통합](#)
- [SigV4와의 AWS Elemental MediaTailor 오리진 상호 작용 보호](#)

MediaTailor 광고 삽입에 대한 입력 소스 요구 사항

MediaTailor를 사용하려면 입력 소스가 다음 요구 사항을 충족해야 합니다.

- Apple HLS(HTTP Live Streaming) 또는 MPEG DASH(Dynamic Adaptive Streaming over HTTP) 사용
- 라이브 스트리밍 또는 온디맨드 비디오(VOD) 사용
- 퍼블릭 인터넷에 액세스를 할 수 있도록 하고 퍼블릭 IP 주소 확보
- 표준 HTTP 포트(포트 80) 또는 HTTPS 포트(포트 443)를 사용합니다. MediaTailor는 오리진 서버 통신을 위한 사용자 지정 포트를 지원하지 않습니다.
- [MediaTailor 광고 삽입 시작하기 자습서에 설명된 형식 중 하나로 광고](#) 마커 포함

HLS 소스 통합

AWS Elemental MediaTailor 는 라이브 스트리밍 및 온디맨드 비디오(VOD)에 대해 3 이상이 EXT-X-VERSION인 .m3u8 HLS 매니페스트를 지원합니다. MediaTailor는 광고 중단이 발생하면 콘텐츠 유형에 따라 광고 삽입 또는 교체를 시도합니다. 기간을 채울 광고가 충분하지 않은 경우 나머지 광고 시간 동안 MediaTailor는 기본 콘텐츠 스트림 또는 구성된 슬레이트를 표시합니다. 콘텐츠 유형에 따른 HLS 광고 동작에 대한 자세한 내용은 섹션을 참조하세요 [AWS Elemental MediaTailor 광고 삽입 동작 이해](#).

다음 섹션에서는 MediaTailor가 HLS 매니페스트를 처리하는 방법에 대한 자세한 정보를 제공합니다.

주제

- [HLS 지원 광고 마커](#)
- [광고 마커 패스스루 활성화](#)
- [HLS 매니페스트 태그 처리](#)
- [HLS 매니페스트 예제](#)

HLS 지원 광고 마커

AWS Elemental MediaTailor 는 지원되는 광고 마커에 대한 입력 매니페스트를 구문 분석하여 HLS 매니페스트의 광고 가용성 경계를 식별합니다. 다음 섹션에서는 MediaTailor가 사용하는 마커에 대해 설명합니다.

EXT-X-ASSET

EXT-X-ASSET 태그에는 광고 의사결정 서버(ADS)가 최종 사용자를 위한 콘텐츠를 개인화하기 위해 사용하는 메타데이터가 포함되어 있습니다. EXT-X-ASSET 파라미터는 쉼표로 구분된 키-값 페어입니다.

이 태그를 사용하려면 다음 요구 사항을 충족해야 합니다.

- 오리진 매니페스트의 EXT-X-ASSET 값을 URL 인코딩해야 합니다. 다음 예제에서는 키와 URL 인코딩된 값이 포함된 EXT-X-ASSET 태그를 보여줍니다.

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

- MediaTailor ADS 구성에 동적 [asset.] 변수와 키를 포함해야 합니다. 다음 예제에서는 동적 [asset.] 변수 및 키를 사용하는 MediaTailor ADS 구성을 보여줍니다.

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

VAST 요청 예제

다음 예제에서는 ADS에 대한 VAST GET 요청을 보여줍니다.

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

EXT-X-CUE-OUT 및 EXT-X-CUE-IN

이 유형의 광고 마커가 가장 일반적입니다. 다음 예제는 이러한 신호 마커에 대한 옵션을 보여 줍니다.

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

```
...
```

```
#EXT-X-CUE-IN
```

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

```
...
```

```
#EXT-X-CUE-IN
```

```
#EXT-X-CUE-OUT
```

```
...
```

```
#EXT-X-CUE-IN
```

EXT-X-DATERANGE

EXT-X-DATERANGE 광고 마커 태그를 사용하여 SCTE35-OUT 특성을 통해 광고 시간의 타이밍을 지정합니다.

 Note

AWS Elemental MediaTailor 는 EXT-X-DATERANGE 광고 마커에 제공된 모든 START-DATE 속성을 무시합니다.

광고 시간은 다음 방법 중 하나를 사용하여 지정할 수 있습니다.

- EXT-X-DATERANGE 태그와 SCTE35-OUT 및 DURATION 사양.

예제

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

- 연결된 EXT-X-DATERANGE 태그, 첫 번째는 SCTE35-OUT 사양이고 두 번째는 SCTE35-IN 사양.

예제


```
#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
```

- 이전 옵션의 조합. EXT-X-DATERANGE 태그를 SCTE35-OUT 및 DURATION 사양으로 지정한 다음 EXT-X-DATERANGE 태그를 SCTE35-IN 사양으로 지정합니다. 이 경우 MediaTailor는 두 사양의 가장 빠른 큐인 설정을 사용합니다.

예제

```
#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

EXT-X-SPLICEPOINT-SCTE35 광고 마커에 SCTE-35 페이로드를 base64 인코딩된 이진수로 추가합니다. 디코딩된 이진수는 공급자 배치 기회 시작에 대해 신호 출력 마커 0x34를 포함하는 SCTE-35 splice_info_section을 제공하고, 공급자 배치 기회 종료에 대해 신호 입력 마커 0x35를 제공해야 합니다.

다음 예제에서는 신호 출력 및 신호 입력 마커를 지정하는 base64로 인코딩된 이진수 페이로드가 있는 스플라이스 포인트 사양을 보여줍니다.

```
#EXT-X-SPLICEPOINT-SCTE35:/DA9AAAAAAAAAAP/wBQb+uYbZqwAnAiVDVUVJAAAKqX//
AAEjW4AMEU1EU05CMDAXMTMyMjE5M190NAAAmXz5JA==
...
#EXT-X-SPLICEPOINT-SCTE35:/DA4AAAAAAAAAAP/wBQb+tTeaawAiAiBDVUVJAAAKqH+/
DBFNRFNOQjAwMTEzMjIxOTJfTjUAAlGK1s=
```

광고 마커 패스스루 활성화

기본적으로 HLS의 경우 MediaTailor 개인 맞춤형 매니페스트에는 오리진 매니페스트의 SCTE-35 광고 마커가 포함되지 않습니다. 광고 마커 패스스루가 활성화되면 MediaTailor는 오리진 매니페스트에서 다음 광고 마커를 맞춤형 매니페스트로 전달합니다.

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

광고 마커 패스스루는 선택적 설정입니다. MediaTailor 개인 맞춤형 매니페스트에 SCTE 광고 마커를 포함하려면 광고 마커 패스스루를 사용합니다. 일반적인 사용 사례는 다음과 같습니다.

- 콘텐츠 대체 - 콘텐츠 대체 또는 콘텐츠 제한을 수행합니다.
- 광고 추적 - 하나 이상의 광고 마커가 있는지 여부에 따라 광고 추적 정보를 트리거합니다.
- 플레이어 설정 - 광고 마커의 유무에 따라 플레이어의 UI에서 스크러빙 또는 카운트다운 타이머 기능을 활성화합니다.

Note

MediaTailor는 이러한 마커의 값을 변경하지 않습니다. 예를 들어 EXT-X-CUE-OUT의 값이 오리진 매니페스트60에 있지만 광고가 배치되지 않은 경우 MediaTailor는 개인화된 매니페스트0에서 값을 로 변경하지 않습니다.

광고 마커 패스스루 활성화

AWS Management Console 또는 ()를 사용하여 광고 마커 패스스루를 활성화할 수 있습니다 AWS Command Line Interface AWS CLI.

콘솔을 사용하여 광고 마커 패스스루를 활성화하려면

1. <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.
2. 새 구성 또는 구성 편집을 선택합니다.
3. 고급 설정 섹션의 드롭다운 메뉴에서 활성화를 선택합니다.

AWS Command Line Interface (AWS CLI)를 사용하여 광고 마커 패스스루를 활성화하려면

[put-playback-configuration](#) 명령을 사용합니다.

HLS 매니페스트 태그 처리

이 섹션에서는가 개인화된 출력 매니페스트에서 태그를 AWS Elemental MediaTailor 관리하는 방법을 설명합니다.

EXT-X-CUE 태그

MediaTailor는 입력 매니페스트의 EXT-X-CUE-OUTEXT-X-CUE-OUT-CONT, 및 EXT-X-CUE-IN 태그를 출력 매니페스트의 EXT-X-DISCONTINUITY 태그로 바꿉니다. DISCONTINUITY 태그는 다음 경계를 표시합니다.

- 주요 콘텐츠가 광고로 이전되는 위치
- 하나의 광고가 다른 광고로 이전되는 경우
- 광고가 주요 콘텐츠로 재이전되는 경우

EXT-X-DATERANGE 태그

MediaTailor는 입력 매니페스트에서 출력 매니페스트로 EXT-X-DATERANGE 태그를 전달합니다. 또한 MediaTailor는 EXT-X-DISCONTINUITY 태그에 해당하는 DATERANGE 태그를 삽입합니다. DISCONTINUITY 태그는 다음 경계를 표시합니다.

- 주요 콘텐츠가 광고로 이전되는 위치
- 하나의 광고가 다른 광고로 이전되는 경우
- 광고가 주요 콘텐츠로 재이전되는 경우

EXT-X-KEY 태그

MediaTailor는 입력 매니페스트에서 EXT-X-KEY 태그를 전달합니다. 이 태그는 기본 콘텐츠가 암호화되었음을 나타냅니다. 광고는 암호화되지 않으므로 MediaTailor는 광고 가능성 시작 EXT-X-KEY:METHOD=NONE 시를 삽입합니다. 재생이 기본 콘텐츠로 돌아가면 MediaTailor는 암호화 유형으로 정의된 METHOD 값이 있는 EXT-X-KEY 태그를 삽입하여 암호화를 다시 활성화합니다.

인식할 수 없는 태그

MediaTailor는 모든 알 수 없는 태그와 사용자 지정 태그를 입력 매니페스트에서 출력 매니페스트로 전달합니다.

HLS 매니페스트 예제

다음 섹션에는 HLS 오리진 매니페스트와 맞춤형 매니페스트에 대한 예가 나와 있습니다. 이러한 예제를 이해하면 MediaTailor 워크플로를 구성하고 문제를 해결하는 데 도움이 될 수 있습니다.

쿼리 파라미터가 HLS 매니페스트 및 세그먼트에 적용되는 방법에 대한 자세한 내용은 섹션을 참조하세요 [MediaTailor HLS 암시적 세션 초기화](#).

HLS 재생 목록 유형 이해

HTTP 라이브 스트리밍(HLS)은 두 가지 기본 유형의 재생 목록을 사용합니다.

다변량 재생 목록

다변량 재생 목록은 콘텐츠의 사용 가능한 모든 변환을 나열하는 최상위 인덱스 파일입니다. 여기에는 미디어 재생 목록에 대한 참조가 포함되지만 미디어 세그먼트 자체는 포함되지 않습니다. 이 재생 목록을 사용하면 플레이어가 네트워크 조건, 디바이스 기능 또는 사용자 기본 설정에 따라 가장 적합한 변환을 선택할 수 있습니다.

이 재생 목록 유형은 마스터 재생 목록, 마스터 매니페스트, 기본 재생 목록, 기본 재생 목록, 인덱스 파일 또는 마스터 M3U8 등 다양한 컨텍스트의 다른 여러 이름으로도 알려져 있습니다.

MediaTailor 워크플로에서 다변량 재생 목록은 재생 요청의 진입점이며 광고 개인화가 시작되는 곳입니다.

미디어 재생 목록

미디어 재생 목록에는 콘텐츠의 특정 변환(품질 수준)에 대한 실제 미디어 세그먼트 정보가 포함됩니다. 여기에는 타이밍 정보, 세그먼트 URLs 및 단일 변환 재생에 필요한 기타 메타데이터가 포함됩니다.

이 재생 목록 유형을 미디어 재생 목록, 하위 매니페스트, 청크리스트, 미디어 M3U8 또는 변환 재생 목록이라고도 합니다.

MediaTailor 워크플로에서 미디어 재생 목록은 콘텐츠 세그먼트와 광고 세그먼트를 모두 적절한 순서로 포함하도록 개인화됩니다.

HLS 재생 목록 유형에 대한 자세한 내용은 섹션을 참조하세요 [HLS 재생 목록 유형](#).

HLS 오리진 매니페스트 예제

다음 예제는 콘텐츠 오리진에서 HLS가 AWS Elemental MediaTailor 수신한 HLS 다변량 재생 목록을 보여줍니다.

```
#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_
```

이 다변량 재생 목록 예제에서는 다음과 같습니다.

- #EXT-X-STREAM-INF 태그는 해상도와 비트레이트가 다양한 다양한 변환을 정의합니다.
- 각 변환은 미디어 재생 목록(예: index_1.m3u8)을 참조합니다.
- #EXT-X-MEDIA 태그는 자막 트랙을 정의합니다.

다음 예제는 콘텐츠 오리진에서 HLS가 AWS Elemental MediaTailor 수신한 HLS 미디어 재생 목록을 보여줍니다. 이 예에서는 EXT-X-CUE-OUT 및 EXT-X-CUE-IN 태그를 사용하여 광고 가능 기회를 설명합니다.

```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-TARGETDURATION:7
#EXT-X-MEDIA-SEQUENCE:8779957
#EXTINF:6.006,
index_1_8779957.ts?m=1566416212
#EXTINF:6.006,
index_1_8779958.ts?m=1566416212
#EXTINF:5.372,
index_1_8779959.ts?m=1566416212
#EXT-OATCLS-SCTE35:/DA1AAAAAsvhAP/wFAXwAAAGf+/+AdLfIP4AG3dAAAEBQAAXytxmQ==
#EXT-X-CUE-OUT:20.020
#EXTINF:0.634,
```

```

index_1_8779960.ts?m=1566416212
#EXT-X-CUE-OUT-CONT:ElapsedTime=0.634,Duration=21,SCTE35=/DA1AAAAAsvhAP/wFAXwAAAGf
+AdLfiP4AG3dAAAEBAQAAXytxmQ==
#EXTINF:6.006,
index_1_8779961.ts?m=1566416212
#EXT-X-CUE-OUT-CONT:ElapsedTime=6.640,Duration=21,SCTE35=/DA1AAAAAsvhAP/wFAXwAAAGf
+AdLfiP4AG3dAAAEBAQAAXytxmQ==
#EXTINF:6.006,
index_1_8779962.ts?m=1566416212
#EXT-X-CUE-OUT-CONT:ElapsedTime=12.646,Duration=21,SCTE35=/DA1AAAAAsvhAP/wFAXwAAAGf
+AdLfiP4AG3dAAAEBAQAAXytxmQ==
#EXTINF:6.006,
index_1_8779963.ts?m=1566416212
#EXT-X-CUE-OUT-CONT:ElapsedTime=18.652,Duration=21,SCTE35=/DA1AAAAAsvhAP/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

```

이 미디어 재생 목록 예제에서는 다음과 같습니다.

- #EXTINF 태그는 각 세그먼트의 기간을 지정합니다.
- #EXT-X-CUE-OUT 태그는 광고 시간의 시작을 표시합니다.
- #EXT-X-CUE-OUT-CONT 태그는 진행 중인 광고 시간에 대한 정보를 제공합니다.
- #EXT-X-CUE-IN 태그는 광고 시간의 끝을 표시합니다.

HLS 개인 맞춤형 매니페스트 예제

다음 예제에서는 개인화된 HLS 다변량 재생 목록을 AWS Elemental MediaTailor 보여줍니다.

```

#EXTM3U
#EXT-X-VERSION:3

```

```
#EXT-X-MEDIA:LANGUAGE="eng",AUTOSELECT=YES,FORCED=NO,TYPE=SUBTITLES,URI=" ../../../../
manifest/43f3e412052f2808dd84ea1da90e92e914edddde/external-
canary-hls/ee1696a8-4f7f-4c4c-99de-9821131847e8/3.m3u8",GROUP-
ID="subtitles",DEFAULT=YES,NAME="caption_1"
#EXT-X-INDEPENDENT-SEGMENTS
#EXT-X-STREAM-INF:CODECS="avc1.640029,mp4a.40.2",AVERAGE-
BANDWIDTH=2526299,RESOLUTION=960x540,SUBTITLES="subtitles",FRAME-
RATE=29.97,BANDWIDTH=2665726
../../../../manifest/43f3e412052f2808dd84ea1da90e92e914edddde/external-canary-hls/
ee1696a8-4f7f-4c4c-99de-9821131847e8/0.m3u8
#EXT-X-STREAM-INF:CODECS="avc1.640029,mp4a.40.2",AVERAGE-
BANDWIDTH=3736264,RESOLUTION=1280x720,SUBTITLES="subtitles",FRAME-
RATE=29.97,BANDWIDTH=3956044
../../../../manifest/43f3e412052f2808dd84ea1da90e92e914edddde/external-canary-hls/
ee1696a8-4f7f-4c4c-99de-9821131847e8/1.m3u8
#EXT-X-STREAM-INF:CODECS="avc1.4D401E,mp4a.40.2",AVERAGE-
BANDWIDTH=951107,RESOLUTION=640x360,SUBTITLES="subtitles",FRAME-
RATE=29.97,BANDWIDTH=995315
../../../../manifest/43f3e412052f2808dd84ea1da90e92e914edddde/external-canary-hls/
ee1696a8-4f7f-4c4c-99de-9821131847e8/2.m3u8
```

MediaTailor가 맞춤형 광고 삽입을 활성화하는 세션별 정보를 포함하도록 미디어 재생 목록 URLs을 수정한 방법을 확인합니다.

다음 예제에서는 개인화된 미디어 재생 목록을 AWS Elemental MediaTailor 보여줍니다.

```
#EXTM3U
#EXT-X-VERSION:6
#EXT-X-TARGETDURATION:7
#EXT-X-MEDIA-SEQUENCE:8779957
#EXT-X-DISCONTINUITY-SEQUENCE:0
#EXTINF:6.006,
https://777788889999.mediapackage.us-west-2.amazonaws.com/out/v1/
e309ffd02ba8498d864dcaacff7a5ad9/index_1_8779957.ts?m=1566416212
#EXTINF:6.006,
https://777788889999.mediapackage.us-west-2.amazonaws.com/out/v1/
e309ffd02ba8498d864dcaacff7a5ad9/index_1_8779958.ts?m=1566416212
#EXTINF:5.372,
https://777788889999.mediapackage.us-west-2.amazonaws.com/out/v1/
e309ffd02ba8498d864dcaacff7a5ad9/index_1_8779959.ts?m=1566416212
#EXT-X-DISCONTINUITY
#EXTINF:3.066667,
```

```

    .././../././segment/43f3e412052f2808dd84ea1da90e92e914edddde/external-canary-hls/
ee1696a8-4f7f-4c4c-99de-9821131847e8/0/8779960
    #EXTINF:3.0,
    .././../././segment/43f3e412052f2808dd84ea1da90e92e914edddde/external-canary-hls/
ee1696a8-4f7f-4c4c-99de-9821131847e8/0/8779961
    #EXTINF:3.0,
    .././../././segment/43f3e412052f2808dd84ea1da90e92e914edddde/external-canary-hls/
ee1696a8-4f7f-4c4c-99de-9821131847e8/0/8779962
    #EXTINF:3.0,
    .././../././segment/43f3e412052f2808dd84ea1da90e92e914edddde/external-canary-hls/
ee1696a8-4f7f-4c4c-99de-9821131847e8/0/8779963
    #EXTINF:2.966667,
    .././../././segment/43f3e412052f2808dd84ea1da90e92e914edddde/external-canary-hls/
ee1696a8-4f7f-4c4c-99de-9821131847e8/0/8779964
    #EXT-X-DISCONTINUITY
    #EXTINF:6.006,
    https://777788889999.mediapackage.us-west-2.amazonaws.com/out/v1/
e309ffd02ba8498d864dcaacff7a5ad9/index_1_8779963.ts?m=1566416212
    #EXTINF:1.368,
    https://777788889999.mediapackage.us-west-2.amazonaws.com/out/v1/
e309ffd02ba8498d864dcaacff7a5ad9/index_1_8779964.ts?m=1566416212
    #EXTINF:4.638,
    https://777788889999.mediapackage.us-west-2.amazonaws.com/out/v1/
e309ffd02ba8498d864dcaacff7a5ad9/index_1_8779965.ts?m=1566416212
    #EXTINF:6.006,
    https://777788889999.mediapackage.us-west-2.amazonaws.com/out/v1/
e309ffd02ba8498d864dcaacff7a5ad9/index_1_8779966.ts?m=1566416212
    #EXTINF:6.006,
    https://777788889999.mediapackage.us-west-2.amazonaws.com/out/v1/
e309ffd02ba8498d864dcaacff7a5ad9/index_1_8779967.ts?m=1566416212
    #EXTINF:6.006,
    https://777788889999.mediapackage.us-west-2.amazonaws.com/out/v1/
e309ffd02ba8498d864dcaacff7a5ad9/index_1_8779968.ts?m=1566416212

```

이 개인화된 미디어 재생 목록 예제에서는 다음과 같습니다.

- MediaTailor가 콘텐츠 세그먼트 사이에 광고 세그먼트를 삽입했습니다.
- #EXT-X-DISCONTINUITY 태그는 콘텐츠와 광고 간의 전환을 표시합니다.
- 콘텐츠 세그먼트는 오리진 서버에서 제공됩니다(이 예제에서는 MediaPackage).
- 광고 세그먼트는 MediaTailor의 광고 세그먼트 스토리지에서 제공됩니다.

개인화된 매니페스트의 주요 차이점

MediaTailor가 HLS 매니페스트를 개인화하면 다음과 같은 몇 가지 중요한 변경이 이루어집니다.

다변량 재생 목록 변경 사항

- 변형 재생 목록 URLs 세션 정보가 포함된 MediaTailor 관리형 URLs 가리키도록 다시 작성됩니다.
- 최적의 재생을 위해 태그 순서를 재구성할 수 있습니다.

변형 재생 목록 변경 사항

- 광고 마커(EXT-X-CUE-OUT, EXT-X-CUE-IN)는 실제 광고 세그먼트로 대체됩니다.
- 불연속성 마커(EXT-X-DISCONTINUITY)가 콘텐츠/광고 경계에 추가됩니다.
- 콘텐츠 세그먼트 URLs 오리진 또는 CDN을 가리키도록 다시 작성됩니다.
- MediaTailor의 광고 세그먼트 스토리지를 가리키도록 광고 세그먼트 URLs이 추가됩니다.

이러한 변경 사항을 이해하면 MediaTailor 워크플로의 문제를 해결하고 CDN 및 플레이어의 적절한 구성을 보장하는 데 도움이 될 수 있습니다.

관련 주제

HLS 매니페스트 및 MediaTailor에 대한 자세한 내용은 다음 주제를 참조하세요.

- [HLS 재생 목록 유형](#) - HLS 재생 목록 유형에 대한 자세한 설명
- [CDN을 사용하여 MediaTailor 광고 개인화 및 콘텐츠 전송 최적화](#) - MediaTailor에서 CDN을 사용하는 방법에 대한 정보
- [MediaTailor 광고 삽입 작동 방식](#) - MediaTailor 광고 삽입 작동 방식 개요

MPEG-DASH 소스 통합

AWS Elemental MediaTailor 는 DASH 동적 프로파일에 대한 지침을 따르는 .mpd 라이브 및 온디맨드 비디오(VOD) 매니페스트를 지원합니다. MediaTailor는 다중 기간 및 단일 기간 DASH 호환 매니페스트 입력을 수락하고 다중 기간 DASH 호환 매니페스트 출력을 제공합니다.

입력 매니페스트에는 다음이 있어야 합니다.

- `splice insert` 또는 `time signal`에 대한 스플라이스 정보 설정이 있는 SCTE-35 이벤트 스트림. 설정은 XML 또는 base64로 인코딩된 이진수로 제공될 수 있습니다.

- segment timelines가 있는 Segment templates

게시된 매니페스트의 경우 MediaTailor는 오리진 서버의 업데이트를 변경하지 않고 그대로 두도록 요구합니다.

- start 속성에 지시된 기간 시작 시간.
- 기간 표시의 세그먼트 템플릿에서 presentationTimeOffset의 값입니다.

가장 좋은 방법은 광고에 콘텐츠 스트림 기간과 동일한 AdaptationSet 및 Representation 설정을 제공하는 것입니다. 이러한 설정을 AWS Elemental MediaTailor 사용하여 콘텐츠 스트림과 일치하도록 광고를 트랜스코딩하여 둘 간에 원활하게 전환할 수 있습니다.

다음 섹션에서는 MediaTailor가 DASH 매니페스트에서 광고를 처리하는 방법에 대한 자세한 정보를 제공합니다.

주제

- [DASH 광고 마커](#)
- [DASH 광고 가능 기간](#)
- [DASH의 SCTE-35 마커 및 기간 경계](#)
- [DASH 매니페스트 세그먼트 번호 지정](#)
- [DASH MPD 예제](#)
- [DASH 위치 기능](#)

DASH 광고 마커

MediaTailor

AWS Elemental MediaTailor 는 SCTE-35 큐아웃 마커를 사용하여 다음 로직을 사용하여 DASH 매니페스트의 광고 시간을 식별합니다.

- 다중 기간 DASH: MediaTailor는 SpliceInsert 또는 큐아웃 마커Period가 포함된 각의 첫 번째 TimeSignal Event에 대한 광고를 삽입합니다. MediaTailor는의 추가 Event 마커를 무시합니다Period.
- 단일 기간 DASH: MediaTailor는 SpliceInsert 또는 큐아웃 마커Period가 포함된 TimeSignal Event에 각각 광고를 삽입합니다.

기본적으로는 DASH 매니페스트를 다중 기간 매니페스트로 AWS Elemental MediaTailor 관리합니다. 오리진 서버에서 단일 기간 DASH 매니페스트를 처리하도록 구성을 변경할 수 있습니다. 자세한 내용은 [the section called “구성 생성”](#) 단원을 참조하세요.

다음 섹션에서는 DASH 광고 마커 처리에 대한 추가 세부 정보를 제공하고 오리진에서 데코레이션된 매니페스트를 제공합니다.

DASH 오리진 매니페스트 XML 요구 사항

MediaTailor가 광고 시간을 식별하려면 오리진의 DASH 매니페스트에 있는 광고 마커의 형식을 올바르게 지정해야 합니다. 다음 주제에서는 이러한 형식 지정 요구 사항을 일반 XML로 설명합니다.

SpliceInsert 일반 XML의

SpliceInsert 일반 XML의 광고 마커에는 다음이 포함되어야 합니다.

- EventStream에는 다음 속성이 있어야 합니다. `schemeIdUri=urn:scte:scte35:2013:xml`
- Event는 다음을 가져야 합니다. `scte35:SpliceInfoSection`
- `scte35:SpliceInfoSection`은 다음을 보유해야 합니다. `scte35:SpliceInsert`
- `scte35:SpliceInsert`에는 다음 속성이 있어야 합니다. `outOfNetworkIndicator="true"`

Example SpliceInsert XML의

다음 예제에서 필수 SCTE 마커는 굵게 표시됩니다.

```
<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 일반 XML의

TimeSignal 일반 XML의 광고 마커에는 다음이 포함되어야 합니다.

- EventStream에는 다음 속성이 있어야 합니다. `schemeIdUri=urn:scte:scte35:2013:xml`
- Event는 다음을 가져야 합니다. `scte35:SpliceInfoSection`
- `scte35:SpliceInfoSection`는 다음을 가져야 합니다. `scte35:TimeSignal`
- `scte35:SpliceInfoSection` 도를 보유해야 합니다. `scte35:SegmentationDescriptor`
- `scte35:SegmentationDescriptor`에는 다음 속성이 있어야 합니다. 여기서 값은 유효한입니다. [큐아웃 번호](#) `segmentationTypeId="xx"`
- `scte35:SegmentationDescriptor`를 보유해야 합니다. `scte35:SegmentationUpid`

큐아웃 번호

다음은에 지원되는 큐아웃 번호입니다segmentationTypeId.

분할 메시지	segmentationTypeId 값	16진수 값
배포자 광고 종료	51	0x51
배포자 광고 시작	50	0x32
배포자 배치 기회 종료	55	0x37
배포자 배치 기회 시작	54	0x36
휴식 종료	35	0x23
공급자 광고 종료	49	0x31
공급자 광고 시작	48	0x30
공급자 오버레이 배치 기회 종료	57	0x39
공급자 오버레이 배치 기회 시작	56	0x38

분할 메시지	segmentationTypeId 값	16진수 값
공급자 배치 기회 종료	53	0x35
공급자 배치 기회 시작	52	0x34
휴식 시작	34	0x22

Example **TimeSignal** XML의

다음 예제에서 필수 SCTE 마커는 굵게 표시됩니다.

```
<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 오리진 매니페스트 base64 인코딩 바이너리 요구 사항

MediaTailor가 광고 시간을 식별하려면 오리진의 DASH 매니페스트에 있는 광고 마커의 형식을 올바르게 지정해야 합니다. 다음 주제에서는 base64 인코딩 바이너리의 이러한 형식 지정 요구 사항에 대해 설명합니다.

base64 인코딩 매니페스트의 TimeSignal 및 SpliceInsert 광고 마커에는 다음이 포함되어야 합니다.

- EventStream에는 다음 속성이 있어야 합니다. `urn:scte:scte35:2014:xml+bin`
- Event는를 보유해야 합니다. `scte35:Signal`
- `scte35:Signal`는 base64로 인코딩된 바이너리 `scte35:Binary`가 포함된를 보유해야 합니다.

디코딩된 바이너리는 일반 XML 광고 마커에 필요한 것과 동일한 정보를 `splice_info_section`에 제공해야 합니다.

- 명령 유형은 `splice_insert()` 또는 중 하나여야 합니다. `time_signal()`
- 추가 설정은 [TimeSignal 일반 XML의](#) 및에 설명된 설정을 준수해야 합니다 [SpliceInsert 일반 XML의](#).

디코딩된 이진수는 XML이 `scte35:SpliceInfoSection` 요소에 제공하는 것과 동일한 정보 집합을 담은 `splice_info_section`을 제공해야 합니다. 명령 유형은 `splice_insert()` 또는 `time_signal()`이어야 하며 추가적인 설정은 이전에 XML 제공에 관해 설명한 설정에 부합해야 합니다.

다음 예제에서는 이 옵션을 필수 마커를 굵은 글씨로 표시하여 보여줍니다.

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

        <scte35:Binary>QW5vdGhlciB0ZXN0IHN0cmLuZyBmb3IgaWZ5b2RpbmcgdG8gQmFzZTY0IGVuY29kZWQgYmLuYXJ5J5Lg=
scte35:Binary>
        </scte35:Signal>
      </Event>
    .
    .
    .
```

</Period>

다음은 이전 예제에서 열거한 첫 번째 이벤트에 대한 디코딩된 이진수입니다.
splice_command_type에 대한 설정은 5이며 splice_insert를 나타냅니다.

```
{
  "table_id": 252,
  "section_syntax_indicator": false,
  "private_indicator": false,
  "section_length": 33,
  "protocol_version": 0,
  "encrypted_packet": false,
  "encryption_algorithm": 0,
  "pts_adjustment": 0,
  "cw_index": 0,
  "tier": "0xFFFF",
  "splice_command_length": 16,
  "splice_command_type": 5,
  "splice_command": {
    "splice_event_id": 448,
    "splice_event_cancel_indicator": false,
    "out_of_network_indicator": true,
    "program_splice_flag": true,
    "duration_flag": true,
    "splice_immediate_flag": false,
    "utc_splice_time": {
      "time_specified_flag": false,
      "pts_time": null
    },
  },
  "component_count": 0,
  "components": null,
  "break_duration": {
    "auto_return": false,
    "duration": {
      "pts_time": 2160000,
      "wall_clock_seconds": 24.0,
      "wall_clock_time": "00:00:24:00000"
    }
  },
  "unique_program_id": 49152,
  "avail_num": 0,
  "avails_expected": 0
},
```

```

    "splice_descriptor_loop_length": 0,
    "splice_descriptors": null,
    "Scte35Exception": {
      "parse_status": "SCTE-35 cue parsing completed with 0 errors.",
      "error_messages": [],
      "table_id": 252,
      "splice_command_type": 5
    }
  }
}

```

DASH 광고 가능 기간

재생 중에에서 광고 가능 구간이 AWS Elemental MediaTailor 발견되면 해당 구간의 일부 또는 전부를 광고로 대체합니다. MediaTailor는 광고 시간이 시작될 때 광고 교체를 시작하고 다음과 같이 광고를 포함합니다.

- 광고 가능 구간에서 기간을 지정하는 경우 MediaTailor에는 다음 콘텐츠를 덮어쓰지 않고 기간 경계 내에 들어갈 수 있는 만큼의 광고가 포함됩니다.
- 기간이 제공되지 않으면 MediaTailor는 광고 가능 구간이 끝날 때까지 광고를 포함합니다. 여러 기간 매니페스트의 경우 이는 기간의 끝 부분입니다. 단일 기간 매니페스트의 경우 이벤트의 끝입니다. MediaTailor는 광고 시간의 끝을 지나 광고를 재생하지 않으며, 광고가 끝날 때 다음 콘텐츠를 덮어 쓰는 대신 현재 광고를 잘라냅니다.

에서 광고 가능 기간을 AWS Elemental MediaTailor 찾는 방법

AWS Elemental MediaTailor 는 다음 순서로 기간 설정을 검색합니다.

1. Event duration
2. 스포라이스 삽입의 경우 scte35:BreakDuration duration
3. 시간 기호의 경우 scte35:SegmentationDescriptor segmentationDuration

AWS Elemental MediaTailor 가 이러한 설정을 찾지 못하면 기간 없이 광고 포함을 관리합니다.

다음 예제에서는 duration가 있는 Event를 보여줍니다.

```

<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>
    ...

```

다음 예제에서는 지속시간을 지정하지 않은 광고 시간을 보여줍니다. Event에는 duration이 없으며 scte35:SpliceInsert 요소는 scte35:BreakDuration 하위 요소를 포함하지 않습니다.

```

<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>
    ...

```

DASH의 SCTE-35 마커 및 기간 경계

MediaTailor는 단일 기간 또는 다중 기간 매니페스트를 전달하는 DASH 오리진을 지원합니다. 단일 기간 매니페스트의 경우 MediaTailor는의 타이밍 정보를 사용하여 매니페스트를 다중 기간 형식으로 변환EventStream하여 기간 경계를 정의합니다. 변환 후 단일 기간 매니페스트와 다중 기간 매니페스트는 모두 동일한 광고 개인화 워크플로를 거칩니다.

MediaTailor는 기간 내에 선형 광고 중단 이벤트를 감지하면 전체 기간을 광고 중단으로 취급하고 그에 따라 개인화합니다. MediaTailor는 이벤트 자체 내에서 신호되는 타이밍 오프셋에 관계없이 기간 시작 시간(Period@start)을 광고 시간 시작 시간으로 사용합니다.

주요 동작

MediaTailor는 DASH 기간에 SCTE-35 이벤트를 처리할 때 다음 규칙을 적용합니다.

- 선형 광고 시간의 경우 이벤트 타이밍이 기간 경계 이후에 광고 시간이 시작됨을 나타내는 경우 MediaTailor는 신호된 시작 시간을 무시하고에서 개인화를 시작합니다Period@start. 비선형(오버레이) 이벤트의 경우 MediaTailor는 신호된를 준수합니다presentationTime.
- 기존 선형 광고 시간의 중간에 새 SCTE 이벤트가 나타나면 MediaTailor는 이를 무시합니다.
- 활성 비선형(오버레이) 이벤트가 오리진 기간에 이미 있는 경우 MediaTailor는 해당 기간의 새 선형 이벤트를 무시합니다.

이벤트 신호 전송 기간 중

AWS Elemental MediaPackage v2를 사용하면 MediaTailor는 마커 경계에 새 기간이 생성되지 않은 경우에도 SCTE-35 마커를 수신합니다. 이로 인해 콘텐츠 기간 중에 SCTE 마커가 나타날 수 있습니다.

새로운 SCTE-35 이벤트가 콘텐츠 기간 중에 선형 광고 시간에 신호를 보내면 MediaTailor는 이를 광고 시간 기회로 인식하지만 신호 이벤트 시간이 Period@start 아닌에서 개인화를 시작합니다. 이 불일치로 인해 다음과 같은 문제가 발생할 수 있습니다.

- 최종 사용자에게 표시되지 않는 광고
- 타이밍 갭으로 생성된 나머지 기간을 채우기 위해 삽입된 슬레이트

기간 시작부터 이벤트 오프셋 계산

다음 공식을 사용하여 이벤트가 기간 시작으로부터 얼마나 멀리 오프셋되는지 확인합니다.

$$\text{Offset} = (\text{presentationTime} - \text{presentationTimeOffset}) / \text{timescale}$$

다음 예제는 기간 시작에서 오프셋presentationTime된이 EventStream 있는를 보여줍니다.

```
<Period id="1741948755359" start="PT10522H39M15.359S">
  <EventStream schemeIdUri="urn:scte:scte35:2013:xml" timescale="900000"
    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>

```

이 예제에서 오프셋 계산은 다음과 같습니다.

$156775484008800 - 156775387982400 = 96026400 \text{ ticks}$

$96026400 / 90000 = 1066.96 \text{ seconds (approximately 17 minutes 46 seconds)}$

알려진 휴식 기간은 다음과 같습니다.

$36900000 / 90000 = 410 \text{ seconds (approximately 6 minutes 50 seconds)}$

이벤트는 기간 경계에서 시작되지 않으므로 MediaTailor는 의도Period@start한 휴식 시간보다 17 분 이상 빠른 개인화를 시작합니다. 이로 인해 광고가 실제 휴식 기간과 일치하지 않습니다. 슬레이트가 구성된 경우 MediaTailor는 슬레이트 기간을 삽입하여 간격을 채웁니다.

조기 큐인

초기 신호는 광고 중단이 원래 신호 지속 시간 이전에 종료될 때 발생합니다. 다중 기간 매니페스트에서는 원래 광고 시간을 조기에 종료하는 새 기간이 생성됩니다.

동작은 AWS Elemental MediaPackage 버전마다 다릅니다.

- MediaPackage v1 - 새 기간을 생성하고 해당 기간 내의 모든 이벤트 정보를 제거합니다.
- MediaPackage v2 -에 원래 큐 아웃과 큐 인EventStream이라는 두 개의 광고 마커가 포함되어 있습니다. 신호음을 보내려면 outOfNetworkIndicator로 설정합니다false.

MediaTailor는 outOfNetworkIndicator 값을 확인하여 초기 신호를 인식합니다. MediaTailor는 같은 기간에 큐아웃과 일치하는 큐인을 찾으면 큐인을 새 광고 브레이크 기회로 취급하지 않습니다.

다음 예제는 큐아웃과 초기 큐인을 모두 EventStream 포함하는를 보여줍니다.

```

<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>

```

첫 번째 이벤트(outOfNetworkIndicator="true")는 광고 중단 시작 신호를 보냅니다. 두 번째 이벤트(outOfNetworkIndicator="false")는 브레이크를 종료하는 초기 신호를 보냅니다.

중간 기간 SCTE-35 마커 문제 완화

DASH 라이브 스트림에서 기간 중간에 나타나는 SCTE-35 마커로 인해 예기치 않은 광고 개인화 동작이 발생하는 경우 MediaTailor는 광고 개인화 워크플로에 적용되지 않는 특정 마커를 억제하거나 전달할 수 있습니다.

마커 억제가 활성화되면 MediaTailor는 다음 작업을 수행합니다.

- 마커 억제 - MediaTailor는 지정된 마커를 무시하고 생성된 매니페스트에서 해당 이벤트를 제거합니다. 다른 이벤트가 남아 있지 않으면 MediaTailor는 해당 기간EventStream에서 빈 도 제거합니다.
- 패스스루 마커 - MediaTailor는 개인 맞춤을 위해 지정된 마커를 무시하지만 마커는 생성된 매니페스트에 남아 있습니다.

억제 또는 패스스루에는 다음과 같은 브레이크 유형이 지원됩니다.

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

MediaTailor는 억제되거나 전달된 각 마커를 로깅하고 매니페스트 응답에서 하나 이상의 마커가 억제 될 때 개수 지표를 내보냅니다.

Important

이 구성은 라이브 스트리밍 모드의 DASH 매니페스트에만 적용됩니다. 구성에 대해 마커 억제 또는 패스스루를 구성하려면 AWS Support에 문의하세요. 지원에 문의할 때 억제하거나 전달 할 중단 유형을 지정합니다.

DASH 매니페스트 세그먼트 번호 지정

MediaTailor<SegmentTemplate>는 <SegmentTimeline> 및 media 속성을 사용하여 정의된 미디어 세그먼트를 지원합니다. \$Number\$ 식별자 또는 \$Time\$ 식별자를 사용하여 media 속성에 미디어 세그먼트 목록을 지정할 수 있습니다.

다음 예제에서는 \$Number\$ 식별자를 사용하는 media 속성 설정이 있는 SegmentTemplate을 보여줍니다.

```
<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>
```

다음 예제에서는 \$Time\$ 식별자를 사용하는 media 속성 설정이 있는 SegmentTemplate을 보여줍니다.

```

<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 예제

다음 섹션에서는 DASH 오MPDs 및 개인화된 MPDs의 예를 제공합니다. 이러한 예제를 이해하면 MediaTailor 워크플로를 구성하고 문제를 해결하는 데 도움이 될 수 있습니다.

쿼리 파라미터가 DASH 매니페스트 및 세그먼트에 적용되는 방법에 대한 자세한 내용은 섹션을 참조하세요 [MediaTailor DASH 암시적 세션 초기화](#).

DASH MPD 구조 이해

Dynamic Adaptive Streaming over HTTP(DASH)는 미디어 프레젠테이션 설명(MPD) 매니페스트를 사용하여 스트리밍 콘텐츠를 제공합니다. MPD는 미디어 콘텐츠의 구조와 가용성을 설명하는 XML 문서입니다.

MPD(미디어 프레젠테이션 설명)

MPD는 미디어 콘텐츠의 구조와 가용성을 설명하는 DASH 스트리밍의 기본 매니페스트 파일입니다. 여기에는 스트리밍 콘텐츠를 구성하는 기간, 적응 세트, 표현 및 세그먼트에 대한 정보가 포함되어 있습니다.

이 매니페스트 유형은 DASH 매니페스트, DASH MPD, 마스터 매니페스트(HLS와 비교 시) 또는 프레젠테이션 매니페스트 등 다양한 컨텍스트에서 다른 여러 이름으로도 알려져 있습니다.

MediaTailor 워크플로에서 MPD는 재생 요청의 진입점이며 광고 개인화가 시작되는 곳입니다.

Period

기간은 DASH 프레젠테이션의 임시 섹션입니다. 각 기간은 하나 이상의 적응 세트를 포함하며 미디어 시간 범위를 나타냅니다. 광고 삽입 워크플로에서는 일반적으로 콘텐츠와 광고를 구분하는 데 별도의 기간이 사용됩니다.

MediaTailor 워크플로에서 기간은 기본 콘텐츠를 광고 콘텐츠와 분리하는 데 사용되며, 각 광고는 일반적으로 자체 기간으로 표시됩니다.

AdaptationSet

AdaptationSet는 하나 또는 여러 미디어 콘텐츠 구성 요소의 교체 가능한 인코딩 버전 세트를 그룹화합니다. 예를 들어 한 AdaptationSet에는 여러 오디오 언어 옵션이 포함될 수 있습니다.

MediaTailor 워크플로에서는 콘텐츠와 광고 간에 일관된 미디어 유형을 유지하기 위해 광고 삽입 중에 AdaptationSets가 보존됩니다.

표시

표현은 AdaptationSet 내의 미디어 콘텐츠의 특정 인코딩 버전입니다. 각 표현은 일반적으로 비트 전송률, 해상도 또는 기타 인코딩 파라미터가 다르므로 클라이언트가 네트워크 조건 및 디바이스 기능에 따라 가장 적합한 버전을 선택할 수 있습니다.

MediaTailor 워크플로에서 광고 기간의 표현은 원활한 시청 경험을 보장하기 위해 콘텐츠 기간의 표현과 최대한 가깝게 매칭됩니다.

DASH 매니페스트 유형에 대한 자세한 내용은 섹션을 참조하세요 [DASH 매니페스트 유형](#).

의 DASH 매니페스트 구조 및 구성에 대한 자세한 내용은 DASH 개요의 MediaPackage 사용 설명서 섹션을 AWS Elemental MediaPackage참조하세요.

라이브 DASH MPD 예제

이 섹션에서는 라이브 DASH MPDs의 예를 제공합니다. 각 예제에서는 MediaTailor가 광고로 MPD를 개인화한 후 오리진 서버에서 수신한 MPD를 나열합니다.

DASH MPD 스포라이스 삽입 예제

스포츠라이스 삽입에 대한 DASH 오리진 MPD 예제

MPD의 다음 예제는 콘텐츠 오리진에서 DASH가 수신한 매니페스트의 광고 시간을 보여줍니다. 이 예제에서는 SpliceInsert 마커를 사용하여 광고 시간을 나타냅니다.

```
<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"/>
            </SegmentTimeline>
        </SegmentTemplate>
    </Representation>
</AdaptationSet>

```



```

        <S t="16635663105" d="95232"/>
        <S t="16635758337" d="96256" r="5"/>
        <S t="16636335873" d="71680"/>
    </SegmentTimeline>
</SegmentTemplate>
</Representation>
</AdaptationSet>
</Period>

```

이 오리진 MPD 예제에서는 다음과 같습니다.

- <EventStream> 요소에는 광고 시간을 나타내는 SCTE-35 마커가 포함되어 있습니다.
- <scte35:SpliceInsert> 요소는 광고 가능 여부에 대한 세부 정보를 제공합니다.
- <scte35:BreakDuration> 요소는 광고 중단 기간을 지정합니다.
- <AdaptationSet> 요소는 사용 가능한 비디오 및 오디오 스트림을 정의합니다.

스플라이스 삽입에 대한 DASH 맞춤형 MPD 예제

AWS Elemental MediaTailor 는 광고 사양으로 광고 시간을 개인화합니다. 개인화는 플레이어에서 수신되는 최종 사용자 데이터와 현재 진행 중인 광고 캠페인을 나타냅니다.

다음 예제는가 광고 시간을 AWS Elemental MediaTailor 개인화한 후 광고 시간을 보여줍니다.

```

<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>

```

이 개인화된 MPD 예제에서는 다음과 같습니다.

- MediaTailor가 광고 콘텐츠에 대한 새 기간을 생성했습니다.
- <BaseURL> 요소는 광고 콘텐츠 위치를 가리킵니다.
- <AdaptationSet> 요소는 콘텐츠와 유사한 구조를 유지합니다.
- <Representation> 요소는 광고 콘텐츠에 대해 다양한 품질 수준을 제공합니다.

DASH MPD 시간 신호 예제

시간 신호에 대한 DASH 오리진 MPD 예제

MPD의 다음 예제는 콘텐츠 오리진에서 DASH가 수신한 매니페스트의 광고 시간을 보여줍니다. 이 예제에서는 TimeSignal 마커를 사용하여 광고 가능 여부를 나타냅니다.

```

<Period start="PT173402.036S" id="46041">
    <EventStream timescale="90000" schemeIdUri="urn:scte:scte35:2013:xml">
        <Event duration="9450000">

```

```

    <scte35:SpliceInfoSection protocolVersion="0" ptsAdjustment="183265" tier="4095">
      <scte35:TimeSignal>
        <scte35:SpliceTime ptsTime="7835775000"/>
      </scte35:TimeSignal>
      <scte35:SegmentationDescriptor segmentationEventId="99"
segmentationEventCancelIndicator="false" segmentationDuration="9450000">
        <scte35:DeliveryRestrictions webDeliveryAllowedFlag="true"
noRegionalBlackoutFlag="true" archiveAllowedFlag="true" deviceRestrictions="3"/>
        <scte35:SegmentationUpid segmentationUpidType="8" segmentationUpidLength="0"/>
      </scte35:SegmentationDescriptor>
    </scte35:SpliceInfoSection>
  </Event>
</EventStream>
  <AdaptationSet mimeType="video/mp4" segmentAlignment="true"
subsegmentAlignment="true" startWithSAP="1" subsegmentStartsWithSAP="1"
bitstreamSwitching="true">
    <Representation id="1" width="960" height="540" frameRate="30000/1001"
bandwidth="1000000" codecs="avc1.4D401F">
      <SegmentTemplate timescale="30000" media="index_video_1_0_$Number
$.mp4?m=1528475245" initialization="index_video_1_0_init.mp4?m=1528475245"
startNumber="178444" presentationTimeOffset="10395907501">
        <SegmentTimeline>
          <S t="10395907501" d="60060" r="29"/>
          <S t="10397709301" d="45045"/>
        </SegmentTimeline>
      </SegmentTemplate>
    </Representation>
  </AdaptationSet>
</Period>

```

이 오리진 MPD 예제에서는 다음과 같습니다.

- <scte35:TimeSignal> 요소가 대신 사용됩니다. <scte35:SpliceInsert>
- 는 광고 가능 여부에 대한 추가 정보를 <scte35:SegmentationDescriptor> 제공합니다.

시간 신호에 대한 DASH 맞춤형 MPD 예제

AWS Elemental MediaTailor 는 광고 사양으로 광고 시간을 개인화합니다. 개인화는 플레이어에서 수신되는 최종 사용자 데이터와 현재 진행 중인 광고 캠페인을 나타냅니다.

다음 예제는가 광고 시간을 AWS Elemental MediaTailor 개인화한 후 광고 시간을 보여줍니다.

```
<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>
```

시간 신호에 대한 개인화된 MPD는 스포라이스 삽입과 유사하며 MediaTailor는 광고 콘텐츠에 대한 새 기간을 생성합니다.

VOD DASH MPD 예제

이 섹션에서는 온디맨드 비디오(VOD) DASH MPDs의 예를 제공합니다. 각 예제에서는 MediaTailor가 광고로 MPD를 개인화한 후 오리진 서버에서 수신한 MPD를 나열합니다.

VOD DASH MPDs 라이브 MPDs와 동일한 구조를 따르지만 일반적으로 MPD 요소에 `type="static"` 속성이 있으며 서로 다른 콘텐츠 세그먼트에 대해 여러 기간을 포함할 수 있습니다.

VOD DASH MPDs의 예는의 MediaTailor 설명서를 참조하세요 [DASH 광고 마커](#).

개인 맞춤형 MPDs의 주요 차이점

MediaTailor가 DASH MPDs 개인화하면 다음과 같은 몇 가지 중요한 변경이 이루어집니다.

기간 처리

- 광고 콘텐츠에 대한 새 기간이 생성됩니다.

- 기간 시작 시간은 타임라인 연속성을 유지하기 위해 조정됩니다.
- SCTE-35 마커가 있는 EventStream 요소가 처리 및 제거됩니다.

AdaptationSet 및 Representation 처리

- 광고 기간의 AdaptationSets는 AdaptationSets 콘텐츠와 일치하도록 생성됩니다.
- 광고 콘텐츠의 다양한 품질 수준에 대해 표현이 생성됩니다.
- 광고 콘텐츠를 가리키도록 SegmentTemplate 요소가 업데이트됨

이러한 변경 사항을 이해하면 MediaTailor 워크플로의 문제를 해결하고 CDN 및 플레이어의 적절한 구성을 보장하는 데 도움이 될 수 있습니다.

관련 주제

DASH MPDs MediaTailor

- [DASH 매니페스트 유형](#) - DASH 매니페스트 유형에 대한 자세한 설명
- [CDN을 사용하여 MediaTailor 광고 개인화 및 콘텐츠 전송 최적화](#) - MediaTailor에서 CDN을 사용하는 방법에 대한 정보
- [MediaTailor 광고 삽입 작동 방식](#) - MediaTailor 광고 삽입 작동 방식 개요
- DASH 매니페스트 구조 및 MediaPackage 구성에 대한 자세한 내용은 DASH 개요의 MediaPackage 사용 설명서 섹션을 참조하세요.

DASH 위치 기능

이 단원에서는 AWS Elemental MediaTailor에서 기본적으로 활성화되는 DASH의 위치 기능에 대한 정보를 제공합니다. MediaTailor 매니페스트에 액세스하기 위한 콘텐츠 전송 네트워크(CDN) 라우팅 규칙을 생성하는 경우 이 섹션을 참조하세요. 고정 HTTP 리디렉션을 지원하지 않는 플레이어와 함께 서버 측 보고를 사용하는 경우에도 이 단원을 읽습니다.

위치 기능이란 무엇입니까?

위치 기능을 통해 고정 HTTP 리디렉션을 지원하지 않는 플레이어는 매니페스트 업데이트 요청에서 고정 동작을 제공할 수 있습니다.

AWS Elemental MediaTailor 는 세션 없는 초기화를 사용하며 플레이어의 고정 HTTP 리디렉션 동작이 필요합니다. 서버 측 보고를 사용하면 플레이어가 MediaTailor에 대한 매니페스트 업데이트를 요청하면 서비스는 302 임시 리디렉션을 실행하여 플레이어를 개인화된 매니페스트의 엔드포인트로 전달합니다. MediaTailor는 응답에 세션 ID를 쿼리 파라미터로 포함합니다. 이는 플레이어가 세션의 전체에

대한 URL을 따르게 하기 위한 것이지만 고정 HTTP 리디렉션을 지원하지 않는 플레이어는 리디렉션을 중단하고 원래 URL로 복귀합니다. 플레이어가 원래 URL로 돌아가면 MediaTailor는 각 새 요청에 대해 원래 세션을 유지하지 않고 새 세션을 생성합니다. 이로 인해 매니페스트가 손상될 수 있습니다.

DASH 사양은 AWS Elemental MediaTailor 구성에서 기본적으로 활성화되는 위치 기능에서이 문제에 대한 솔루션을 제공합니다. 이 기능이 활성화되면 MediaTailor는 매니페스트 <Location> 태그에 절대 URL을 넣습니다. 고정 HTTP 리디렉션을 지원하지 않는 플레이어는 <Location>에 제공된 URL을 사용하여 매니페스트 업데이트를 요청할 수 있습니다.

구성에서 위치 기능을 비활성화해야 합니까?

위치 기능은 AWS Elemental MediaTailor 매니페스트에 액세스하기 위해 설정한 CDN 라우팅 규칙을 재정의하므로 이를 비활성화해야 할 수도 있습니다. 위치 기능은 콘텐츠 또는 광고 세그먼트의 CDN 캐싱에 영향을 주지 않습니다.

다음 목록에서 해당 상황을 찾아 구성에 대해 위치 기능을 비활성화해야 하는지 여부와 처리 방법을 결정합니다.

- AWS Elemental MediaTailor 매니페스트에 액세스하기 위해 설정된 CDN 라우팅 규칙이 없는 경우 위치 설정을 활성화 상태로 둡니다.
- 그렇지 않은 경우 다음 규칙을 따릅니다.
 - 서버 측 보고를 사용하지 않거나 플레이어가 모두 고정 HTTP 리디렉션을 지원하는 경우 위치 기능을 비활성화합니다. 콘솔에서 이를 수행하는 방법은 [the section called “구성 생성”](#) 단원을 참조하십시오.
 - 그렇지 않으면 [AWS Support](#)에 문의하십시오.

위치 기능을 사용해야 합니까?

고정 HTTP 리디렉션을 지원하지 않는 플레이어의 경우 위치 기능을 사용해야 합니다. 모든 매니페스트 업데이트 요청에 대해 <Location> 태그에 제공된 URL을 사용합니다.

예제

예제 URL 및 예제 <Location> 태그:

- Example예: 최초 요청 URL

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

- Example예: 리디렉션된 302 응답

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

- Example예: 매니페스트의 위치 태그

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

SigV4와의 AWS Elemental MediaTailor 오리진 상호 작용 보호

서명 버전 4(SigV4)는 HTTPS를 통해 지원되는 오리진에 대한 MediaTailor 요청을 인증하는 데 사용되는 서명 프로토콜입니다. MediaTailor는 HTTPS 통신만 지원하며 HTTP 연결을 허용하지 않습니다. SigV4 서명을 사용하면 MediaTailor 채널 어셈블리, Amazon S3 및 AWS Elemental MediaPackage 버전 2에 대한 HTTPS 오리진 요청에 MediaTailor가 서명된 권한 부여 헤더를 포함합니다.

오리진에서 SigV4를 사용하여 매니페스트 요청이 MediaTailor에서 전송되고 서명된 권한 부여 헤더가 포함된 경우에만 매니페스트 요청이 이행되도록 할 수 있습니다. 이렇게 하면 승인되지 않은 MediaTailor 재생 구성이 오리진 콘텐츠에 액세스할 수 없습니다. 서명된 권한 부여 헤더가 유효한 경우 오리진은 요청을 이행합니다. 유효하지 않으면 요청이 실패합니다.

다음 섹션에서는 지원되는 오리진에 MediaTailor SigV4 서명을 사용하기 위한 요구 사항을 설명합니다.

MediaTailor 채널 어셈블리 요구 사항

SigV4를 사용하여 MediaTailor 채널 어셈블리 오리진을 보호하는 경우 MediaTailor가 매니페스트에 액세스하려면 다음 요구 사항을 충족해야 합니다.

- MediaTailor 구성의 오리진 기본 URL은 다음 형식의 채널 어셈블리 채널이어야 합니다. `channel-assembly.mediatailor.region.amazonaws.com`
- HTTPS를 사용하도록 오리진을 구성해야 합니다. MediaTailor는 HTTPS 통신만 지원하며 HTTP 연결을 허용하지 않습니다. 오리진에서 HTTPS가 활성화되지 않은 경우 MediaTailor는 요청에 서명하지 않습니다.
- 채널에는 다음을 포함하는 오리진 액세스 정책이 있어야 합니다.

- MediaTailor가 채널에 액세스하기 위한 보안 주체 액세스. `mediatailor.amazonaws.com` 액세스 권한을 부여합니다.
- IAM 권한 `Mediatailor:GetManifest`는 MediaTailor 구성에서 참조하는 모든 다변량 재생 목록을 읽습니다.

채널에서 정책을 설정하는 방법에 대한 자세한 내용은 섹션을 참조하세요 [MediaTailor 콘솔을 사용하여 채널 생성](#).

Example MediaTailor 구성 계정으로 범위가 지정된 채널 어셈블리에 대한 오리진 액세스 정책

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

Example MediaTailor 재생 구성으로 범위가 지정된 채널 어셈블리에 대한 오리진 액세스 정책

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

Amazon S3 요구 사항

SigV4를 사용하여 Amazon S3 오리진을 보호하는 경우 MediaTailor가 매니페스트에 액세스하려면 다음 요구 사항을 충족해야 합니다.

- MediaTailor 구성의 오리진 기본 URL은 다음 형식의 S3 버킷이어야 합니다.
`s3.region.amazonaws.com`

- HTTPS를 사용하도록 오리진을 구성해야 합니다. MediaTailor는 HTTPS 통신만 지원하며 HTTP 연결을 허용하지 않습니다. 오리진에서 HTTPS가 활성화되지 않은 경우 MediaTailor는 요청에 서명하지 않습니다.
- 채널에는 다음을 포함하는 오리진 액세스 정책이 있어야 합니다.
 - MediaTailor가 버킷에 액세스하기 위한 보안 주체 액세스. `mediatailor.amazonaws.com` 액세스 권한을 부여합니다.

IAM에서 액세스를 구성하는 방법에 대한 자세한 내용은 AWS Identity and [Access Management](#) 사용 설명서의 액세스 관리를 참조하세요. AWS Identity and Access Management

- MediaTailor 구성에서 참조하는 모든 최상위 매니페스트를 읽을 수 있는 IAM 권한 `s3:GetObject`.

Amazon S3용 SigV4에 대한 일반적인 정보는 Amazon S3 API 참조의 [요청 인증\(AWS 서명 버전 4\)](#) 주제를 참조하세요. Amazon S3

Example MediaTailor 계정으로 범위가 지정된 Amazon S3에 대한 오리진 액세스 정책

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

Example MediaTailor 재생 구성으로 범위가 지정된 Amazon S3에 대한 오리진 액세스 정책

```
{
  "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 요구 사항

SigV4를 사용하여 MediaPackage v2 오리진을 보호하는 경우 MediaTailor가 매니페스트에 액세스하려면 다음 요구 사항을 충족해야 합니다.

- MediaTailor 구성의 오리진 기본 URL은 다음 형식의 MediaPackage v2 엔드포인트여야 합니다.
`mediapackagev2.region.amazonaws.com`
- HTTPS를 사용하도록 오리진을 구성해야 합니다. MediaTailor는 HTTPS 통신만 지원하며 HTTP 연결을 허용하지 않습니다. 오리진에서 HTTPS가 활성화되지 않은 경우 MediaTailor는 요청에 서명하지 않습니다.
- 채널에는 다음을 포함하는 오리진 액세스 정책이 있어야 합니다.
 - MediaTailor가 엔드포인트에 액세스하기 위한 보안 주체 액세스. `mediatailor.amazonaws.com` 액세스 권한을 부여합니다.
 - IAM 권한 `mediapackagev2:GetObject`는 MediaTailor 구성에서 참조하는 모든 다변량 재생 목록을 읽을 수 있습니다.

MediaPackage v2용 SigV4에 대한 일반적인 정보는 MediaPackage v2 API 참조의 [요청 인증\(AWS 서명 버전 4\)](#) 주제를 참조하세요.

Example MediaTailor 계정으로 범위가 지정된 MediaPackage v2에 대한 오리진 액세스 정책
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 MediaTailor 재생 구성으로 범위가 지정된 MediaPackage v2에 대한 오리진 액세스 정책
MediaTailor

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

```

    "Action": "mediapackagev2:GetObject",
    "Resource": "arn:aws:mediapackagev2:us-west-2:444455556666:channelGroup/emp-origin-channel-group/channel/emp-origin-channel/originEndpoint/emp-origin-endpoint",
    "Condition": {
        "StringEquals": {"AWS:SourceArn": "arn:aws:mediatailor:us-west-2:444455556666:playbackConfiguration/test"}
    }
}

```

Google의 광고 플랫폼 AWS Elemental MediaTailor 과 통합

MediaTailor는 Google Ad Manager(GAM), Google Campaign Manager(GCM), Google Display & Video 360(DV360)을 사용하여 server-to-server 연결을 자동으로 인증합니다. MediaTailor는 광고 요청 또는 추적 이벤트가 Google 광고 서버로 향하는 것을 감지하면 필요한 안전하고 인증된 연결을 자동으로 설정합니다.

관련 Google 광고 플랫폼으로 계정을 설정해야 합니다. MediaTailor는 다음 통합 유형을 지원합니다.

- 서버 측 통합: MediaTailor는 Google의 광고 플랫폼에 대한 서버 측 광고 요청 및 노출 추적 요청을 자동으로 보호합니다. 여기에는 다음이 포함됩니다.
 - Google Ad Manager(GAM): 게시자에 대한 Google의 광고 서버에 대한 광고 요청은 자동으로 보호되며, 이는 Google의 실시간 광고 판매 마켓플레이스 및 광고 교환인 공인 구매자에게 액세스하는 데 필요합니다.
 - Google Campaign Manager(GCM): 광고 서비스 및 추적 요청은 Google의 인증된 엔드포인트 및 보안을 통해 자동으로 라우팅되며, 보다 정확한 보고와 거부된 노출 감소로 이러한 플랫폼에서 캠페인을 실행하는 광고자를 지원합니다.
 - 디스플레이 및 비디오 360(DV360): 광고 요청은 Google의 인증된 엔드포인트 및 보안을 통해 자동으로 라우팅되며, 더 정확한 보고와 거부된 노출 감소로 이러한 플랫폼에서 캠페인을 실행하는 광고자를 지원합니다.
- 클라이언트 측 통합: Google 프로그래밍 방식 액세스 라이브러리(PAL) SDK를 사용한 플레이어 통합입니다. Google Ad Manager에서 공개 경매 트랜잭션 유형을 사용하려면이 통합이 필요합니다.

다른 모든 광고 요청은 수정 없이 계속 작동합니다.

프로그래밍 방식 트랜잭션 유형에 대한 Ad Manager 지원은 사용 중인 통합 유형에 따라 다릅니다. 사용 가능한 옵션 목록은 [트랜잭션 유형](#)을 참조하거나 Google 계정 팀에 문의하세요.

주제

- [Google의 광고 플랫폼과 서버 측 AWS Elemental MediaTailor 통합](#)
- [Google Ad Manager와의 클라이언트 측 AWS Elemental MediaTailor 통합](#)
- [Google Ad Manager를 사용하여 광고 채우기 속도 최적화](#)

Google의 광고 플랫폼과 서버 측 AWS Elemental MediaTailor 통합

MediaTailor는 Google에서 발급한 SSL 인증서를 사용하여 Google Ad Manager(GAM), Google Campaign Manager(GCM) 및 Google Display & Video 360(DV360)과의 server-to-server 연결을 자동으로 인증합니다. MediaTailor는 Google 광고 서버로 향하는 광고 요청 또는 추적 이벤트를 감지하면 필요한 안전하고 인증된 연결을 자동으로 설정합니다. 추가 SSL 인증서 설정 또는 지원 티켓은 필요하지 않습니다.

사전 조건: 관련 Google 광고 플랫폼(Google Ad Manager, Google Campaign Manager 또는 Display & Video 360)이 있는 활성 계정이 있어야 합니다.

MediaTailor는 다음 대상 URI 규칙을 기반으로 Google에서 발급한 SSL 인증서 인증을 적용합니다.

Google Ad Manager(GAM): 서버 측 광고 요청

- 이 규칙은 해당 도메인으로 ADS 템플릿 URL을 구성하거나 리디렉션 광고 요청이 확인되는 경우를 `serverside.doubleclick.net` 포함하여에 대한 광고 요청에 대해 트리거됩니다.
- MediaTailor는 `ssss=mediatailor` 쿼리 파라미터가 없는 경우 쿼리 파라미터를 추가한 다음 Google에서 발급한 SSL 인증서를 사용하여 연결을 보호합니다.
- 이를 통해 Google의 실시간 광고 판매 마켓플레이스 및 광고 교환인 공인 구매자에게 액세스할 수 있습니다.

Google Ad Manager(GAM): VMAP 내에서 리디렉션된 광고 요청

- 이 규칙은 MediaTailor가 이전에 현재 요청 `serverside.doubleclick.net`에서 로 VAST `pubads.g.doubleclick.net` 리디렉션을 보고 이전 ADS 응답이 VMAP 응답인 경우 리디렉션된 광고 요청에 대해 트리거됩니다.
- MediaTailor는 도메인을 로 자동 변경 `serverside.doubleclick.net`하고 Google에서 발급한 SSL 인증서를 사용하여 인증된 연결을 설정합니다.
- 이렇게 하면 VMAP 기반 광고 워크플로에 대해 승인된 구매자에 액세스할 수 있습니다.

Google Campaign Manager(GCM): 요청 광고 서비스 및 추적

- 이 규칙은 URI에 `/ddm/pfadx/` 또는 `/ddm/trackimp/` (암호화 또는 비암호화)가 포함된 경우 `ad.doubleclick.net` 광고 서비스 및 추적 요청에 대해 트리거됩니다.

- MediaTailor는 도메인을 로 자동 변경serverside.doubleclick.net하고 Google에서 발급한 SSL 인증서를 사용하여 인증된 연결을 설정합니다.
- 이는 보다 정확한 보고와 거부된 노출 감소로 이러한 플랫폼에서 캠페인을 실행하는 광고주를 지원합니다.

디스플레이 및 비디오 360(DV360): 광고 요청

- 이 규칙은 URI에 /dbm/vast/ (암호화 또는 비암호화)가 포함된 경우 bid.g.doubleclick.net 또는 pagead2.googlesyndication.com 광고 요청에 대해 트리거됩니다.
- MediaTailor는 도메인을 로 자동 변경serverside.doubleclick.net하고 Google에서 발급한 SSL 인증서를 사용하여 인증된 연결을 설정합니다.
- 이렇게 하면 Display & Video 360 캠페인에 대해 인증된 광고 서비스가 활성화됩니다.

다른 모든 광고 요청은 Google에서 발급한 SSL 인증서 인증 없이 처리됩니다.

Note

SSL 인증서 인증은 이제 자동입니다. SSL 인증서를 활성화하기 위해 지원 티켓을 이전에 제출한 경우 별도의 조치가 필요하지 않습니다. 구성은 계속 작동합니다.

Google Ad Manager와의 클라이언트 측 AWS Elemental MediaTailor 통합

Google Ad Manager 프로그래밍 방식 액세스 라이브러리(PAL) SDKs를 사용하고 Google Ad Manager의 공개 경매 트랜잭션 유형을 사용하려면 MediaTailor 클라이언트 측 통합이 필요합니다.

PAL SDKs는 재생 세션의 콘텐츠, 디바이스 및 사용자 데이터에 대한 정보를 제공합니다. PAL SDK를 통해 Google Ad Manager에이 정보를 제공하면 표시할 대상 광고를 더 잘 결정할 수 있습니다. SDKs Android, iOS, HTML5 및 Cast에서 사용할 수 있습니다. PAL SDKs. <https://developers.google.com/ad-manager/pal>

Ad Manager와의 클라이언트 측 통합을 생성하려면

1. PAL SDK를 사용하여 nonce를 생성합니다. nonce 설정에 대한 자세한 내용은 [Google Ad Manager PAL SDK의 SDK](#) 설명서를 참조하세요.

Note

nonce는 PAL이 스트림 요청에 대해 생성하는 암호화된 문자열입니다. 각 요청에는 고유한 nonce가 있어야 합니다.

2. ADS 요청에서 givn 파라미터를 사용하여 nonce 값을 전달합니다. 이렇게 하려면 포함하도록 ADS URL을 업데이트합니다 &givn=[*player_params.givn*]. 지침은 [클라이언트 측 추적 활성화](#) 섹션을 참조하세요.

Datazoom 플레이어 SDKs

MediaTailor는 Datazoom과 협력하여 Google Ad Manager PAL에서 제공되는 SDKs와 같은 SDK와의 통합을 용이하게 하기 위해 무료 플레이어 SDKs를 제공합니다. Datazoom 및 MediaTailor 파트너십에 대한 자세한 내용은 섹션을 참조하세요 [Datazoom 프리 플레이어 SDKs](#). Datazoom 플레이어 SDKs에 액세스하려면 사이트와 [함께 Datazoom AWS](#)의 연락처 정보를 사용합니다.

Google Ad Manager를 사용하여 광고 채우기 속도 최적화

MediaTailor에서 Google Ad Manager(GAM)를 광고 서버로 사용하면 ADS 템플릿 URL 파라미터가 광고 채우기 비율 및 수익에 직접적인 영향을 미칩니다. 잘못된 파라미터 값은 채우기 속도가 낮은 가장 일반적인 원인 중 하나입니다.

권장 GAM 파라미터

ADS 템플릿 URL에 다음 파라미터를 포함하여 GAM이 광고를 일관되게 반환하고 광고 시간을 전체 기간으로 채우도록 합니다.

`correlator=[avail.random]`

GAM이 각 광고 시간을 고유한 요청으로 취급하고 새로운 광고를 반환하도록 합니다. 이 파라미터가 없으면 GAM은 동일한 세션 내에서 반복되는 광고 시간에 대해 빈 응답을 반환할 수 있습니다.

`pmxd=[session.avail_duration_ms]`

밀리초 단위의 최대 광고 포드 기간입니다. [session.avail_duration_ms]이 값이 실제 SCTE 광고 마커 기간과 자동으로 일치하도록 MediaTailor 동적 변수를 사용합니다. 이 값을 하드코딩하지 마십시오. 하드코딩된 기간이 실제 광고 시간보다 짧으면 남은 시간이 채워지지 않습니다.

`pmad=-1`

포드당 최대 광고 수입입니다. 가능한 한 많은 광고가 브레이크를 채우-1도록 허용하려면이 값을 로 설정하거나 원하는 제한(예: 10)으로 설정합니다. 낮은 값(예: 4)은 더 많은 인벤토리를 사용할 수 있더라도 휴식 시간을 채울 수 있는 광고 수를 인위적으로 제한합니다.

`vpos`

비디오 위치(프리롤, 미드롤 또는 포스트롤). 실제 광고 슬롯 유형과 일치하는 값으로 설정합니다. 사용 안 함 [`avail.type`]- MediaTailor는이 변수를 지원하지 않으며 빈 값으로 확인되어 GAM이 요청을 거부하거나 낮은 값 인벤토리를 반환할 수 있습니다.

GAM의 플레이어 파라미터

일부 GAM 파라미터에는 게시자 또는 플레이어 애플리케이션이 세션 초기화 시 제공하는 값이 필요합니다. 이를 MediaTailor에 플레이어 파라미터로 전달하고 ADS 템플릿 URL에서 참조합니다. 플레이어 파라미터 구성에 대한 자세한 내용은 섹션을 참조하세요 [ADS 요청에 대한 MediaTailor 플레이어 변수](#).

`ppid=[player_params.ppid]`

프로그래밍 방식 게시자 제공 식별자입니다. GAM이 세션을 추적하고 각 광고 시간에 광고를 계속 반환할 수 있도록이 게시자 할당 값을 제공합니다. 이렇게 하지 않으면 GAM은 세션의 첫 번째 휴식 후 광고 반환을 중지할 수 있습니다.

`rdid=[player_params.rdid]` (재설정 가능한 디바이스 식별자)

디바이스 광고 ID입니다. CTV 및 스마트 TV 디바이스의 경우 프로그래밍 방식 광고 구매자는 인벤토리에 입찰하려면이 식별자가 필요합니다. 그렇지 않으면 CTV 디바이스의 채우기 속도가 크게 낮아질 수 있습니다.

`msid=[player_params.msid]` (앱 ID)

애플리케이션 식별자입니다. 와 마찬가지로 `rdid`이는 CTV 및 스마트 TV 디바이스 타겟팅을 위한 프로그래밍 방식 구매자에게 필요합니다.

GAM이 지원하는 디바이스 대상 지정 파라미터에 대한 자세한 내용은 [Google 웹 사이트의 Google Ad Manager 대상 지정 파라미터를](#) 참조하세요.

일반적인 구성 실수

다음은 GAM에서 채우기 속도가 낮은 일반적인 원인입니다.

- 하드코딩 **pmxd**- 실제 광고 시간이 더 길 때(예: 90초pmxd=60000) 고정 값(예:)을 사용하면 남은 시간이 채워지지 않습니다. 대신 [session.avail_duration_ms]을 사용하세요.
- 낮은 **pmad** 값 - 포드당 최대 광고를 너무 낮게 설정하면 인벤토리를 사용할 수 있는 경우에도 GAM이 전체 휴식 기간을 채우지 못합니다. -1를 무제한으로 사용하거나 원하는 한도를 설정합니다.
- 누락 **correlator**-이 파라미터가 없으면 GAM은 동일한 세션 내에서 반복되는 광고 시간에 대해 빈 VAST 응답을 반환할 수 있습니다.
- 누락 **ppid**-이 플레이어 파라미터가 없으면 GAM은 세션의 첫 번째 광고 중단 후 광고 반환을 중지할 수 있습니다.
- CTV 디바이스에서 디바이스 ID 누락 - 없이 rdid프로그래밍 방식 구매자는 디바이스를 대상으로 지정할 수 없으므로 CTV 인벤토리 채우기가 크게 줄어듭니다.
- 사용 **vpos=[avail.type]**—MediaTailor는 [avail.type] 변수를 지원하지 않습니다. 이렇게 하면 빈 문자열로 확인되어 요청 실패 또는 저가 광고 응답이 발생할 수 있습니다.
- 세션 초기화 시 플레이어 파라미터가 전달되지 않음 - ADS 템플릿 URL이 플레이어 파라미터(예: [player_params.ppid])를 참조하지만 세션 초기화 호출이 해당 값을 전달하지 않습니다. GAM은 해당 매크로에 대해 빈 값 또는 null 값을 수신하므로 광고 요청이 채우기 없음을 반환할 수 있습니다.

광고 중단 억제를 사용하여 광고 중단 동작 사용자 지정

에서 구성을 생성할 때 광고 중단 억제를 구성하는 기능을 포함하여 광고 중단의 동작을 제어하는 선택적 광고 중단 구성 설정을 지정할 AWS Elemental MediaTailor 수 있습니다. 이를 통해 특정 요구 사항에 맞게 비디오 콘텐츠에 대한 광고 시간 경험을 조정할 수 있습니다.

호환성 제한

다음과 함께 광고 중단 금지를 사용할 수 없습니다.

- VOD 및 live-to-VOD 워크플로. 라이브 워크플로만 지원됩니다.
- 서버 가이드 광고 삽입(SGAI) 메서드. 서버 안내 메서드는 광고 결정을 다르게 처리하며 억제 구성이 필요하지 않습니다.

주제

- [광고 중단 억제 구성](#)

광고 중단 억제 구성

라이브 콘텐츠에 대한 광고 시간 개인화를 건너뛰도록 MediaTailor를 구성할 수 있습니다. 이를 광고 중단 억제 또는 가용성 억제라고 합니다. 이 주제에서는 방법을 보여주고 광고 중단 억제 구성의 작동 방식도 설명합니다.

광고 중단 억제는 다음 사용 사례에 사용할 수 있습니다.

- 긴 매니페스트 록백 윈도우 - 시청자가 매니페스트의 라이브 끝에서 재생을 시작하지만 록백 윈도우가 큰 경우 시청자가 시청을 시작한 후 시작하는 광고만 삽입하려고 할 수 있습니다. 또는 매니페스트에 전체 록백 윈도우의 일부에 대한 광고를 삽입하려고 할 수 있습니다. MediaTailor가 라이브 엷지 뒤의 지정된 시간 범위 내에서 광고 시간을 개인화하도록 광고 억제를 구성할 수 있습니다.
- 중간 참여 - 시청자가 광고 시간 도중에 라이브 비디오 스트림 시청을 시작하는 경우 해당 사용자는 채널을 바꾸고 광고를 시청하지 않을 수 있습니다. 광고 억제를 사용하면 최종 사용자가 스트림에 조인하기 전에 광고 시간이 시작된 경우 광고 시간 개인화를 건너뛸 수 있습니다.

광고 억제 구성

광고 억제를 사용하려면 다음과 같은 방법으로 가용 억제 모드, 가용 억제 값 및 가용 억제 채우기 정책을 구성합니다.

- MediaTailor 콘솔에서
- AWS Command Line Interface (AWS CLI) 사용
- MediaTailor API를 사용하거나 클라이언트의 재생 세션 요청에서 파라미터로 사용

파라미터를 사용한 구성에 대한 자세한 내용은 섹션을 참조하세요 [광고 억제 파라미터 구성 - 재생 세션 요청](#).

광고 억제 구성 파라미터

광고 금지를 켜거나 끄도록 선택할 수 있습니다. 광고 금지를 켜면 라이브 재생 엷지 이후 또는 라이브 스트림의 라이브 재생 엷지 이전에 해당 금지가 발생하는지 여부를 지정합니다. 어느 경우든 라이브 엷지를 기준으로 시간을 지정합니다. 여기서 MediaTailor는 광고를 개인화하지 않습니다. 가용성 억제를 켜면 세션이 중단 중간에 시작될 때 MediaTailor가 부분 광고 중단 채우기에 사용하는 가용성 억제 정책을 지정할 수 있습니다.

다음은 광고 억제 구성 파라미터입니다.

- 가능 구간 억제 모드 - 광고 억제 모드를 설정합니다. 기본적으로 광고 억제는 해제되어 있습니다. 허용되는 값: OFF, BEHIND_LIVE_EDGE 또는 AFTER_LIVE_EDGE.
- OFF: 광고 억제가 없으며 MediaTailor는 모든 광고 시간을 개인화합니다.
- BEHIND_LIVE_EDGE: MediaTailor는 라이브 엣지 이전에 시작되는 광고 시간을 가용성 억제 값을 뺀 값으로 개인화하지 않습니다. 이는 개별 광고 시간뿐만 아니라 전체 광고 시간에도 영향을 미칩니다.
- AFTER_LIVE_EDGE: MediaTailor는 라이브 엣지 내에 있는 광고 시간 및 가용성 억제 값을 개인화하지 않습니다. 이는 전체 광고 시간에 영향을 미치거나 광고 시간의 부분 채우기를 허용하도록 구성할 수 있습니다.
- 가능 구간 억제 값 - 라이브 스트림의 라이브 끝에 상대적인 시간입니다. 허용되는 값: HH:MM:SS의 시간 값입니다.
- 가용성 억제 채우기 정책 - MediaTailor가 가용성 억제 모드에 적용하는 정책을 정의합니다. 허용되는 값: PARTIAL_AVAIL, FULL_AVAIL_ONLY.
- BEHIND_LIVE_EDGE 모드는 항상 FULL_AVAIL_ONLY 억제 정책을 사용합니다.
- AFTER_LIVE_EDGE 모드는 세션이 중단 중간에 시작될 때 PARTIAL_AVAIL 광고 중단 채우기를 호출하는 데 사용할 수 있습니다.

광고 억제 설정 예제

[광고 억제 구성 파라미터](#)가 서로 상호 작용하는 방식을 사용하면 라이브 스트림의 라이브 엣지 이전, 현재 또는 이후에 광고 억제 및 가용성 채우기를 처리하는 여러 가지 방법을 지정할 수 있습니다. 이 섹션에서는 이러한 상호 작용 중 일부를 보여주는 예제를 제공합니다. 다음 예제를 사용하면 특정 상황에 맞게 구성 파라미터를 설정하는 데 도움이 됩니다.

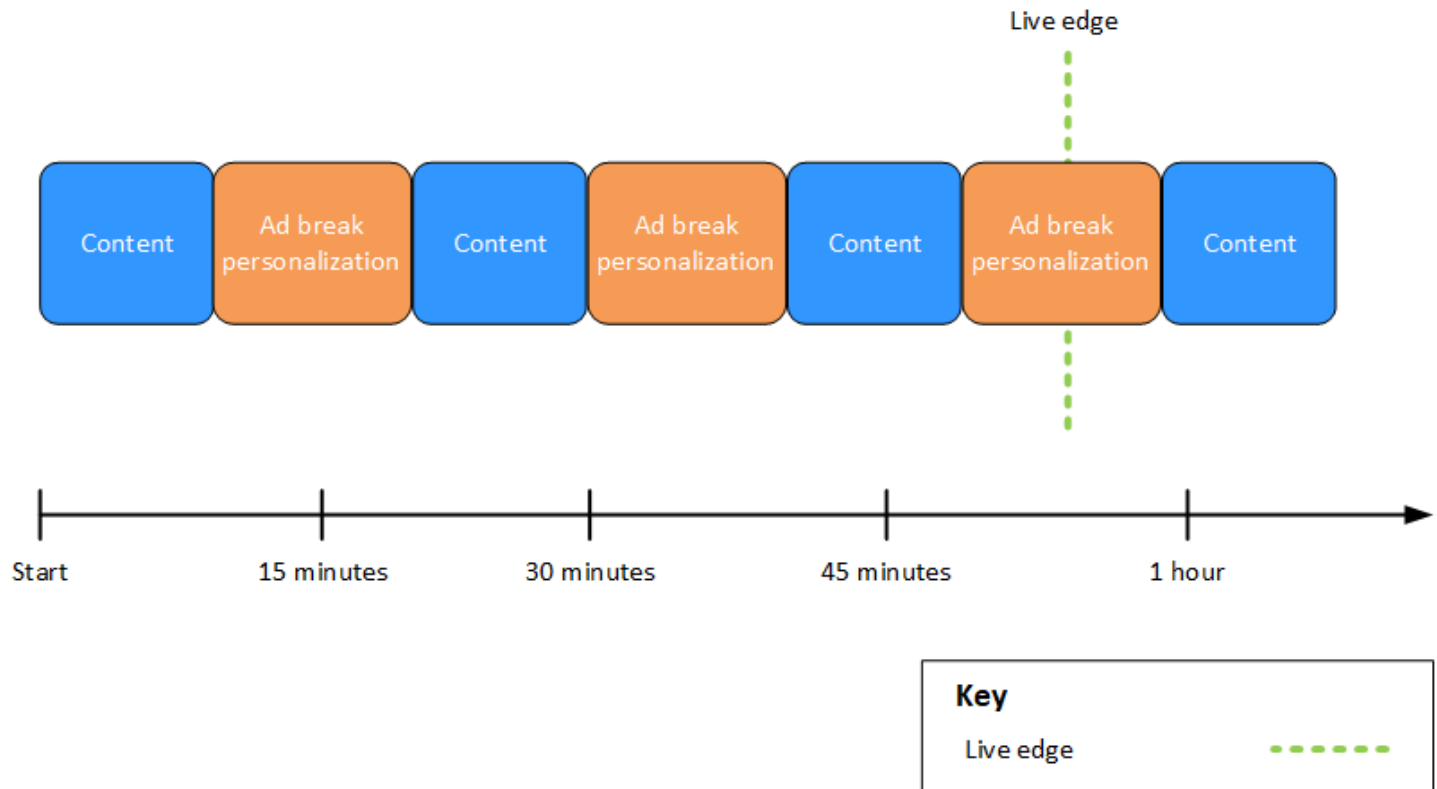
다음은 광고 억제 설정의 예입니다.

Example 1: 광고 억제 없음

가용 금지 모드가 OFF인 경우 광고 금지가 없으며 MediaTailor는 모든 광고 시간을 개인화합니다.

다음 그림에서는 왼쪽에서 오른쪽으로 진행되는 타임라인을 따라 다양한 블록이 가로로 배열됩니다. 각 블록은 라이브 스트림의 콘텐츠 또는 맞춤형 광고 시간이 재생되는 시간을 나타냅니다. 점선은 라이브 스트림의 현재 라이브 엣지를 나타냅니다. 라이브 엣지 전에 두 번의 광고 시간이 발생하고 라이브 엣지에서 또 다른 광고 시간이 진행 중입니다. 그림과 같이 가용 억제 모드가 인 경우 OFF MediaTailor는 타임라인의 라이브 엣지 이전에 발생하는 모든 광고 시간을 개인화합니다. 또한 MediaTailor는 라이브 엣지에서 진행 중인 광고 시간을 개인화합니다.

Avail suppression mode (default): OFF



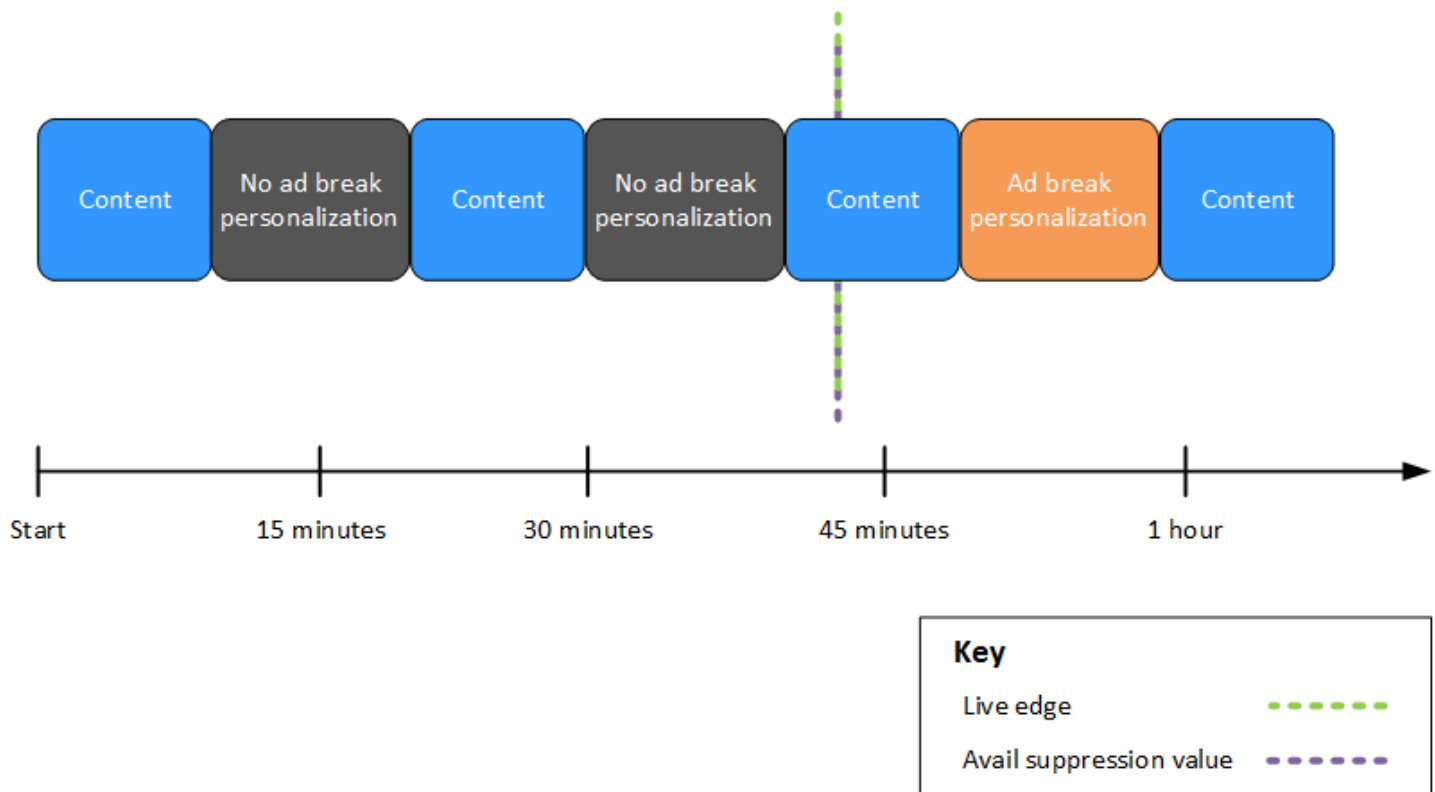
Example 2: 라이브 엣지와 동기화된 값이 있는 **BEHIND_LIVE_EDGE** 광고 억제

가용 억제 모드를 로 설정하고 **BEHIND_LIVE_EDGE** 가용 억제 값을 로 설정하면 00:00:00가용 억제 값이 라이브 엣지와 동기화됩니다. MediaTailor는 라이브 엣지에서 또는 그 이전에 시작되는 광고 시간을 개인화하지 않습니다.

다음 그림에서는 왼쪽에서 오른쪽으로 진행되는 타임라인을 따라 다양한 블록이 가로로 배열됩니다. 각 블록은 라이브 스트림의 콘텐츠, 개인화된 광고 시간 또는 개인화되지 않은 광고 시간이 재생되는 시간을 나타냅니다. 점선은 라이브 스트림의 현재 라이브 엣지를 나타냅니다. 로 설정된 가용 억제 값을 나타내는 또 다른 점선은 라이브 엣지의 점선과 00:00:00겹칩니다. 라이브 엣지 전에 두 번의 광고 중단이 발생하고 라이브 엣지 후에 또 다른 광고 중단이 발생합니다. 그림과 같이, 가용 억제 모드가 로 설정BEHIND_LIVE_EDGE되고 가용 억제 값이 라이브 엣지와 동기화00:00:00되도록 로 설정되면 MediaTailor는 타임라인의 라이브 엣지 이전에 발생하는 광고 시간을 개인화하지 않습니다. MediaTailor는 라이브 엣지 이후에 발생하는 광고 시간을 개인화합니다.

Avail suppression mode: **BEHIND_LIVE_EDGE**

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



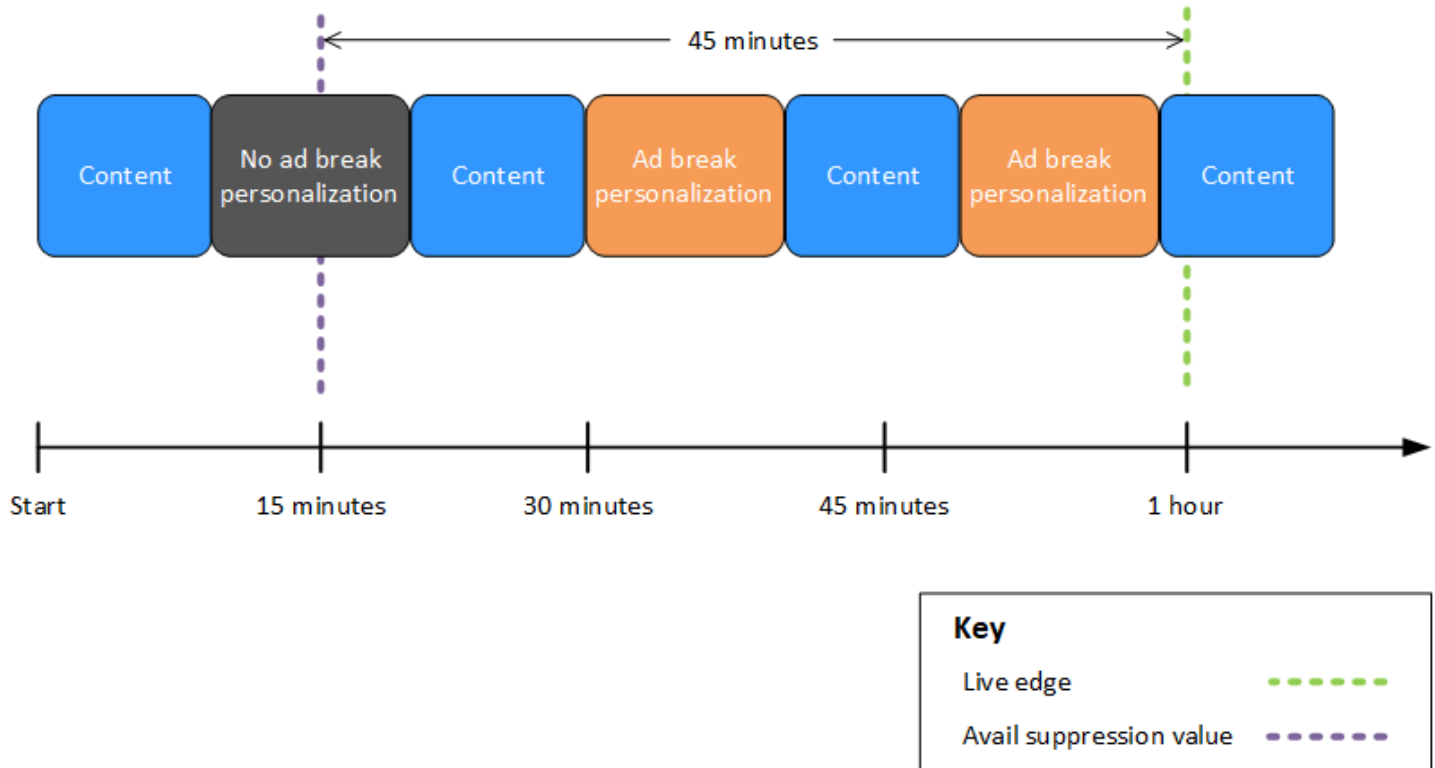
Example 3: 라이브 엣지 뒤에 값이 있는 **BEHIND_LIVE_EDGE** 광고 억제

가용성 억제 모드가 로 설정된 경우 **BEHIND_LIVE_EDGE** MediaTailor는 해당 시간 또는 그 이전에 광고 시간을 개인화하지 않습니다. 이 예제에서 MediaTailor는 라이브 엣지에서 최대 45분 뒤에 시작하는 광고 시간을 개인화합니다. MediaTailor는 라이브 엣지보다 45분 이상 늦게 시작하는 광고 시간을 개인화하지 않습니다.

다음 그림에서는 왼쪽에서 오른쪽으로 진행되는 타임라인을 따라 다양한 블록이 가로로 배열됩니다. 각 블록은 라이브 스트림의 콘텐츠, 개인화된 광고 시간 또는 개인화되지 않은 광고 시간이 재생되는 시간을 나타냅니다. 점선은 라이브 스트림의 현재 라이브 엣지를 나타냅니다. 로 설정된 가용 억제 값을 나타내는 또 다른 점선은 라이브 엣지의 점선을 기준으로 타임라인의 45분 전에 00:45:00 발생합니다. 점선 사이의 45분 기간은 가용 억제 기간을 나타냅니다. 가용성 억제 기간이 시작될 때 광고 시간이 진행 중입니다. 가용성 억제 기간 동안 다른 두 개의 광고 중단이 발생합니다. 그림과 같이 가용 억제 모드가 로 설정 **BEHIND_LIVE_EDGE** 되고 가용 억제 값이 라이브 엣지 00:45:00 뒤에서 로 설정되면 MediaTailor는 가용 억제 기간 내에 발생하는 모든 광고 시간을 개인화합니다. MediaTailor는 가용성 억제 기간이 시작될 때 진행 중인 광고 시간을 개인화하지 않습니다.

Avail suppression mode: **BEHIND_LIVE_EDGE**

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



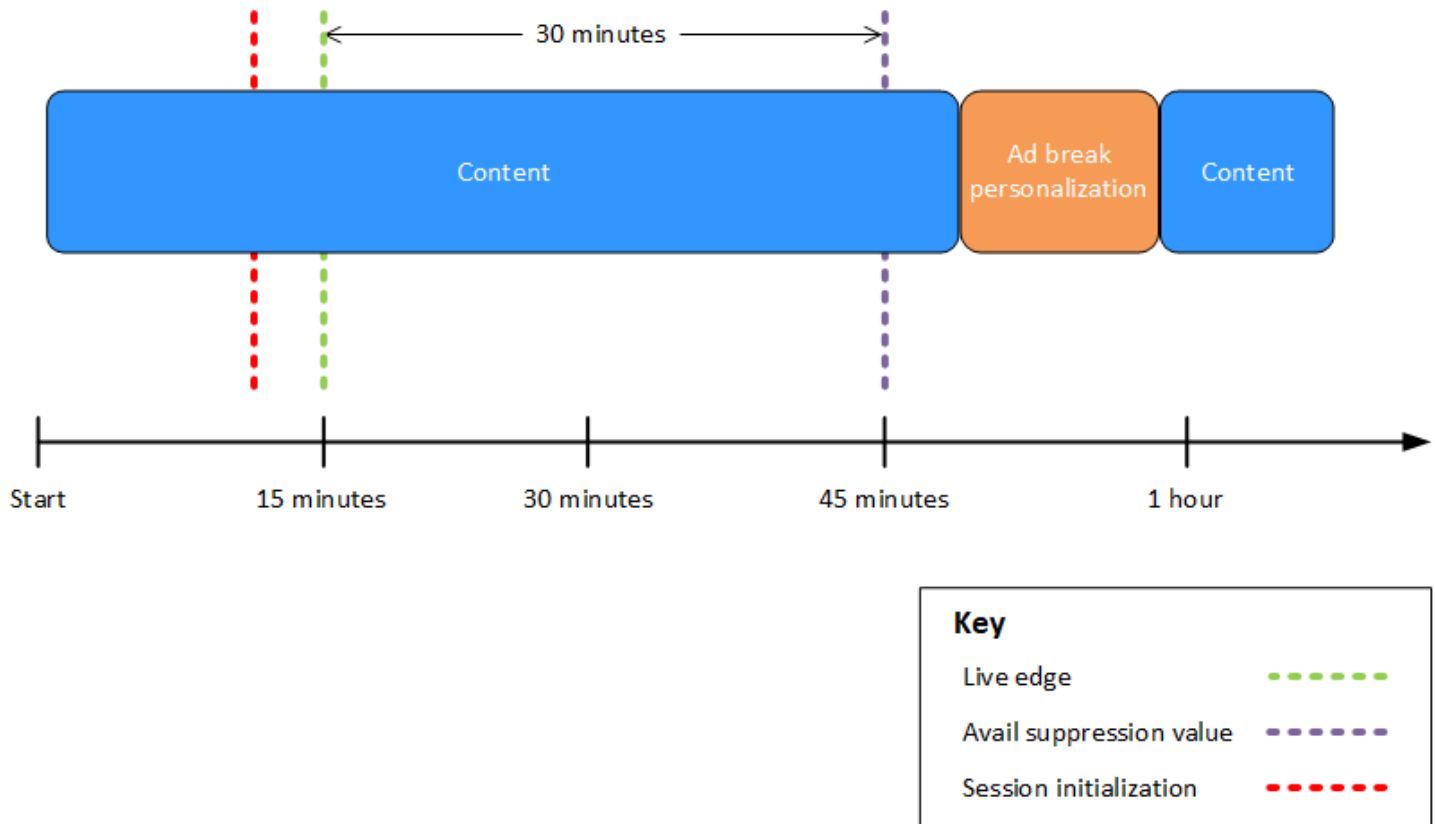
Example 4: 가용성 억제 기간 동안 **AFTER_LIVE_EDGE** 광고 중단이 발생하지 않는 광고 억제

가용 억제 모드가 로 설정 **AFTER_LIVE_EDGE** 되고 가용 억제 값이 0보다 크면 MediaTailor 는 세션의 경과 시간이 해당 값에 도달할 때까지 광고 시간을 개인화하지 않습니다.

다음 그림에서는 왼쪽에서 오른쪽으로 진행되는 타임라인을 따라 다양한 블록이 가로로 배열됩니다. 각 블록은 라이브 스트림의 콘텐츠 또는 맞춤형 광고 시간이 재생되는 시간을 나타냅니다. 점선은 라이브 스트림의 현재 라이브 엣지를 나타냅니다. 로 설정된 가용 억제 값을 나타내는 또 다른 점선은 라이브 엣지의 점선을 기준으로 타임라인에서 30분 후에 **00:30:00** 발생합니다. 세션 초기화를 나타내는 세 번째 점선은 라이브 엣지의 점선을 기준으로 타임라인 앞부분에서 발생합니다. 라이브 엣지 시간과 avail-suppression-value 시간 사이의 30분 기간은 가용 억제 기간을 나타냅니다. 광고 중단은 가용 억제 기간 후에 발생합니다. 그림과 같이, 가용 억제 모드가 로 설정되면 **AFTER_LIVE_EDGE** 가용 억제 값은 라이브 엣지 **00:30:00** 뒤에 로 설정되고 세션 초기화는 라이브 엣지 앞에 발생합니다. MediaTailor는 가용 억제 기간 이후에 발생하는 모든 광고 시간을 개인화합니다.

Avail suppression mode: **AFTER_LIVE_EDGE**

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



Example 5: 가용성 **AFTER_LIVE_EDGE 억제 기간이 끝날 때
PARTIAL_AVAIL 채우기 정책 및 진행 중인 광고 시간을 포함한 광고 억제**

가용 억제 모드가 로 설정 **AFTER_LIVE_EDGE** 되고 가용 억제 값이 0보다 크면 MediaTailor 는 세션의 경과 시간이 해당 값에 도달할 때까지 광고 시간을 개인화하지 않습니다.

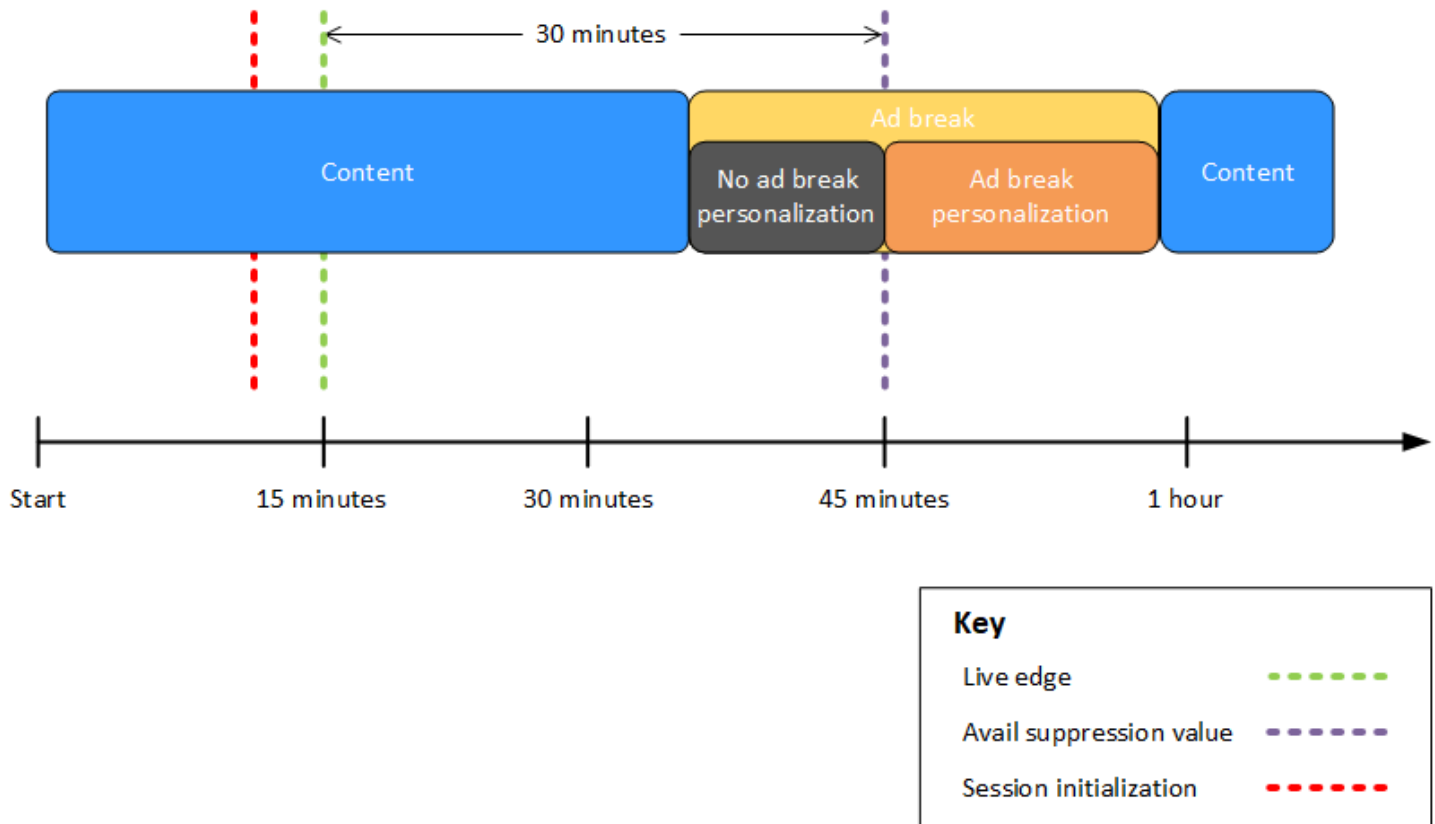
다음 그림에서는 왼쪽에서 오른쪽으로 진행되는 타임라인을 따라 다양한 블록이 가로로 배열됩니다. 각 블록은 라이브 스트림의 콘텐츠, 개인화된 광고 시간 또는 개인화되지 않은 광고 시간이 재생되는 시간을 나타냅니다. 점선은 라이브 스트림의 현재 라이브 엣지를 나타냅니다. 로 설정된 가용 억제 값을 나타내는 또 다른 점선은 라이브 엣지의 점선을 기준으로 타임라인에서 30분 후에 00:30:00 발생합니다. 세션 초기화를 나타내는 세 번째 점선은 라이브 엣지의 점선을 기준으로 타임라인 앞부분에서 발생합니다. 라이브 엣지 시간과 avail-suppression-value 시간 사이의 30분 기간은 가용 억제 기간을 나타냅니다. 가용성 억제 기간이 끝날 때 광고 시간이 진행 중입니다. 그림과 같이, 가용 억제 모드가 로 설정되면 **AFTER_LIVE_EDGE** 가용 억제 값은 라이브 엣지 00:30:00 뒤에 로 설정되

고, 가용 억제 채우기 정책은 로 설정되며PARTIAL_AVAIL, 세션 초기화는 라이브 엣지 앞에 발생합니다. MediaTailor는 가용 억제 기간 이후에 발생하는 광고 종단을 개인화합니다. 가용 억제 기간이 끝날 때 진행 중인 광고 시간의 경우 MediaTailor는 가용 억제 기간 이후에 발생하는 해당 광고 시간의 일부를 개인화하지만, 가용 억제 기간 중에 발생하는 해당 광고 시간의 부분은 개인화하지 않습니다.

Avail suppression mode: **AFTER_LIVE_EDGE**

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

Avail suppression fill policy: **PARTIAL_AVAIL**



Example 6: 세션 초기화 전부터 가용 억제 기간 종료 후까지 진행 중인
AFTER_LIVE_EDGE 광고 종단 및 **PARTIAL_AVAIL** 채우기 정책 포함 광고 억제

가용 억제 모드가 로 설정AFTER_LIVE_EDGE되고 가용 억제 값이 0보다 크면 MediaTailor는 세션의 경과 시간이 해당 값에 도달할 때까지 광고 시간을 개인화하지 않습니다.

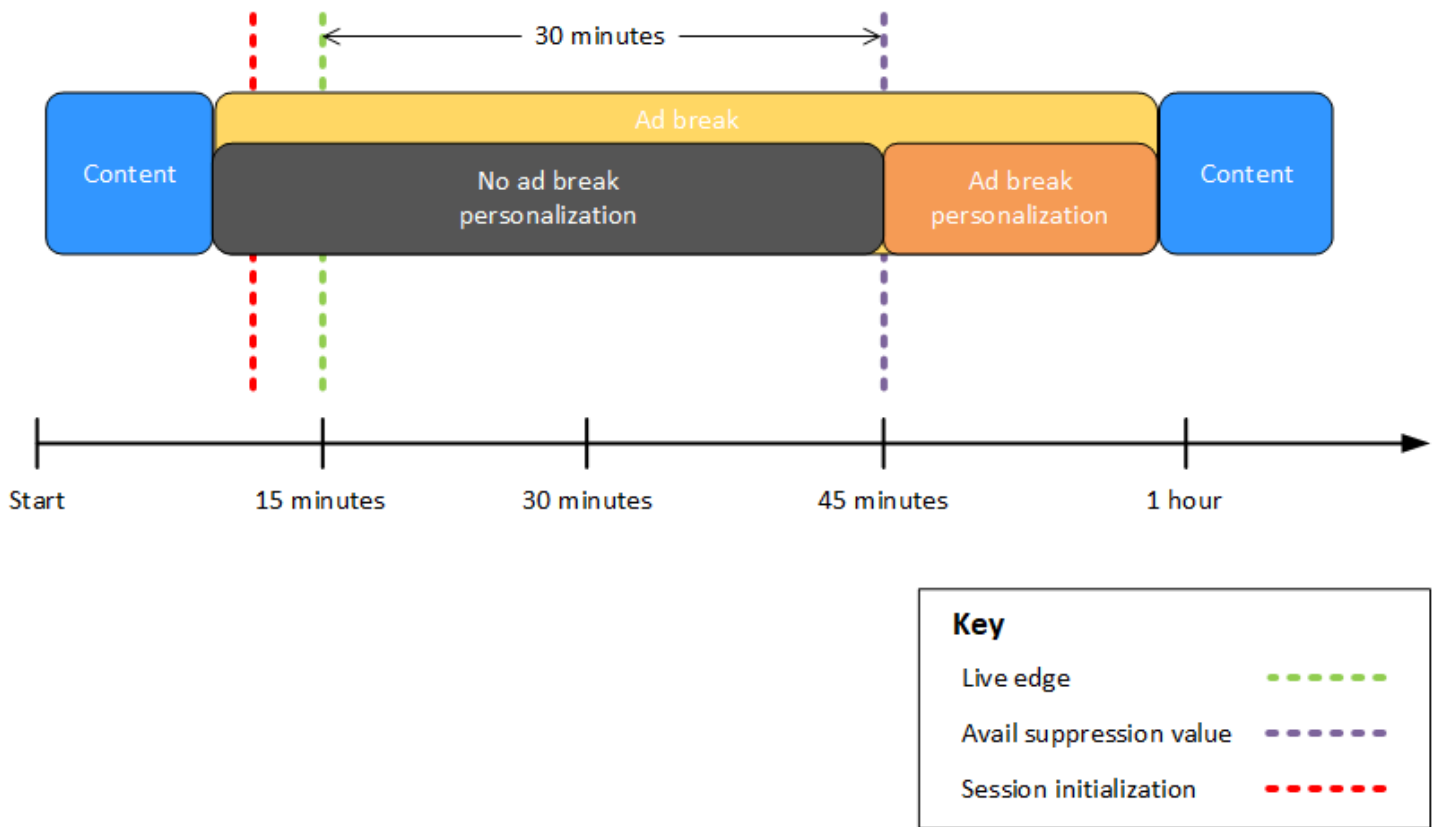
다음 그림에서는 왼쪽에서 오른쪽으로 진행되는 타임라인을 따라 다양한 블록이 가로로 배열됩니다. 각 블록은 라이브 스트림의 콘텐츠, 개인화된 광고 시간 또는 개인화되지 않은 광고 시간이 재생되는 시간을 나타냅니다. 점선은 라이브 스트림의 현재 라이브 엣지를 나타냅니다. 로 설정된 가용 억제 값을 나타내는 또 다른 점선은 라이브 엣지의 점선을 기준으로 타임라인에서 30

분 후에 00:30:00 발생합니다. 세션 초기화를 나타내는 세 번째 점선은 라이브 엣지의 점선을 기준으로 타임라인 앞부분에서 발생합니다. 라이브 엣지 시간과 avail-suppression-value 시간 사이의 30분 기간은 가용 억제 기간을 나타냅니다. 광고 시간이 세션 초기화 전 시간부터 가용 억제 기간 후 시간까지 진행 중입니다. 그림과 같이, 가용 억제 모드가 로 설정되면 AFTER_LIVE_EDGE가 용 억제 값은 라이브 엣지 00:30:00 뒤에 로 설정되고, 가용 억제 채우기 정책은 로 설정되며 PARTIAL_AVAIL, 세션 초기화는 라이브 엣지 앞에 발생합니다. MediaTailor는 가용 억제 기간 이후에 발생하는 광고 종단을 개인화합니다. 가용 억제 기간 이전, 도중 및 이후에 진행 중인 광고 시간의 경우 MediaTailor는 가용 억제 기간 이후에 발생하는 해당 광고 시간의 일부를 개인화하지만, 가용 억제 기간 이전 또는 도중에 발생하는 해당 광고 시간의 부분은 개인화하지 않습니다.

Avail suppression mode: AFTER_LIVE_EDGE

Avail suppression value: 00:30:00

Avail suppression fill policy: PARTIAL_AVAIL



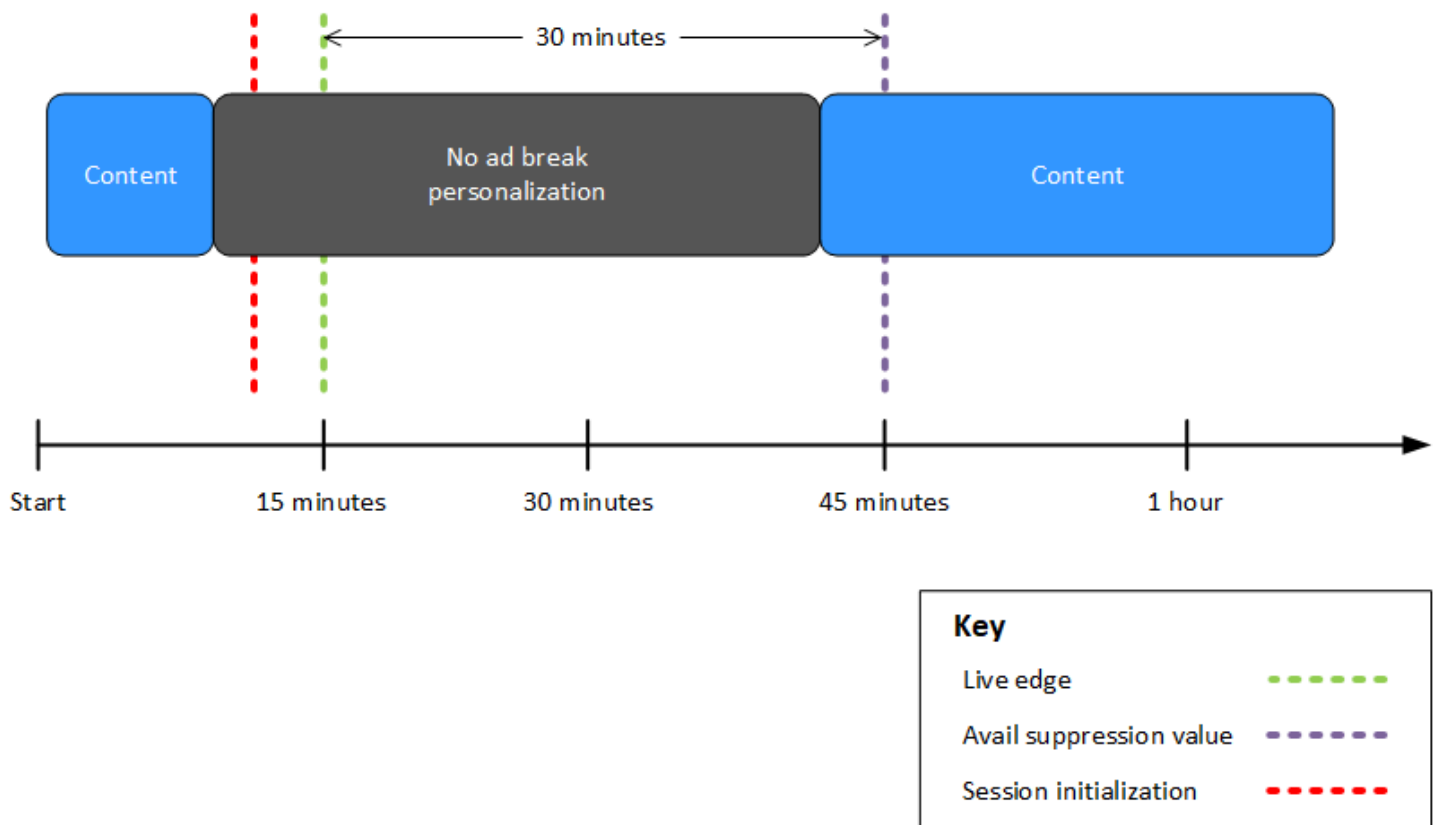
Example 7: 가용성 억제 기간이 시작될 때 AFTER_LIVE_EDGE 광고 시간이 진행 중인 광고 억제

가용 억제 모드가 로 설정AFTER_LIVE_EDGE되고 가용 억제 값이 0보다 크면 MediaTailor는 세션의 경과 시간이 해당 값에 도달할 때까지 광고 시간을 개인화하지 않습니다.

다음 그림에서는 왼쪽에서 오른쪽으로 진행되는 타임라인을 따라 다양한 블록이 가로로 배열됩니다. 각 블록은 라이브 스트림의 콘텐츠 또는 개인화되지 않은 광고 시간이 재생되는 시간을 나타냅니다. 점선은 라이브 스트림의 현재 라이브 엣지를 나타냅니다. 로 설정된 가용 억제 값을 나타내는 또 다른 점선은 라이브 엣지의 점선을 기준으로 타임라인에서 30분 후에 00:30:00 발생합니다. 세션 초기화를 나타내는 세 번째 점선은 라이브 엣지의 점선을 기준으로 타임라인 앞부분에서 발생합니다. 라이브 엣지 시간과 avail-suppression-value 시간 사이의 30분 기간은 가용 억제 기간을 나타냅니다. 광고 시간이 세션 초기화 전 시간부터 가용 억제 기간 내의 시간으로 진행 중입니다. 그림과 같이, 가용 억제 모드가 로 설정되면 AFTER_LIVE_EDGE 가용 억제 값은 라이브 엣지 00:30:00 뒤에 로 설정되고 세션 초기화는 라이브 엣지 시간 전에 발생하지만 광고 브레이크가 시작된 후에는 MediaTailor가 해당 광고 브레이크를 개인화하지 않습니다.

Avail suppression mode: AFTER_LIVE_EDGE

Avail suppression value: 00:30:00



광고 억제 파라미터 구성 - 재생 세션 요청

MediaTailor에 대한 초기 서버 측 또는 클라이언트 측 재생 세션 요청의 파라미터를 통해 광고 억제 설정을 구성할 수 있습니다. MediaTailor 콘솔 또는 AWS Elemental MediaTailor API를 통해 광고 억제 설정을 이미 구성한 경우 이러한 파라미터가 해당 설정을 재정의합니다.

광고 억제가 제대로 작동하려면 가능 구간 억제 모드와 가능 구간 억제 값이 모두 필요합니다. 이러한 파라미터는 다른 소스에서 구성할 수 없습니다. 예를 들어 MediaTailor 콘솔을 사용하여 하나의 파라미터를 구성하고 쿼리 파라미터를 사용하여 다른 파라미터를 구성할 수 없습니다.

MediaTailor는 다음과 같은 광고 억제 파라미터를 지원합니다.

이름	설명	허용되는 값
<code>availSuppressionMode</code>	광고 억제에 대한 모드를 설정합니다. 기본적으로 광고 억제는 <code>OFF</code> 로 설정하면 <code>BEHIND_LIVE_EDGE</code> MediaTailor는 <code>aws.availSuppressionValue</code> 해당 시간 또는 그 이후의 광고 시간을 채우지 않습니다. <code>AFTER_LIVE_EDGE</code> MediaTailor는 가용 금지 기간 또는 그 이후의 광고 시간을 채우지 않습니다. 가용 억제 기간은 라이브 엣지 시간부터 <code>aws.availSuppressionValue</code> 시간 및 추가 버퍼 시간까지입니다.	• <code>OFF</code> • <code>BEHIND_LIVE_EDGE</code> • <code>AFTER_LIVE_EDGE</code>
<code>availSuppressionValue</code>	라이브 스트림의 라이브 끝에 상대적인 시간입니다.	의 UTF-8 URL로 인코딩된 시간 코드입니다. <code>HH:MM:SS</code> . 예를 들어, 1시간 30분은 <code>01%3A30%3A00</code> 이 됩니다.
<code>availSuppressionFullPolicy</code>	가용성 억제 모드에 적용할 정책을 정의합니다. <code>BEHIND_LIVE_EDGE</code> 항상 전체 가용성 억제 정책을 사용합니다. 세션이 중간 브레이크를 시작할 때 부분 광고 브레이크 채우기를 호출하는 데를 사용할 <code>AFTER_LIVE_EDGE</code> 수 있습니다.	• <code>PARTIAL_AVAILABLE_BEFORE_LIVE_EDGE</code> 억제 모드에서 사용할 수 없음

이름	설명	허용되는 값
		FULL_AVAI L_ONLY - AFTER_LIV E_EDGE 억제 모 드의 기본값

서버 측 구성

기본 쿼리 파라미터는 `aws.availSuppression`이며, 그 뒤에는 선택적 파라미터 이름과 값 페어가 옵니다. 쿼리를 구성하려면 MediaTailor에 재생 세션 요청 `aws.availSuppression=` 끝을 추가한 다음 파라미터 이름과 값을 추가합니다. 서버 측 재생 세션 요청을 구성하는 방법에 대한 자세한 내용은 [MediaTailor 서버 측 광고 추적 및 보고](#) 단원을 참조하십시오.

예: HLS

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

서버 측 쿼리 구문은 다음 표에 나열되어 있습니다.

쿼리 문자열 구성 요소	설명
?	쿼리의 시작을 표시하는 제한된 문자입니다.
aws.	뒤에 이름과 값 페어로 구성된 파라미터가 오는 기본 쿼리입니다. 사용 가능한 모든 파라미터 목록은 광고 억제 파라미터 구성 - 재생 세션 요청 단원을 참조하십시오.
=	파라미터 이름을 값과 연결합니다. 예를 들어 <code>aws.availSuppressionMode= BEHIND_LIVE_EDGE</code> 입니다.
&	쿼리 파라미터를 연결합니다. 예를 들어, <code>aws.availSuppressionMode= BEHIND_LIVE_EDGE &aws.availSuppressionValue= 00:30:00&aws.availSuppressionFillPolicy= FULL_AVAAIL_ONLY</code> >입니다.

클라이언트 측 구성

MediaTailor에 대한 클라이언트의 POST 요청에 `availSuppression` 파라미터를 포함합니다. 클라이언트 측 재생 세션 요청을 구성하는 방법에 대한 자세한 내용은 [클라이언트 측 광고 추적](#) 단원을 참조하십시오.

예: HLS

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

MediaTailor 범퍼 광고 삽입

범퍼는 광고 시간 시작 시 또는 종료 전에 재생되는 짧고 건너뛴 수 없는 비디오 또는 오디오 클립입니다 AWS Elemental MediaTailor.

범퍼에는 다음 조건이 적용됩니다.

- 범퍼는 10초 이하여야 합니다.
- 범퍼는 광고 시간 시작 시, 광고 시간 종료 직전 또는 둘 다에 삽입할 수 있습니다.
- 프리롤이 구성되지 않은 한 재생 세션의 모든 광고 시간 동안 범퍼가 재생됩니다. 프리롤이 구성된 경우 프리롤 브레이크 중에는 범퍼가 재생되지 않습니다. 대신 프리롤 이후 모든 후속 브레이크에서 재생됩니다.
- HLS의 경우 각 SCTE-35 EXT-X-CUE-OUT 태그에 `duration` 속성을 포함해야 합니다.
- 범퍼는 소스 콘텐츠와 일치하도록 트랜스코딩됩니다.
- 범퍼에 대해서는 요금이 부과되지 않습니다.

범퍼 구성

범퍼를 사용하려면 MediaTailor 콘솔, MediaTailor API 또는 AWS Command Line Interface ()를 사용하여 범퍼 URLs을 구성합니다AWS CLI. 시작 범퍼, 끝 범퍼 또는 둘 다를 구성할 수 있습니다. 범퍼는

Amazon Simple Storage Service(Amazon S3)와 같은 서버에 저장됩니다. 범퍼 URLs은 저장된 범퍼 자산(들)의 위치를 나타냅니다.

시작 및 종료 범퍼 URLs에:

범퍼 URL 시작: `https://s3.amazonaws.com/startbumperad`

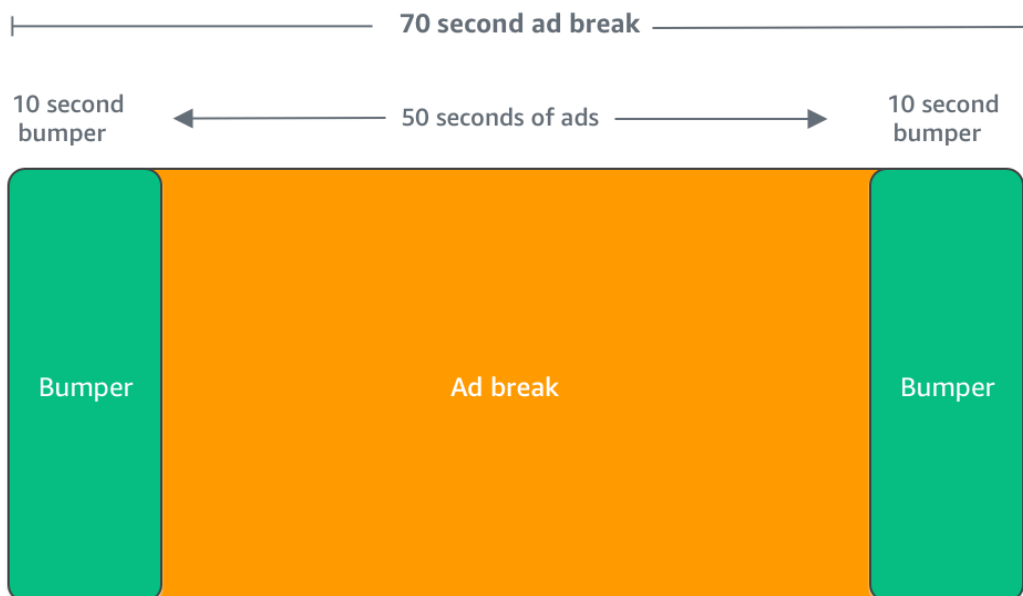
엔드 범퍼 URL: `https://s3.amazonaws.com/endbumperad`

예제

다음은 범퍼 광고 동작의 예입니다.

Example예제 1: 범퍼 시작 및 종료

이 예제에서는 시작 및 종료 범퍼가 활성화됩니다. 광고 결정 서버에는 70초의 광고 시간을 채우기 위해 50초의 맞춤형 광고가 있습니다. 10초 시작 범퍼는 광고 시간이 시작될 때 재생되고 50초의 광고가 재생된 다음 10초 종료 범퍼가 재생됩니다.



MediaTailor 프리롤 광고 삽입

Note

구성 가능한 프리롤 광고는 라이브 워크플로에서만 사용할 수 있습니다. VOD에 대한 광고 삽입(프리롤 포함)의 작동 방식에 대한 자세한 내용은 [섹션을 참조하세요](#) [VOD에 대한 광고 스티칭 동작](#).

AWS Elemental MediaTailor 는 기본 콘텐츠가 시작되기 전에 재생 세션 시작 시 광고를 삽입할 수 있습니다. 이러한 광고가 프리롤 광고입니다.

프리롤 광고를 삽입하려면 설명된 대로 구성의 추가 설정에서 라이브 프리롤 광고 결정 서버 및 라이브 프리롤 최대 허용 기간 필드를 작성합니다 [선택적 구성 설정](#).

- MediaTailor는 재생 요청을 수신하면 MediaTailor 재생 구성의 다음 필드를 기반으로 프리롤 광고에 대한 요청을 ADS에 보냅니다.
 - 라이브 프리롤 광고 결정 서버는 MediaTailor가 프리롤 광고 요청을 보내는 광고 결정 서버(ADS) URL입니다.
 - 라이브 프리롤 최대 허용 기간은 프리롤 광고의 총 최대 시간입니다. MediaTailor는 최대 허용 기간에 따라 다음 작업을 수행합니다.
 - ADS 응답의 총 광고 기간이 라이브 프리롤 최대 허용 기간에 제공한 값보다 작으면 MediaTailor는 모든 광고를 삽입합니다. 마지막 광고가 완료되면 MediaTailor는 즉시 기본 콘텐츠로 돌아갑니다.
 - ADS 응답의 총 광고 지속 시간이 라이브 프리롤 최대 허용 지속 시간에서 제공한 값보다 크면 MediaTailor는 계속 진행하지 않고 지속 시간에 맞는 광고 세트를 선택합니다. MediaTailor 는 클리핑이나 잘림 없이 이러한 광고를 삽입합니다. 마지막으로 선택한 광고가 완료되면 MediaTailor가 기본 콘텐츠로 돌아갑니다.
- MediaTailor는 ADS로부터 프리롤 응답을 수신하면 매니페스트를 조작하여 프리롤 광고에 대한 링크를 추가합니다. MediaTailor는 다음과 같이 롤링 전 광고 시간의 시작 시간을 계산합니다.
 - DASH의 경우 공식은 $(publishTime - availabilityStartTime) - \max(suggestedPresentationDelay, minBufferTime)$ 입니다.
 - HLS의 경우 공식은 $\max(2 * EXT-X-TARGETDURATION, EXT-X-START:TIMEOFFSET)$ 입니다.
- MediaTailor는 프리롤이 아닌 광고 시간에 수행할 작업을 결정합니다. 프리롤이 다른 광고 시간과 겹치는 경우 MediaTailor는 광고 시간의 겹치는 부분을 개인화하지 않습니다.

MediaTailor 슬레이트 광고 삽입

Note

Slate는 라이브 및 live-to-VOD 워크플로에만 사용할 수 있습니다.

를 사용하면 광고 시간에 대한 슬레이트 광고를 지정할 AWS Elemental MediaTailor 수 있습니다. 슬레이트는 라이브 콘텐츠 대신 재생되는 스틸 이미지 또는 루프 비디오와 같은 스트림에 삽입되는 기본 MP4 자산입니다.

AWS Elemental MediaTailor 는 다음과 같은 특정 상황에서 광고 시간 동안 슬레이트를 표시합니다.

- 개인 맞춤형 광고로 대체되지 않은 휴식 시간에 남은 시간을 채우려면
- Ad Decision Server(ADS)가 빈 VAST 또는 VMAP 응답으로 응답하는 경우
- ADS 제한 시간 또는 ADS의 HTTP 500 오류와 같은 오류 조건의 경우
- MediaTailor에서 광고를 삽입할 수 없는 경우(예: 트랜스코딩을 완료하지 않은 경우)

슬레이트를 구성하지 않으면 위의 조건 중 하나가 충족되면 MediaTailor가 기본 콘텐츠 스트림으로 기본 설정됩니다.

슬레이트 구성

[MediaTailor 콘솔](#)의 추가 구성 창에서 슬레이트를 지정합니다. MediaTailor는 지정한 URL에서 슬레이트를 다운로드하고 콘텐츠와 동일한 변환으로 트랜스코딩합니다. MediaTailor 콘솔의 선택적 개인화 임계값 구성을 통해 슬레이트가 표시되는 최대 시간을 제어할 수 있습니다. 자세한 내용은 [the section called “개인화 임계값 작동 방식”](#) 단원을 참조하십시오.

슬레이트는 오디오 및 비디오가 모두 포함된 품질이 높은 MP4 자산이어야 합니다. 슬레이트 구성은 비 VPAID 구성의 경우 선택 사항이지만 VPAID 워크플로의 경우 필수입니다.

Note

슬레이트를 호스팅하는 서버가 HTTPS를 사용하는 경우 인증은 잘 알려진 인증 기관에서 받아야 합니다. 자체 서명된 인증서는 될 수 없습니다. 자체 서명된 인증서를 사용하는 경우는 콘텐츠 오리진에서 슬레이트를 검색하여 매니페스트에 연결할 AWS Elemental MediaTailor 수 없습니다.

개인화 임계값 작동 방식

개인 맞춤 임계값은 광고 시간에 허용되는 과소 채워진 광고 시간의 최대 기간(초)을 정의합니다. 이 기능은 기본 콘텐츠 스트림에 의존하기 때문에 광고 삽입이 아닌 라이브 및 VOD 스트림의 광고 교체에만 적용됩니다.

동작은 다음 세 가지 시나리오에 따라 달라집니다.

1. 개인 맞춤이 비활성화된 경우:
 - a. 채워지지 않은 전체 시간 동안 슬레이트가 삽입됩니다.
 - b. 구성 시 시작/종료 범퍼가 삽입됩니다(자세한 내용은 참조 [MediaTailor 범퍼 광고 삽입](#)).
 - c. 광고는 정상적으로 삽입됩니다.
2. 개인화가 활성화되고 임계값이 휴식 시간 미만인 경우:
 - a. 채워지지 않은 시간이 개인화 임계값을 초과하는 경우:
 - i. MediaTailor가 광고 시간의 개인화를 중단합니다.
 - ii. 기본 콘텐츠가 표시됩니다.
 - iii. 광고, 슬레이트 또는 범퍼가 삽입되지 않음
 - b. 채워지지 않은 시간이 개인화 임계값보다 작은 경우:
 - i. 광고 및 슬레이트 삽입
 - ii. 구성된 경우 범퍼가 삽입됩니다.
3. 개인화가 활성화되고 임계값이 휴식 시간보다 큰 경우:
 - a. 광고가 삽입됩니다.
 - b. 광고 시간의 남은 시간 동안 슬레이트가 삽입됩니다.
 - c. 구성된 경우 범퍼가 삽입됩니다.

다음 표에는 HLS 및 DASH 프로토콜 모두에서 특정 시나리오에 대한 자세한 동작이 나와 있습니다.

세부 동작 매트릭스

시나리오	개인화 비활성화됨	휴식 시간보다 짧은 개인화 활성화	휴식 시간보다 오래 활성화된 개인화
빈 VAST 또는 VMAP	삽입된 슬레이트	삽입된 슬레이트 없음	삽입된 슬레이트

시나리오	개인화 비활성화됨	휴식 시간보다 짧은 개인화 활성화	휴식 시간보다 오래 활성화된 개인화
ADS 제한 시간	삽입된 슬레이트	삽입된 슬레이트 없음	삽입된 슬레이트
광고를 사용할 수 없음(Vast 404)	삽입된 슬레이트	삽입된 슬레이트 없음	삽입된 슬레이트
광고 기간이 광고 시간보다 길니다.	삽입된 광고	삽입된 광고 없음	삽입된 광고
광고 교체에서 완전히 사용되지 않는 시간 채우기	삽입된 슬레이트	개인 맞춤 임계값이 채워지지 않은 시간보다 큰 경우: 광고/슬레이트가 삽입되지 않음, 그렇지 않은 경우: 광고 및 슬레이트가 삽입됨	삽입된 슬레이트
빈 VAST로 사전 롤링	슬레이트 삽입됨, 프리롤 삽입되지 않음	삽입된 슬레이트 없음, 삽입된 프리롤 없음	슬레이트 삽입됨, 프리롤 삽입되지 않음
ADS 제한 시간으로 사전 롤링	슬레이트 삽입됨, 프리롤 삽입되지 않음	삽입된 슬레이트 없음, 삽입된 프리롤 없음	슬레이트 삽입됨, 프리롤 삽입되지 않음
광고를 사용할 수 없는 경우 사전 등록 (Vast 404)	슬레이트 삽입됨, 프리롤 삽입되지 않음	삽입된 슬레이트 없음, 삽입된 프리롤 없음	슬레이트 삽입됨, 프리롤 삽입되지 않음
빈 VAST가 있는 범퍼	슬레이트 삽입, 범퍼 삽입	삽입된 슬레이트 없음, 삽입된 범퍼 없음	슬레이트 삽입, 범퍼 삽입

시나리오	개인화 비활성화됨	휴식 시간보다 짧은 개인화 활성화	휴식 시간보다 오래 활성화된 개인화
ADS 제한 시간이 있는 범퍼	슬레이트 삽입, 범퍼 삽입	삽입된 슬레이트 없음, 삽입된 범퍼 없음	슬레이트 삽입, 범퍼 삽입
광고를 사용할 수 없는 경우 범퍼 (Vast 404)	슬레이트 삽입, 범퍼 삽입	삽입된 슬레이트 없음, 삽입된 범퍼 없음	슬레이트 삽입, 범퍼 삽입
광고 대체에 완전히 사용되지 않는 시간을 채우는 범퍼	슬레이트 삽입, 범퍼 삽입, 광고 삽입	개인 맞춤 임계값이 채워지지 않은 시간보다 큰 경우: bumpers/ads/slates 삽입되지 않음, 그렇지 않은 경우: bumpers/ads/slates 삽입되지 않음	슬레이트 삽입, 범퍼 삽입, 광고 삽입

중요 고려 사항

슬레이트 및 개인화 임계값을 사용할 때는 다음 사항에 유의하세요.

- 이 동작은 HLS 프로토콜과 DASH 프로토콜 모두에서 일관됩니다.
- 개인화 임계값 기능은 슬레이트가 구성된 경우에만 적용됩니다.
- 슬레이트가 구성되고 개인 맞춤 임계값이 구성되지 않은 경우 슬레이트는 전체 오리진 가용 기간 동안 또는 광고를 채운 후 남은 시간 동안 재생됩니다.
- VPAID 광고의 경우 MediaTailor는 VPAID 광고 기간 동안 슬레이트를 삽입하여 비디오 플레이어에 삽입할 광고 공간을 확보합니다.

슬레이트 구성 및 VPAID

Important

VPAID 광고를 사용할 때는 슬레이트 구성이 필수입니다. MediaTailor는 비디오 플레이어에 삽입할 VPAID 광고의 공간을 확보하기 위해 슬레이트를 삽입합니다. 슬레이트 지속 시간은 사용자 상호 작용을 수용하기 위해 VPAID 광고 지속 시간보다 약간 더 길 수 있습니다.

그런 다음 비디오 플레이어는 설명된 대로 MediaTailor가 반환하는 클라이언트 측 보고 메타데이터를 기반으로 VPAID 광고를 처리합니다. [VPAID 요구 사항](#). 클라이언트 측 보고에 대한 자세한 내용은 [the section called “클라이언트 측 추적”](#) 단원을 참조하십시오.

의 광고 트랜스코딩 AWS Elemental MediaTailor

AWS Elemental MediaTailor 는 오리진 비디오 스트림의 형식 및 사양과 일치하도록 광고, 범퍼 및 슬레이트를 트랜스코딩하여 광고 중단 시 원활한 재생을 보장합니다. MediaTailor는 동적 트랜스코딩(기본값)과 사용자 지정 트랜스코딩 프로파일이라는 두 가지 트랜스코딩 접근 방식을 지원합니다. 이러한 접근 방식의 차이점을 이해하면 워크플로에 적합한 구성을 결정하는 데 도움이 됩니다.

주제

- [동적 트랜스코딩](#)
- [사용자 지정 트랜스코딩 프로파일](#)

동적 트랜스코딩

동적 트랜스코딩은 MediaTailor의 기본 광고 트랜스코딩 동작입니다. 광고, 범퍼 또는 슬레이트를 트랜스코딩해야 하는 경우 MediaTailor는 자동으로 오리진 매니페스트를 검사하고 사용자 측의 구성 없이 런타임에 트랜스코딩 설정을 도출합니다.

동적 트랜스코딩이 처리하는 내용

동적 트랜스코딩은 HLS 및 DASH 워크플로에 대해 다음을 자동으로 처리합니다.

- 표준 코덱 일치 - MediaTailor는 오리진 스트림의 코덱 프로파일과 일치합니다. 코덱 수준 일치는 보장되지 않습니다.
- 오디오 정규화 - MediaTailor는 기본 대상 수준을 사용하여 트랜스코딩된 광고에 오디오 정규화를 자동으로 적용합니다-24 LKFS. 이렇게 하면 추가 구성 없이 광고 오디오 수준이 오리진 콘텐츠와 일

관성을 유지할 수 있습니다. 워크플로에 기본값인 이외의 사용자 지정 오디오 정규화 대상이 필요한 경우 -24 LKFS 사용자 지정 트랜스코드 프로파일이 필요합니다.

- 표준 세그먼트 길이 일치 - 기본적으로 MediaTailor는 2초 세그먼트를 사용하여 광고를 트랜스코딩합니다. 플레이어에 세그먼트 길이 일치에 필요한 경우 `calculateSegmentLength` 플러그를 활성화하면 MediaTailor가 오리진에서 처음 10개의 세그먼트를 분석하고 가장 일반적인 기간을 세그먼트 길이로 선택하여 올바른 세그먼트 길이를 자동으로 계산합니다. 이 플러그는 기본적으로 비활성화되어 있으며 AWS Support에서 활성화해야 합니다. 계정 또는 특정 구성에 `calculateSegmentLength` 대한 활성화를 요청하려면 AWS Support에 문의하세요.
- 표준 WebVTT 자막 - HLS 워크플로의 경우 MediaTailor는 오리진 매니페스트에 자막이 있을 때 개인화된 다변량 재생 목록에 WebVTT 자막 그룹 참조를 보존합니다. 광고 크리에이티브에는 일반적으로 자막 트랙이 포함되지 않으므로 광고 세그먼트 중에 MediaTailor는 자막 트랙 연속성을 유지하기 위해 빈 WebVTT 세그먼트를 삽입합니다. DASH 워크플로의 경우 MediaTailor에서는 오리진 매니페스트의 자막 적응 세트를 분할해야 합니다(예: 비디오 및 오디오). MediaTailor에서 구성한 광고 기간에는 광고 크리에이티브에서 제공하는 적응 세트만 포함됩니다. 광고 크리에이티브에는 일반적으로 자막 트랙이 포함되지 않으므로 광고 기간에는 자막이 포함되지 않습니다 `AdaptationSet`. 콘텐츠 기간은 자막 적응 세트를 정상적으로 유지합니다.
- CMAF 형식 지원 - MediaTailor는 HLS 오리진 스트림이 CMAF (fMP4) 세그먼트 또는 표준 TS 세그먼트를 사용하는지 여부를 감지하고 그에 따라 일치하도록 광고를 트랜스코딩합니다. 이 경우 기본적으로 ISO-BMFF (fMP4)를 컨테이너 형식으로 DASH 사용하므로 DASH CMAF 패키징이 투명하게 지원됩니다. 어느 프로토콜에도 추가 구성이 필요하지 않습니다.

Note

MediaTailor는 TS 세그먼트가 있는 코덱으로 HVC1 (HEVC)를 사용하는 HLS 스트림을 지원하지 않습니다. Apple 디바이스의 HLS 작성 사양, 섹션 1.5에 따라 HEVC 비디오의 컨테이너 형식은 여야 합니다 fMP4. 따라서 사용하는 HLS 스트림은 조각화된 MP4 세그먼트 CMAF와 함께 사용해야 HEVC 합니다.

- 트릭플레이 개인화 - MediaTailor는 오리진 콘텐츠 사양과 DASH 일치하는 HLS 및 모두에 대해 트릭플레이 변형 및 관련 이미지 스트림(I-프레임 매니페스트, 썸네일 및 이미지 스트림 포함)으로 광고를 자동으로 트랜스코딩합니다. 이 기능에는 사용자 지정 트랜스코딩 프로파일이 더 이상 필요하지 않습니다.
- 소형 DASH 매니페스트 - MediaTailor는 자동으로 소형 DASH 매니페스트를 지원하여 `SegmentTemplate` 요소를 개별 `Representation` 요소에서 `AdaptationSet` 수준으로 높여 전체 매니페스트 크기를 줄이고 전송 효율성을 개선합니다. 이 기능에는 사용자 지정 트랜스코딩 프로파일이 더 이상 필요하지 않습니다.

동적 트랜스코딩으로 충분하지 않을 수 있는 경우

동적 트랜스코딩은 대부분의 광고 트랜스코딩 시나리오를 처리합니다. 그러나 추가 구성 또는 사용자 지정 트랜스코딩 프로파일이 필요할 수 있는 경우가 있습니다.

- 비트레이트 일치 - 동적 트랜스코딩은 기본적으로 정확한 비트레이트 일치를 보장하지 않습니다. 워크플로에서 오리진 비트레이트와 거의 일치하도록 광고 비트레이트가 필요한 경우 `enableBitrateRounding` 플래그를 활성화하여 최대 5%의 비트레이트 반올림이 오리진 값과 더 잘 일치하도록 할 수 있습니다. 이 플래그는 AWS Support에서 활성화해야 합니다. 계정 또는 특정 구성에 `enableBitrateRounding` 대한 활성화를 요청하려면 AWS Support에 문의하세요. 워크플로에 5%를 초과하는 비트레이트 일치가 필요한 경우 사용자 지정 트랜스코딩 프로파일이 필요합니다.
- MediaTailor 기본값이 아닌 사용자 지정 오디오 정규화 대상 -24 LKFS
- 정수가 아닌 세그먼트 길이(예: 1.92초 또는 3.84초)
- Dolby Vision 오디오
- 다운스트림 시스템 또는 플레이어에 필요한 특정 PMT, 비디오 또는 오디오 PID 값
- 사용자 지정 오디오 샘플 속도(예: 44.1kHz)
- 사용자 지정 픽셀 종횡비(PAR) - 예: 사각형 픽셀의 경우 1:1 또는 D1/DV NTSC 와이드스크린의 경우 40:33
- 명명된 특정 오디오 트랙 또는 자막 트랙

사용자 지정 트랜스코딩 프로파일

사용자 지정 트랜스코딩 프로파일(CTP)은 MediaTailor에 등록하는 명시적으로 작성된 AWS Elemental MediaConvert 작업 구성입니다. CTP가 MediaTailor 구성과 연결되면 MediaTailor는 정의된 설정을 사용하여 모든 광고, 범퍼 및 슬레이트를 트랜스코딩하므로 코덱, 비트레이트, 오디오 정규화 수준, 세그먼트 길이, PID 값, 샘플 속도 등을 정확하게 제어할 수 있습니다. CTP는 동적 트랜스코딩을 완전히 재정의하므로 프로필 내에서 트릭플레이 기능(I-프레임 매니페스트, 썸네일 및 이미지 스트림)을 포함하여 필요한 모든 기능을 명시적으로 구성해야 합니다.

CTP는 특정 오리진 구성과 일치하도록 설계되었습니다. 소스 콘텐츠가 CTP가 원래 생성된 구성과 다른 경우 `NO_VARIANT_MATCH` 오류로 인해 광고가 제대로 트랜스코딩되지 않을 수 있습니다. 이는 CTP 출력이 오리진 매니페스트에 있는 변형과 일치하지 않을 때 발생합니다. 이는 라이브 및 VOD 워크플로 모두에 적용됩니다. 예를 들어, 다른 소스의 콘텐츠를 VOD 포함하거나 다른 인코딩 프로파일(예: 다양한 해상도, 비트레이트, 코덱 또는 세그먼트 길이)로 인코딩된 라이브러리는 CTP가 라이브러리 전체에 존재하는 모든 변형 조합을 고려하지 않는 경우 `NO_VARIANT_MATCH` 오류를 생성할 수

있습니다. 여러 오리진을 지원하는 단일 CTP 또는 혼합 인코딩 프로파일이 있는 VOD 라이브러리가 필요한 경우 모든 대상 오리진의 변형을 포함하는 수퍼셋 CTP를 생성할 수 있습니다. 그러나 수퍼셋 CTPs에는 비디오 변형 36개의 하드 제한이 적용되므로 단일 프로파일로 수용할 수 있는 오리진 또는 변형 조합의 수가 제한됩니다.

Important

인코더 설정을 변경하거나 새 VOD 콘텐츠 소스를 사용할 때마다 CTP를 동기화된 상태로 유지해야 합니다. 이렇게 하지 않으면 NO_VARIANT_MATCH 오류와 광고 트랜스코딩 실패가 발생합니다. 여기에는 해상도 추가 또는 제거, 비트 전송률 조정, 코덱 변경, 세그먼트 길이 수정 또는 인코딩 프로필이 다른 새 공급자의 VOD 콘텐츠 온보딩과 같은 변경 사항이 포함됩니다.

CTP를 업데이트할 때 다음 동작에 유의하세요.

- 기존 CTP를 업데이트하면 새로 트랜스코딩된 광고에만 새 설정이 적용됩니다. 이전 CTP 설정에서 이전에 트랜스코딩된 광고는 다시 트랜스코딩되지 않습니다.
- 새 CTP를 생성하고 MediaTailor 구성과 연결하면 MediaTailor에서 새 프로필을 사용하여 모든 광고를 다시 트랜스코딩하라는 메시지가 표시되므로 광고 인벤토리 전반에서 완전한 일관성을 보장할 수 있습니다.

따라서 인코더 설정이 변경되거나 새 콘텐츠 소스를 온보딩하는 경우, 이전에 트랜스코딩된 광고를 포함한 모든 광고가 업데이트된 설정을 반영하도록 기존 CTP를 업데이트하는 대신 새 CTP를 생성하는 것이 가장 좋습니다.

사용자 지정 트랜스코딩 프로파일을 사용해야 하는 경우

워크플로에 다음 중 하나 이상이 필요한 경우 사용자 지정 트랜스코딩 프로파일을 사용합니다.

- MediaTailor 기본값 이외의 사용자 지정 오디오 정규화 대상 -24 LKFS
- 정수가 아닌 세그먼트 길이(예: 1.92 또는 3.84)
- Dolby Vision 오디오 구성
- 다운스트림 시스템 또는 플레이어에 필요한 특정 PMT, 비디오 또는 오디오 PID 값
- 사용자 지정 오디오 샘플 속도(예: 44.1kHz)
- 구성 플래그를 사용한 동적 트랜스코딩에서 지원하는 5% 임계값을 초과하는 비트레이트 일치
- 사용자 지정 픽셀 종횡비(PAR) - 예: 사각형 픽셀의 경우 1:1 또는 D1/DV NTSC 와이드스크린의 경우 40:33

- 명명된 특정 오디오 또는 자막 트랙

사용자 지정 트랜스코딩 프로파일 시작하기

CTP를 등록하는 데 필요한 MediaTailor 트랜스코딩 API에 대한 액세스는 허용 목록에 있는 계정으로 제한됩니다. 시작하려면 다음 옵션 중 하나를 사용하여 AWS Support에 문의하세요.

- 허용 목록 액세스 요청 - 자체 CTPs 생성하고 관리하려면 AWS Support에 문의하여 계정을 허용 목록에 추가하도록 요청합니다. 계정이 허용 목록에 추가되면 트랜스코딩 API에 액세스하여 기존 MediaTailor 구성이 있는 모든 AWS 리전에서 프로필을 등록하고 관리할 수 있습니다.
- AWS Support에서 CTP 생성 요청 - AWS Support가 사용자를 대신하여 CTP를 생성하고 등록하도록 하려면 MediaTailor가 액세스할 수 있는 non-DRM-protected 오리진 엔드포인트 URL을 사용하여 AWS Support에 문의하세요. AWS Support는 이 엔드포인트를 사용하여 오리진 구성을 분석하고 계정에 적합한 CTP를 생성합니다.

동적 트랜스코딩과 사용자 지정 트랜스코딩 프로파일 비교

다음 표를 사용하여 워크플로에 적합한 접근 방식을 결정합니다.

동적 트랜스코딩과 사용자 지정 트랜스코딩 프로파일 비교

고려 사항	동적 트랜스코딩	사용자 지정 트랜스코딩 프로파일
설정 작업	없음	수동으로 작성된 AWS Elemental MediaConvert 작업 구성
Trickplay 지원(HLS 및 DASH)	자동	CTP에서 명시적으로 구성해야 합니다.
컴팩트 매DASH니페스트	자동	CTP에서 명시적으로 구성해야 합니다.
CMAF 형식 지원	자동	CTP에서 명시적으로 구성해야 합니다.
오리진 변경 처리	자동	수동 CTP 업데이트 필요

고려 사항	동적 트랜스코딩	사용자 지정 트랜스코딩 프로파일
다중 리전 복잡성	없음	리전별로 등록해야 합니다.
오디오 정규화	-24 LKFS(기본값)	완전히 구성 가능
비트레이트 일치	enableBitrateRounding 플래그를 통한 최대 5% 반올림(AWS Support에 문의하여 활성화)	오리진 비트 전송률과 일치하도록 CTP에 명시적으로 구성됨
정수가 아닌 세그먼트 길이	기본값 2초, 정수가 아닌 세그먼트 길이는 지원되지 않음(자동 정수 세그먼트 길이 계산을 활성화하려면 AWS SupportcalculateSegmentLength에 문의)	오리진과 일치하도록 세그먼트 길이를 수동으로 구성해야 합니다.
사용자 지정 PID 값	지원되지 않음	지원됨
Dolby Vision 오디오	지원되지 않음	지원됨
사용자 지정 오디오 샘플 속도	지원되지 않음	지원됨
다중 오리진 지원	자동	슈퍼셋 CTP 필요, 비디오 변형 36개로 제한
MediaTailor 개선 사항	자동	수동 CTP 업데이트 필요
다음과 같은 경우에 가장 적합	표준 코덱 매칭, 공통 비트 전송률 프로파일, 트릭플레이, 압축 DASH	동적 트랜스코딩으로 충족되지 않는 특수 인코딩 요구 사항

광고 미리 가져오기

라이브 스트림에 광고 AWS Elemental MediaTailor 미리 가져오기를 사용하면 광고 결정 서버(ADS)의 피크 로드를 줄이고 각 광고 시간 시작 시 매니페스트 전송 지연 시간을 줄일 수 있습니다. 미리 가져오

기 일정을 정의하면 MediaTailor는 일정에 따라 ADS에서 광고를 검색하고 광고 시간에 필요하기 전에 광고 삽입을 준비합니다. 라이브 스트림 중에 미리 가져오기를 수행하면 광고 요청 및 트랜스코딩 제한 시간 또는 기타 네트워크 지연으로 인한 광고 채우기 비율 감소 및 수익 창출 기회 누락을 완화하는 데 도움이 될 수 있습니다.

Note

광고 미리 가져오기는 기존 서버 가이드 및 HLS Interstitials를 포함한 서버 가이드 광고 삽입 (SGAI) 메서드에서는 작동하지 않습니다. 플레이어는 재생할 광고만 가져오므로 SGAI 메서드는 미리 가져올 필요가 없으며 MediaTailor가 개별 세션 요청을 보지 않고도 CDN에서 매니페스트를 제공할 수 있습니다.

광고 미리 가져오기를 설정하려면 재생 구성에서 하나 이상의 미리 가져오기 일정을 생성합니다. 미리 가져오기 일정은 MediaTailor에 예정된 광고 시간에 대한 광고를 검색하고 준비하는 방법과 시기를 알려줍니다.

- 이벤트에 예측 가능한 일정에 따라 광고 가능 시간이 있는 경우 단일 미리 가져오기 일정을 사용합니다. 각 사전 가져오기 일정은 MediaTailor가 단일 광고 시간에 배치할 단일 광고 세트를 정의합니다. 단일 사전 가져오기 일정을 사용할 때 여러 광고 시간에 대한 광고를 미리 가져오려면 각 광고 시간과 관련된 여러 사전 가져오기 일정(광고 시간 최대 24시간 전)을 생성해야 합니다.
- 이벤트에 예측 가능한 일정에 없는 광고 가능성이 있는 경우 반복 미리 가져오기 일정을 사용합니다. 반복 미리 가져오기 일정은 자동으로 일정을 생성하고 이벤트의 각 광고 중단 전에 광고를 미리 가져옵니다. 반복 미리 가져오기 일정은 정의된 기간(이벤트 종료 전 최대 24시간) 내에 모든 광고 가능 시간에 대한 광고를 검색합니다. 각 광고 시간에 대한 일정을 생성할 필요는 없지만 단일 미리 가져오기가 제공하는 타이밍 제어의 일부를 잃게 됩니다.

다음 주제에서는 광고 미리 가져오기에 대해 자세히 설명합니다.

주제

- [프리페치 작동 방식](#)
- [미리 가져오기 일정 생성](#)
- [TPS 기반 트래픽 셰이핑](#)
- [미리 가져오기 일정 삭제](#)

프리페치 작동 방식

클라이언트가 MediaTailor에 매니페스트 요청을 하면 서비스는 재생 구성과 연결된 모든 미리 가져오기 일정을 평가합니다. MediaTailor가 일치하는 미리 가져오기 일정을 찾지 못하면 서비스는 일반 광고 삽입으로 되돌리고 광고를 미리 가져오지 않습니다.

MediaTailor가 일치하는 미리 가져오기 일정을 찾으면 서비스는 검색 및 소비라는 두 가지 구성 요소를 기반으로 일정을 평가합니다. 각 구성 요소의 구성은 다음 섹션에 설명된 대로 단일 미리 가져오기 일정과 반복되는 미리 가져오기 일정에 따라 다릅니다.

단일 미리 가져오기 일정 흐름

검색

MediaTailor가 ADS에서 광고를 미리 가져올 때의 시간 범위인 검색 창을 정의합니다. 이 기간을 광고 시간 이전으로 예약해야 합니다. 다음은 MediaTailor가 단일 미리 가져오기 일정을 처리하는 방법에 대한 개요를 제공합니다.

콘솔에서 단일 미리 가져오기 일정을 생성하는 단계는 섹션을 참조하세요 [미리 가져오기 일정 생성](#). API 지침은 API 참조의 [PrefetchSchedules](#)를 AWS Elemental MediaTailor 참조하세요.

지정된 검색 기간 동안 MediaTailor는 ADS에 요청을 보내 나중에 재생 세션에 삽입할 광고를 검색하고 준비합니다.

- MediaTailor는 선택적으로 트래픽 셰이핑을 사용하여 한 번에 ADS에 대한 요청 수를 제한합니다. 두 가지 접근 방식 중에서 선택할 수 있습니다.

기간 트래픽 셰이핑 - MediaTailor는 모든 세션에 대한 요청을 한 번에 보내는 대신 지정된 초수에 걸쳐 요청을 분산합니다. 이렇게 분산된 트래픽 분산은 ADS가 압도되어 제한 시간이 초과되고 광고 채우기 속도가 낮아지는 것을 방지하는 데 도움이 됩니다.

TPS 기반 트래픽 셰이핑 - MediaTailor는 초당 트랜잭션(TPS) 및 동시 사용자를 기준으로 요청을 제한합니다. 이 접근 방식은 ADS 용량 제한에 따라 보다 직관적인 구성을 제공합니다. 자세한 내용은 [TPS 기반 트래픽 셰이핑](#) 단원을 참조하십시오.

- 동적 변수를 설정하면 MediaTailor는 ADS에 대한 요청에 이러한 변수를 포함합니다. MediaTailor는 소비 기간 동안 이러한 변수를 사용하여 광고 시간을 미리 가져오기 일정에 일치시킵니다. 자세한 내용은 다음 사용 섹션을 참조하세요.

Example

라이브 이벤트는 오전 7시 45분부터 10AM시까지 지속되며, 광고 시간은 오전 8시 15분입니다. 트래픽 셰이핑 기간이 60초인 오전 7시 45분부터 8AM시까지 광고를 검색하도록 MediaTailor를 구성합니다. 동시 사용자가 500,000명인 경우 MediaTailor는 ADS 요청을 분산하여 모든 요청을 동시에 보내는 대신 60초 동안 초당 평균 약 8,333개의 트랜잭션($500,000\text{명}/60\text{초}=\text{초당 } 8,333\text{개 요청}$)을 달성합니다.

검색 구성에는 동적 변수 키 `scte.event`와 값이 포함됩니다1234. MediaTailor는 ADS에 대한 요청에이 변수를 포함하며,이 변수를 사용하여 특정 광고자를 이벤트 ID 1234로 대상으로 지정할 수 있습니다.

소비

MediaTailor는 소비 기간 동안 SCTE-35 광고 중단 마커를 발견하면 미리 가져온 광고를 광고 중단 시간에 배치합니다.

- 가용 일치 기준을 설정하지 않은 경우 MediaTailor는 소비 기간의 첫 번째 휴식 시간에 광고를 삽입합니다.
- 가용 일치 기준에 대해 동적 변수 키를 설정한 경우 MediaTailor는 검색 창에서 설정한 동적 변수에 대해 이러한 기준을 평가합니다. 광고 시간은 가용성 일치 기준이 충족되는 경우에만 미리 가져온 광고 삽입에 사용할 수 있습니다. MediaTailor는 기준을 충족하는 첫 번째 휴식 시간에 광고를 삽입합니다.

지원되는 가용 일치 기준 목록은의 표에서 광고 미리 가져오기에 사용 가능 열을 참조하세요[ADS 요청에 대한 MediaTailor 세션 변수](#).

Example계속

사용 시작 시간을 오전 8시 15분으로 설정하고 종료 시간을 오전 8시 17분으로 설정합니다. 가용 일치 기준에 `scte.event_id` 키에를 포함합니다.

MediaTailor가 오전 8시 15분부터 오전 8시 17분까지 접하는 각 광고 시간에 대해 각 광고 시간에 대한 SCTE 이벤트 ID를 평가합니다. 각 재생 세션에서 MediaTailor는 이벤트 ID가 1234(검색 동적 변수에 정의됨)인 첫 번째 광고 시간에 미리 가져온 광고를 삽입합니다. 올바른 이벤트 ID가 포함되지 않은 광고 시간의 경우 MediaTailor는 표준 광고 삽입을 수행합니다.

반복 미리 가져오기 일정 흐름

검색

MediaTailor가 라이브 이벤트에 대한 광고를 미리 가져오고 삽입하는 시간 범위인 반복 검색 기간을 정의합니다(최대 24시간). 다음은 MediaTailor가 반복 미리 가져오기 일정을 처리하는 방법에 대한 개요를 제공합니다.

콘솔에서 반복 프리페치 일정을 생성하는 단계는 섹션을 참조하세요 [미리 가져오기 일정 생성](#). API 지침은 API 참조의 [PrefetchSchedules](#)를 AWS Elemental MediaTailor 참조하세요.

지정된 반복 미리 가져오기 기간 동안 MediaTailor는 최대 24시간의 라이브 이벤트에 대한 광고를 검색하고 삽입합니다. 창의 각 광고 시간이 끝나면 MediaTailor는 다음 광고 시간에 대한 광고를 자동으로 검색합니다.

- 가용 시간이 종료된 후 지연을 설정하면 MediaTailor는 지정된 시간을 기다린 후 다음 광고 시간에 대한 다음 광고 세트를 검색합니다.
- MediaTailor는 선택적으로 트래픽 셰이핑을 사용하여 한 번에 ADS에 대한 요청 수를 제한합니다. 두 가지 접근 방식 중에서 선택할 수 있습니다.

기간 트래픽 셰이핑 - MediaTailor는 모든 세션에 대한 요청을 한 번에 보내는 대신 지정된 초 수에 걸쳐 요청을 분산합니다. 이렇게 분산된 트래픽 분산은 ADS가 압도되어 제한 시간이 초과되고 광고 채우기 속도가 낮아지는 것을 방지하는 데 도움이 됩니다.

TPS 기반 트래픽 셰이핑 - MediaTailor는 초당 트랜잭션(TPS) 및 동시 사용자를 기반으로 요청을 제한합니다. 이 접근 방식은 ADS 용량 제한에 따라 보다 직관적인 구성을 제공합니다. 자세한 내용은 [TPS 기반 트래픽 셰이핑](#) 단원을 참조하십시오.

- 동적 변수를 설정하면 MediaTailor는 ADS에 대한 요청에 이러한 변수를 포함합니다. MediaTailor는 소비 기간 동안 이러한 변수를 사용하여 광고 시간을 미리 가져오기 일정에 일치시킵니다. 자세한 내용은 다음 사용 섹션을 참조하세요.

Example

라이브 이벤트는 7PM시부터 오후 8시 45분까지 지속되며, 해당 시간 동안 4회의 광고가 중단됩니다. 광고 시간은 예측 가능한 일정에 맞지 않습니다. 반복 프리페치를 오후 7PM에서 오후 8시 45분까지 10분의 지연 시간 및 60초의 트래픽 셰이핑 기간으로 구성합니다. 각 가용 시간이 지나면 MediaTailor는 다음 광고 시간에 대한 광고를 검색합니다. 가용 시간이 종료된 후 10분 후에 MediaTailor는 검색 요청을 ADS로 보내기 시작합니다. MediaTailor는 60초의 트래픽 형성 기간과 500,000명의 동시 사용자를 통해 ADS 요청을 분산하여 모든 요청을 동시에 보내는 대신 60초 동안 초당 평균 약 8,333개의 트랜잭션(500,000명의 사용자/60초=초당 8,333개의 요청)을 달성합니다.

검색 구성에는 동적 변수 키 `scte.event`와 값이 포함됩니다¹²³⁴. MediaTailor는 ADS에 대한 요청에 이 변수를 포함하며, 이 변수를 사용하여 특정 광고자를 이벤트 ID 1234로 대상으로 지정할 수 있습니다.

소비

MediaTailor는 SCTE-35 광고 중단 마커를 발견하면 미리 가져온 광고를 광고 중단 시간에 배치합니다.

- 검색된 광고 만료를 설정하면 지정된 만료까지 미리 가져온 광고를 삽입할 수 있습니다.
- 가용 일치 기준을 설정하지 않은 경우 MediaTailor는 소비 기간의 첫 번째 휴식 시간에 광고를 삽입합니다.
- 가용 일치 기준에 대해 동적 변수 키를 설정한 경우 MediaTailor는 검색 창에서 설정한 동적 변수와 비교하여 이러한 기준을 평가합니다. 광고 시간은 가용성 일치 기준이 충족되는 경우에만 미리 가져온 광고 삽입에 사용할 수 있습니다. MediaTailor는 기준을 충족하는 첫 번째 휴식 시간에 광고를 삽입합니다.

지원되는 가용 일치 기준 목록은 [의 표에서 광고 미리 가져오기에 사용 가능 열을 참조하세요](#) [ADS 요청에 대한 MediaTailor 세션 변수](#).

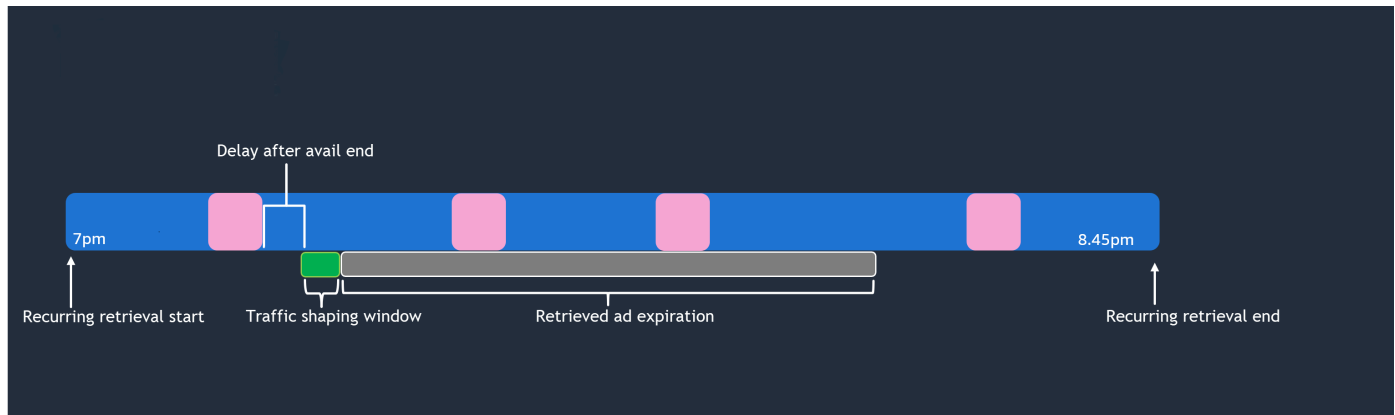
Example계속

소비 시 가용 일치 기준에 `scte.event_id` 키에를 포함합니다.

MediaTailor에서 발생하는 각 광고 시간에 대해 각 광고 시간에 대한 SCTE 이벤트 ID를 평가합니다. 각 재생 세션에서 MediaTailor는 이벤트 ID가 1234(검색 동적 변수에 정의됨)인 미리 가져온 광고를 각 광고 시간에 삽입합니다. 올바른 이벤트 ID가 포함되지 않은 광고 시간의 경우 MediaTailor는 표준 광고 삽입을 수행합니다.

광고 만료를 2700초로 설정하면 검색된 광고를 45분 동안 삽입할 수 있습니다.

다음 그림은 예를 보여 주며, 작은 사각형은 광고 시간을 나타냅니다. 반복 미리 가져오기 일정 설정은 이벤트 타임라인을 따라 표시됩니다.



미리 가져오기 비용 이해

광고 검색 요청에는 비용이 들지 않습니다. 그러나 미리 가져오기 광고 검색의 경우 MediaTailor가 트랜스코딩하는 미리 가져온 광고에 대해 표준 트랜스코딩 요금이 부과됩니다. 미리 가져오기 광고 소비의 경우 MediaTailor가 광고 시간에 배치하는 미리 가져오기된 광고의 광고 삽입에 대한 표준 요금이 부과됩니다. 트랜스코딩 및 광고 삽입 비용에 대한 자세한 내용은 [AWS Elemental MediaTailor 요금을](#) 참조하세요.

미리 가져오기 일정 생성

다음 절차에서는 MediaTailor 콘솔을 사용하여 미리 가져오기 일정을 생성하는 방법을 설명합니다. MediaTailor API를 사용하여 프로그래밍 방식으로 미리 가져오기 일정을 생성하고 관리하는 방법에 대한 자세한 내용은 API 참조의 [PrefetchSchedules](#)를 AWS Elemental MediaTailor 참조하세요.

Note

MediaTailor에서 미리 가져오기 일정을 구성할 때는 다양한 유형의 변수가 처리되는 방식을 이해하는 것이 중요합니다.

가용성 일치 기준

일정에서 가용 일치 기준을 사용하려면 먼저 [동적 세션 변수](#)로 재생 구성의 ADS URL 템플릿을 구성해야 합니다. 그렇지 않으면 가용 일치 기준이 영향을 미치지 않습니다. 동적 세션 변수 작업에 대한 자세한 내용은 MediaTailor 광고 삽입 시작하기 주제 [3단계: ADS 요청 URL 및 쿼리 파라미터 구성](#)의 섹션을 참조하세요.

미리 가져오기 일정의 플레이어 변수

미리 가져오기 일정을 생성할 때 미리 가져오기 구성에서 플레이어 변수를 동적 변수로 정의하지 마십시오. 대신 세션 시작 시와 마찬가지로 플레이어 변수를 전달합니다. 변수가 ADS 템플릿 URL에 매핑된 경우 MediaTailor는 이러한 변수를 미리 가져오기 광고 요청에 자동으로 포함합니다.

콘솔을 사용하여 새 미리 가져오기 일정을 생성하려면

1. <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.
2. 탐색 창에서 구성을 선택합니다. 미리 가져오기 일정을 생성할 재생 구성을 선택합니다.
3. 미리 가져오기 일정 탭에서 미리 가져오기 일정 추가를 선택합니다.
4. 미리 가져오기 일정 세부 정보 창에서 다음을 수행합니다.
 - 이름에 my-prefetch-schedule과 같은 미리 가져오기 일정의 식별자를 입력합니다.
 - 스트림 ID에 선택적으로 고유 ID를 입력합니다. 오리진에 여러 재생 스트림이 포함된 경우 ID를 사용하여 MediaTailor에 특정 스트림에 광고를 배치하도록 지시할 수 있습니다. 예를 들어 재생 구성에 스포츠 스트림과 TV 프로그램 스트림이 있는 경우 스트림 ID를 사용하여 미리 가져오기 일정을 생성하여 스포츠 스트림을 대상으로 하는 광고를 삽입할 수 있습니다. 클라이언트의 세션 초기화 또는 매니페스트 요청에서 스트림 ID 값을 MediaTailor에 전달합니다. 자세한 내용은 다음 예제를 참조하십시오.
 - 서버 측 추적의 경우 MediaTailor 엔드포인트에 대한 클라이언트의 GET HTTP 요청에 ?aws.streamId 쿼리 파라미터와 값을 포함합니다. 서버 측 추적에 대한 일반적인 내용은 섹션을 참조하세요 [MediaTailor 서버 측 광고 추적 및 보고](#). 스트림 ID가 포함된 HLS 엔드포인트에 대한 매니페스트 요청은 다음과 같습니다. 여기서 *myStreamId*는 스트림 ID의 이름입니다.

```
GET <mediatailorURL>/v1/master/<hashed-account-id>/<origin-id>/<asset-id>?
aws.streamId=myStreamId
```

- 클라이언트 측 추적의 경우 MediaTailor/v1/session 엔드포인트에 대한 클라이언트의 POST HTTP 세션 초기화 요청 본문에 streamId 키와 값을 포함합니다. 클라이언트 측 추적에 대한 일반적인 내용은 섹션을 참조하세요 [클라이언트 측 광고 추적](#). 스트림 ID가 포함된 세션 초기화 요청은 다음과 같습니다. 여기서 *myStreamId*는 스트림 ID의 이름입니다.

```
POST <mediatailorURL>/v1/session/<hashed-account-id>/<origin-id>/<asset-id>
```

```
{
  'streamId': 'myStreamId',
  'reportingMode': 'client'
}
```

5. 미리 가져오기 유형에서 선택하고 추가 필드에 도움이 필요한 해당 섹션을 선택합니다.

- 이벤트에서 하나의 광고 시간에 대해 하나의 미리 가져오기 일정을 생성하는 경우 단일을 선택합니다.
- 이벤트에서 각 광고가 중단되기 전에 광고를 자동으로 미리 가져오는 일정을 생성하는 경우 반복을 선택합니다.

단일 미리 가져오기 일정

이벤트에서 하나의 광고 가능 시간 전에 광고를 미리 가져오는 일정을 생성합니다.

1. 검색 창에서 사용하려는 검색 설정을 지정합니다. 이러한 설정은 MediaTailor가 ADS에서 광고를 미리 가져오는 시기를 결정합니다. 또한 ADS에 대한 요청에 포함할 동적 세션 변수도 결정합니다.
 - 시작 시간에 MediaTailor가 이 광고 시간에 대한 미리 가져오기 검색을 시작할 수 있는 시간을 입력합니다. MediaTailor는 이 시간 또는 그 이후에 클라이언트가 수행한 매니페스트 요청에 대한 광고를 미리 가져오려고 시도합니다. 기본값은 현재 시간입니다. 값을 지정하지 않으면 서비스가 가능한 한 빨리 미리 가져오기 검색을 시작합니다.
 - 종료 시간에 MediaTailor가 이 광고 시간에 대한 광고 미리 가져오기를 중지할 시간을 입력합니다. MediaTailor는 이 시점 또는 그 이전에 발생하는 매니페스트 요청에 대한 광고를 미리 가져오려고 시도합니다. 검색 기간은 소비 기간과 겹칠 수 있습니다.
 - 선택적으로 한 번에 ADS에 대한 요청 수를 제한하도록 트래픽 셰이핑을 구성합니다. 다음 방법 중 하나를 선택합니다.

기간 접근 방식: 트래픽 셰이핑 기간 지속 시간에 MediaTailor가 ADS에 요청을 배포해야 하는 초 수를 입력합니다. 자세한 내용은 [단일 미리 가져오기 일정 검색 설명](#)을 참조하세요.

TPS 기반 접근 방식: 초당 트랜잭션 수 및 동시 사용자를 기반으로 요청을 제한하도록 피크 TPS 및 피크 동시 사용자를 구성합니다. 자세한 내용은 [TPS 기반 트래픽 셰이핑](#) 단원을 참조하십시오.

- [동적 변수](#) 섹션에서 최대 100개의 동적 세션 변수를 입력합니다. MediaTailor는 ADS로 보내는 미리 가져오기 요청에서 이러한 변수를 대체에 사용합니다. 동적 세션 변수를 입력하지 않으면 MediaTailor는 [ADS](#) URL에 포함된 동적 변수의 값을 보간하기 위해 최선을 다합니다.

- 동적 변수 추가를 선택합니다.
 - 키에와 같은 동적 세션 변수 키를 입력합니다 `scte.event_id`. MediaTailor가 지원하는 모든 동적 변수를 사용할 수 있습니다. 동적 세션 변수에 대한 자세한 내용은 섹션을 참조하세요 [ADS 요청에 대한 MediaTailor 세션 변수](#).
 - 값에 *my-event*와 같은 동적 변수 값을 입력합니다.
 - 다른 동적 변수를 추가하려면 동적 변수 추가를 선택합니다.
2. 사용 창에서 사용 기간에 사용할 설정을 지정합니다. 이러한 설정은 MediaTailor가 광고를 광고 시간에 배치하는 시기를 결정합니다. 또한 사용하려는 가용 일치 기준도 결정합니다.
- 시작 시간에 MediaTailor가 미리 가져온 광고를 광고 시간에 배치하기 시작할 시간을 입력합니다. 기본값은 현재 시간입니다. 시간을 지정하지 않으면 서비스가 가능한 한 빨리 미리 가져오기 소비를 시작합니다.
 - 종료 시간에 MediaTailor가 미리 가져온 광고를 광고 시간에 배치하는 것을 중지할 시간을 입력합니다. MediaTailor는 이 시점 또는 그 이전에 발생하는 클라이언트의 매니페스트 요청에 대한 광고를 미리 가져오려고 시도합니다. 종료 시간은 시작 시간 이후여야 하며 지금부터 1일 이내여야 합니다. 사용 기간은 검색 기간과 겹칠 수 있습니다.
 - [가용 일치 기준](#) 섹션에서 가용 기준 추가를 선택하고 일정에 최대 5개의 가용 일치 기준을 추가합니다. 그런 다음 동적 변수 키에서와 같은 동적 변수 키를 추가합니다 `scte.event_id`. MediaTailor는 클라이언트가 MediaTailor에 전달하거나 MediaTailor가 세션 데이터와 같은 정보에서 추론하는 동적 변수 값으로 정의된 기준을 충족하는 경우에만 미리 가져온 광고를 광고 시간에 배치합니다. MediaTailor 광고 중단이 지정된 일치 기준을 충족하지 않는 경우 MediaTailor는 해당 중단에 대한 미리 가져오기를 건너뛵니다. 자세한 내용은 [단일 미리 가져오기 일정 소비 설명](#)을 참조하세요.

Important

미리 가져오기 일정당 최대 5개의 가용 일치 기준을 지정할 수 있습니다. 여러 기준을 지정하는 경우 MediaTailor는 논리적 AND를 사용하여 기준을 결합합니다. 논리적 OR 조합을 모델링하려면 여러 사전 가져오기 일정을 생성합니다.

광고 시간당 변경되는 변수(예: `scte.event_id`, `scte.segmentation_upid` 또는 `avail.index`)만 가용성 일치 기준으로 사용합니다. 플레이어 변수는 전체 세션 동안 일정하게 유지되고 광고 시간을 구분하지 않으므로 가용성 일치 기준에 플레이어 변수(예: `player_params.*`)를 사용하지 마세요. 변수가 ADS 템플릿 URL에 매핑된 경우 MediaTailor는 미리 가져오기 광고 요청에 플레이어 변수를 자동으로 포함합니다.

3. 가용 기준 추가를 선택합니다.

미리 가져오기 일정은 소비 기간의 종료 시간 이후에 자동으로 만료됩니다. 진단 목적으로 최소 7일 동안 표시되며, 그 이후에는 MediaTailor가 자동으로 삭제합니다. 또는 언제든지 미리 가져오기 일정을 수동으로 삭제할 수 있습니다. 미리 가져오기 일정을 수동으로 삭제하는 방법에 대한 자세한 내용은 다음 [the section called “미리 가져오기 일정 삭제”](#) 섹션을 참조하세요.

클라이언트가 CreatePrefetchSchedule API를 호출해야 하는 빈도 결정

광고 시간이 정확히 언제 발생할지 알고 있는 경우 클라이언트는 하루에 한 번

[CreatePrefetchSchedule](#) API를 프로그래밍 방식으로 호출하여 검색 및 소비를 설정할 수 있습니다. 또는 클라이언트는 하루 중 여러 번 API를 호출하여 검색 및 소비를 정의할 수 있습니다. API 호출 빈도를 선택할 때는 [최대 활성 프리페치 일정 수](#)와 프리페치 일정(들)을 생성한 후 광고 중단 일정이 변경될 가능성을 고려합니다. 미리 가져오기 일정을 생성한 후 광고 중단 일정이 변경될 가능성이 있는 경우 API를 더 자주 호출할 수 있습니다.

반복 미리 가져오기 일정

이벤트의 각 광고 가능 시간 전에 광고를 미리 가져오는 일정을 생성합니다.

1. 반복 검색 창에서 사용하려는 검색 설정을 지정합니다. 이러한 설정은 MediaTailor가 ADS에서 광고를 미리 가져오는 시기를 결정합니다. 또한 ADS에 대한 요청에 포함할 동적 세션 변수도 결정합니다.
 - 반복 미리 가져오기 창에 MediaTailor가 광고 시간에 대한 미리 가져오기 검색을 시작할 수 있는 시간을 입력합니다. MediaTailor는 이 시간 또는 그 이후에 클라이언트가 수행한 매니페스트 요청에 대한 광고를 미리 가져오려고 시도합니다. 기본값은 현재 시간입니다. 값을 지정하지 않으면 서비스가 가능한 한 빨리 미리 가져오기 검색을 시작합니다.
 - 가용성 종료 후 지연에 MediaTailor가 다음 가용성에 대한 광고를 미리 가져오기 전에 가용성이 종료된 후 대기해야 하는 초 수를 입력합니다. 값을 지정하지 않으면 MediaTailor는 지연 없이 기본 설정됩니다.
 - 선택적으로 한 번에 ADS에 대한 요청 수를 제한하도록 트래픽 셰이핑을 구성합니다. 다음 방법 중 하나를 선택합니다.

기간 접근 방식: 트래픽 셰이핑 기간 지속 시간에 MediaTailor가 ADS에 요청을 배포해야 하는 초 수를 입력합니다. 자세한 내용은 [반복 미리 가져오기 일정 가져오기 설명](#)을 참조하세요.

TPS 기반 접근 방식: 초당 트랜잭션 수 및 동시 사용자를 기반으로 요청을 제한하도록 피크 TPS 및 피크 동시 사용자를 구성합니다. 자세한 내용은 [TPS 기반 트래픽 셰이핑](#) 단원을 참조하십시오.

- [동적 변수](#) 섹션에서 최대 100개의 동적 세션 변수를 입력합니다. MediaTailor는 ADS로 보내는 미리 가져오기 요청에서 이러한 변수를 대체에 사용합니다. 동적 세션 변수를 입력하지 않으면 MediaTailor는 [ADS](#) URL에 포함된 동적 변수의 값을 보간하기 위해 최선을 다합니다.
 - 동적 변수 추가를 선택합니다.
 - 키에와 같은 동적 세션 변수 키를 입력합니다 `scte.event_id`. MediaTailor가 지원하는 모든 동적 변수를 사용할 수 있습니다. 동적 세션 변수에 대한 자세한 내용은 섹션을 참조하세요 [요ADS 요청에 대한 MediaTailor 세션 변수](#).
 - 값에 *my-event*와 같은 동적 변수 값을 입력합니다.
 - 다른 동적 변수를 추가하려면 동적 변수 추가를 선택합니다.
2. 사용 창에서 사용 기간에 사용할 설정을 지정합니다. 이러한 설정은 MediaTailor가 광고를 광고 시간에 배치하는 시기를 결정합니다. 또한 사용하려는 가용 일치 기준도 결정합니다.
- 검색된 광고 만료의 경우 검색 후 광고를 삽입할 수 있는 기간을 지정합니다.
 - [가용 일치 기준](#) 섹션에서 가용 기준 추가를 선택하고 일정에 최대 5개의 가용 일치 기준을 추가합니다. 그런 다음 동적 변수 키에서와 같은 동적 변수 키를 추가합니다 `scte.event_id`. MediaTailor는 클라이언트가 MediaTailor에 전달하거나 MediaTailor가 세션 데이터와 같은 정보에서 추론하는 동적 변수 값으로 정의된 기준을 충족하는 경우에만 미리 가져온 광고를 광고 시간에 배치합니다. MediaTailor 광고 중단이 지정된 일치 기준을 충족하지 않는 경우 MediaTailor는 해당 중단에 대한 미리 가져오기를 건너뛵니다. 자세한 내용은 [반복 미리 가져오기 일정 소비 설명](#)을 참조하세요.

Important

미리 가져오기 일정당 최대 5개의 가용 일치 기준을 지정할 수 있습니다. 여러 기준을 지정하는 경우 MediaTailor는 논리적 AND를 사용하여 기준을 결합합니다. 논리적 OR 조합을 모델링하려면 여러 사전 가져오기 일정을 생성합니다.

광고 시간당 변경되는 변수(예: `scte.event_id`, `scte.segmentation_upid` 또는 `avail.index`)만 가용성 일치 기준으로 사용합니다. 플레이어 변수는 전체 세션 동안 일정하게 유지되고 광고 시간을 구분하지 않으므로 가용성 일치 기준에 플레이어 변수(예: `player_params.*`)를 사용하지 마세요. 변수가 ADS 템플릿 URL에 매핑된 경우 MediaTailor는 미리 가져오기 광고 요청에 플레이어 변수를 자동으로 포함합니다.

3. 가용 기준 추가를 선택합니다.

미리 가져오기 일정은 소비 기간의 종료 시간 이후에 자동으로 만료됩니다. 진단 목적으로 최소 7일 동안 표시되며, 그 이후에는 MediaTailor가 자동으로 삭제합니다. 또는 언제든지 미리 가져오기 일정을 수동으로 삭제할 수 있습니다. 미리 가져오기 일정을 수동으로 삭제하는 방법에 대한 자세한 내용은 다음 [the section called “미리 가져오기 일정 삭제”](#) 섹션을 참조하세요.

TPS 기반 트래픽 셰이핑

AWS Elemental MediaTailor 는 한 번에 ADS에 대한 요청 수를 제한하는 두 가지 선택적 트래픽 셰이핑 접근 방식을 제공합니다. TPS 기반 트래픽 셰이핑은 미리 가져오기 일정을 위한 시간 범위 기반 트래픽 셰이핑의 대안을 제공합니다. 이 접근 방식은 시간 계산 대신 초당 트랜잭션 수(TPS) 및 예상 동시 사용자 측면에서 광고 결정 서버(ADS) 용량을 지정할 수 있도록 하여 보다 직관적인 구성을 제공합니다.

TPS 기반 트래픽 셰이핑의 작동 방식

검색 기간 기간을 지정하는 대신 다음 파라미터를 제공합니다.

피크 TPS

ADS가 처리할 수 있는 초당 최대 요청 수입니다. 이 파라미터에는 기본값이 없습니다.

피크 동시 사용자

콘텐츠에 대한 동시 뷰어의 예상 피크 수입니다. 이 파라미터에는 기본값이 없습니다.

MediaTailor는 동시 세션 수에 관계없이 지정된 TPS 한도 내에서 유지되도록 미리 가져오기 요청을 시간 경과에 따라 자동으로 분산합니다.

Example TPS 기반 구성 예제

ADS는 500TPS를 처리할 수 있으며 피크 시간에는 100,000명의 동시 뷰어가 필요합니다. 다음을 구성합니다.

- 피크 TPS: 500
- 최대 동시 사용자: 100,000명

MediaTailor는 동시 세션 수에 관계없이 지정된 TPS 한도 내에서 유지되도록 미리 가져오기 요청을 시간 경과에 따라 자동으로 분산합니다.

미리 가져오기 일정 삭제

다음 절차에서는 MediaTailor 콘솔을 사용하여 미리 가져오기 일정을 삭제하는 방법을 설명합니다. MediaTailor API를 사용하여 프로그래밍 방식으로 미리 가져오기 일정을 삭제하는 방법에 대한 자세한 내용은 API 참조의 [DeletePrefetchSchedule](#)을 AWS Elemental MediaTailor 참조하세요.

Note

삭제는 실시간으로 발생하지 않습니다. MediaTailor가 미리 가져오기 일정을 삭제하는 동안 지연이 발생할 수 있으며, 이 시간 동안 미리 가져오기 가져오기 및 소비는 백그라운드에서 계속 실행됩니다.

콘솔을 사용하여 미리 가져오기 일정을 삭제하려면

1. <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.
2. 탐색 창에서 구성을 선택합니다. 삭제하려는 미리 가져오기 일정(들)이 포함된 재생 구성을 선택합니다.
3. 미리 가져오기 일정 탭에서 삭제할 미리 가져오기 일정을 선택합니다. 그런 다음 삭제를 선택합니다.

에서 사전 조정된 광고 사용 AWS Elemental MediaTailor

[일반적인 광고 삽입 워크플로](#)에서 MediaTailor는 콘텐츠 스트림과 일치하도록 광고를 동적으로 트랜스코딩하고, 저장하고, 광고를 라이브 스트림에 연결합니다. 이 프로세스는 MediaTailor가 광고 결정 서버(ADS)로부터 VAST 응답으로 광고를 수신한 후에만 수행되므로 광고를 스티칭할 수 있는 시간이 지연됩니다. 광고 스티칭 워크플로에 추가 지연 시간이 발생하는 경우(ADS 제한 시간 또는 기타 콘텐츠 또는 네트워크 문제로 인해) MediaTailor는 시간을 부분적으로 채우거나 광고 시간을 완전히 놓칠 수 있습니다.

콘텐츠에 광고를 연결하는 데 필요한 시간을 줄이기 위해 사전 조정된 광고를 사용할 수 있습니다. 사전 조정된 광고는 MediaTailor 광고 삽입에서 사용하기 전에 트랜스코딩하는 광고입니다. ADS에 조건 없는 광고의 URLs을 제공하는 대신 사전 조건 있는 광고의 URLs을 제공합니다. MediaTailor 요청에 대한 VAST 응답에서 ADS에는 사전 조정된 광고에 대한 직접 링크가 포함됩니다. MediaTailor는 광고 스티칭의 트랜스코딩 부분을 제거하여 광고를 저장하고 콘텐츠 스트림에 연결하기만 하면 됩니다. 사전 조정된 광고가 포함된 광고 스티칭 프로세스는 MediaTailor가 VAST 응답을 통해 광고를 인식하는 시점과 광고가 콘텐츠에 스티칭되는 시점 사이의 시간을 줄입니다.

또는 광고 미리 가져오기를 사용할 수도 있습니다. 즉, 광고 시간이 필요하기 전에 예약된 시간에 광고 스티칭 프로세스를 수행하도록 MediaTailor를 구성하는 경우입니다. 광고 미리 가져오기에 대한 자세한 내용은 섹션을 참조하세요 [광고 미리 가져오기](#).

사전 조건화된 광고 요구 사항

다음은 사전 조정된 광고로 광고 스티칭 워크플로를 설정할 때 고려해야 할 요구 사항입니다.

MediaFiles 요구 사항

광고 서버가 MediaTailor에 보내는 VAST 응답에는 다음 요구 사항을 충족하는 MediaFiles가 포함되어야 합니다.

광고(Creative)에는 콘텐츠 스트림의 비트레이트 변형을 준수하는 변형이 있어야 합니다. VAST 응답이 템플릿 매니페스트와 일치하는 올바른 광고 변형을 사용하도록 하는 것은 사용자의 책임입니다.

사전 조정된 광고를 사용하면 광고 삽입의 효율성을 높이는 데 도움이 될 수 있지만 MediaTailor는 트랜스코딩 프로세스를 관리하여 광고의 미디어 파일이 콘텐츠 매니페스트 사양과 호환되는지 확인할 수 없습니다. 광고가 콘텐츠 스트림과 일치하지 않으면 MediaTailor가 삽입을 놓치거나 불일치로 인해 재생 디바이스에 오류가 발생할 수 있습니다.

또한 MediaTailor 트랜스코딩 없이 콘텐츠 스트림에 연결하려면 다음 요구 사항을 충족해야 MediaFile 합니다.

- MediaTailor가 다운로드할 수 있도록 퍼블릭 인터넷에서 액세스할 수 있어야 합니다.
- VAST 응답 `delivery="streaming"`에서 로 표시된 스트리밍 전송을 사용해야 합니다.
- (HLS.m3u8의 경우) 또는 .mpd (DASH의 경우) 파일이어야 합니다.

Example VAST 응답

다음 예제 VAST 응답에서 MediaTailor는 다음 URLs과 MediaFile 함께를 삽입합니다.

- HLS 스트림의 경우 MediaTailor를 사용합니다 `https://example-ad-origin.amazonaws.com/ad1/index_low.m3u8`. 스트리밍 MediaFile 전송과 지원되는 파일 확장명(.m3u8)이 있는 첫 번째입니다.
- DASH 스트림의 경우 MediaTailor를 사용합니다 `https://example-ad-origin.amazonaws.com/ad1/index.mpd`. 스트리밍 MediaFile 전송과 지원되는 파일 확장명(.mpd)이 있는 첫 번째입니다.

```

<?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>

```

광고 매니페스트 요구 사항

사전 조정된 광고를 사용하려면 상위 및 하위 광고 매니페스트가 다음 요구 사항을 충족해야 합니다.

- VAST 응답의 Creative 섹션에 연결된 매니페스트는 상위 광고 매니페스트여야 합니다.

- 하위 광고 매니페스트의 URLs은 상대 경로여야 합니다.
- 하위 광고 매니페스트는 상위 다변량 재생 목록과 동일한 디렉터리에 동일한 수준으로 있어야 합니다. 하위 매니페스트는 하위 디렉터리 또는 다른 위치에 있을 수 없습니다.

Example지원되는 상위 다변량 재생 목록

다음 상위 광고 다변량 재생 목록에는 하위 광고 미디어 재생 목록URLs이 포함되어 있습니다. 하위 재생 목록은 상위 다변량 재생 목록과 동일한 디렉터리에도 있습니다.

```
#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
```

Example지원되지 않는 상위 다변량 재생 목록: 하위 디렉터리

다음 상위 광고 다변량 재생 목록에는 상위 다변량 재생 목록에 상대적인 하위 디렉터리에 있는 하위 재생 목록이 포함되어 있습니다. 사전 조정된 광고에는 지원되는 재생 목록이 아닙니다.

```
#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
```

Example지원되지 않는 상위 다변량 재생 목록: 절대 URLs

다음 상위 광고 다변량 재생 목록에는 절대 URLs. 사전 조정된 광고에는 지원되는 재생 목록이 아닙니다.

```
#EXTM3U
#EXT-X-STREAM-INF:BANDWIDTH=150000,RESOLUTION=416x234,CODECS="avc1.42e00a,mp4a.40.2"
https://example.mediataylor.us-west-2.amazonaws.com/index_1.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=440000,RESOLUTION=416x234,CODECS="avc1.42e00a,mp4a.40.2"
https://example.mediataylor.us-west-2.amazonaws.com/index_2.m3u8
```



```
#EXT-X-STREAM-INF:BANDWIDTH=640000,RESOLUTION=640x360,CODECS="avc1.42e00a,mp4a.40.2"
https://example.mediatailor.us-west-2.amazonaws.com/index_3.m3u8
```

사전 조절된 광고 워크플로

다음은 MediaTailor를 사용한 광고 스티칭 워크플로에서 사전 조정된 광고가 작동하는 방식에 대한 기본 설명입니다. 워크플로의 첫 번째 부분은 사전 조정된 광고를 사용하도록 설정하기 위해 수행해야 하는 작업입니다. 두 번째 부분에서는 MediaTailor가 광고를 처리하는 방법을 설명합니다.

1부: 사전 조건 광고 설정

다음 단계를 완료하여 MediaTailor에서 사전 조정된 광고를 사용하는 워크플로를 설정합니다.

1. 와 같은 트랜스코더 서비스를 사용하여 템플릿 매니페스트의 다양한 비트 전송률, 해상도 및 코덱을 지원하는 변형으로 크리에이티브를 AWS Elemental MediaConvert조건화합니다.
2. VAST 응답에 사용할 수 있도록 사전 트랜스코딩된 미디어 파일의 URLs을 ADS에 제공합니다.
3. MediaTailor에서 [재생 구성을 생성합니다](#). 사전 조정된 광고를 사용하려면 구성의 스트리밍 미디어 파일 조정 설정에서 **없음**을 선택합니다.
4. 평소와 같이 콘텐츠 전송 설정을 계속합니다.

2부: MediaTailor 광고 처리

MediaTailor 광고 스티칭은에 설명된 대로 완료됩니다 [MediaTailor 광고 삽입 작동 방식](#). MediaTailor는 ADS로부터 VAST 응답을 받으면 다음 로직을 사용하여 광고에 대해 수행할 작업을 결정합니다. 이 로직은 재생 구성의 스트리밍 미디어 파일 컨디셔닝 설정에 따라 결정됩니다.

- 스트리밍 미디어 파일 컨디셔닝이 트랜스코딩으로 설정된 경우 MediaTailor는 progressive 전송으로 미디어 파일을 트랜스코딩하고 매니페스트에 연결합니다. progressive 전송 미디어 파일이 있는 광고가 공간을 채우기에 충분하지 않은 경우 MediaTailor는 이를 트랜스코딩하고 streaming 전송에 사용합니다.
- 스트리밍 미디어 파일 컨디셔닝이 없음으로 설정된 경우 MediaTailor는 streaming 전송 미디어 파일이 포함된 광고를 트랜스코딩하지 않고 매니페스트에 연결합니다. streaming 전송 미디어 파일이 있는 광고가 공간을 채우기에 충분하지 않은 경우 MediaTailor는 이를 트랜스코딩하고 progressive 전송에 사용합니다.

MediaTailor 수익화 함수 작업

MediaTailor 수익화 함수(함수)를 사용하면가 세션 데이터를 AWS Elemental MediaTailor 관리하고, 광고 요청을 빌드하고, 광고 삽입 중에 광고를 선택하는 방법을 사용자 지정할 수 있습니다. 함수를 사용하면 외부 APIs 호출하고, 표현식을 사용하여 데이터를 변환하고, 광고 결정 서버(ADS) 요청 파라미터 또는 플레이어 파라미터를 수정하고, 스트림에 삽입되는 광고를 필터링, 재정렬 또는 보완할 수 있습니다. 사용자 지정 인프라를 배포하거나 관리할 필요가 없습니다.

함수는 JSON 데이터에 대한 간단한 쿼리 및 변환 언어JSONata인를 사용하여 표현식을 평가합니다. JSONata 표현식을 작성하여 세션 데이터를 읽고, 값을 변환하고, 출력을 정의합니다. 지원되는 JSONata 함수의 전체 목록은 섹션을 참조하세요 [JSONata 표현식 참조](#).

다음과 같은 경우 함수를 사용합니다.

- 최종 사용자 데이터로 광고 요청을 강화합니다. 세션 시작 시 자격 증명 서비스를 호출하고 맞춤형 광고 타겟팅을 위한 모든 ADS 요청에 확인된 자격 증명을 포함합니다.
- ADS 요청 파라미터를 사용자 지정합니다. 세션 데이터, SCTE-35 신호 또는 외부 API 응답을 기반으로 ADS URL, 헤더 또는 본문을 동적으로 설정합니다.
- 광고 서버 간에 A/B 테스트를 실행합니다. 세션 속성 또는 무작위 할당을 기반으로 서로 다른 ADS 엔드포인트 간에 트래픽을 분할합니다.
- 다단계 보강 파이프라인을 구축합니다. 여러 함수를 함께 연결하여 한 API에서 데이터를 가져와 변환하고 결과를 다음 단계로 전달합니다.
- 비디오 콘텐츠를 기반으로 광고를 대상으로 지정합니다. IAB 콘텐츠 분류 및 GARM 브랜드 안전 신호에 대한 Elemental Inference를 쿼리한 다음, 상황별 광고 타겟팅을 위해 상황별 메타데이터를 광고 결정 서버에 전달합니다.
- ADS에서 반환한 광고를 필터링하거나 재정렬합니다. 경쟁 분리를 위해 특정 광고 시스템에서 광고를 제거하거나 MediaTailor가 삽입할 광고를 선택하기 전에 광고를 재정렬합니다.
- 채워지지 않은 광고 시간을 자체 광고로 채웁니다. 과소 채워진 광고 시간을 감지하고 보조 VAST 엔드포인트에서 하우스 광고 또는 프로모션을 가져와 남은 시간을 채웁니다.

함수가 MediaTailor 워크플로에 맞는 방식

최종 사용자가 재생 세션을 시작하면 MediaTailor는 수명 주기 후크라는 광고 삽입 흐름의 특정 지점에서 함수를 평가합니다.

1. 함수를 정의하고 함수 매핑을 통해 재생 구성에 연결합니다.

- MediaTailor는 지정된 수명 주기 후크에서 함수를 실행합니다.
- 함수는 세션 데이터를 읽고, 선택적으로 외부 API를 호출하고, MediaTailor가 재생 처리 중에 사용하는 결과를 씁니다.

Tip

함수를 처음 사용하는 경우로 시작한 [빠른 시작 설명서](#) 다음을 읽습니다. [수명 주기 후크](#), 표현식 구문 및 전체 예제는 [JSONata 표현식 참조](#) 및 섹션을 참조하세요. [함수 예제](#).

주제

- [함수 빠른 시작 안내서](#)
- [함수 수명 주기 후크](#)
- [함수 유형 및 구성](#)
- [Elemental Inference를 사용한 컨텍스트 광고 타겟팅](#)
- [함수 생성 및 관리](#)
- [JSONata 함수에 대한 표현식 참조](#)
- [함수 예제](#)
- [함수 문제 해결 및 모니터링](#)
- [함수 제한](#)

함수 빠른 시작 안내서

이 안내서에서는 첫 번째 함수를 생성하고, 재생 구성에 연결하고, 함수가 실행되었는지 확인하는 방법을 안내합니다. 결국에는 각 최종 사용자의 디바이스 유형(ctv, mobile 또는 desktop)을 분류하고 모든 ADS 요청에서 사용할 수 있는 플레이어 파라미터로 저장하는 작동 함수가 있습니다.

사전 조건

시작하기 전에 기존 MediaTailor 재생 구성이 있는지 확인합니다. 없으면 [MediaTailor 시작하기](#) 섹션을 참조하세요.

1단계: 함수 만들기

이 단계에서는 사용자 에이전트 문자열을 기반으로 최종 사용자의 디바이스 유형을 분류하고 결과를 플레이어 파라미터에 저장하는 함수를 생성합니다. 함수는 [JSONata 표현식 참조](#) 표현식과 함께 사용자 지정 출력 유형(외부 API 호출 없음)을 사용하여 사용자 에이전트를 평가합니다.

1. [MediaTailor 콘솔](#)을 엽니다.
2. 탐색 창에서 함수를 선택합니다.
3. 함수 생성을 선택합니다.
4. 생성 마법사 모달에서 처음부터 생성을 선택한 다음 계속을 선택합니다.
5. 함수 유형에서 사용자 지정 출력 타일을 선택합니다.
6. 함수 세부 정보에서 다음을 입력합니다.
 - 함수 ID: myFirstFunction
 - 설명: Classify device type from user agent
7. 사용자 지정 출력 구성의 출력 섹션에서 행을 하나 추가합니다.
 - 키: player_params.deviceType
 - 값: {% \$contains(session.user_agent, 'CTV') ? 'ctv' : \$contains(session.user_agent, 'Mobile') ? 'mobile' : 'desktop' %}
8. 함수 생성을 선택합니다.

성공 알림은 함수가 생성되었음을 확인하며 함수 세부 정보 페이지로 리디렉션됩니다.

결과 함수 구성은 다음과 같습니다.

```
{
  "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' %}"
    }
  }
}
```

2단계: 함수를 재생 구성에 연결

함수를 재생 구성의 수명 주기 후크에 매핑합니다. 매핑은 MediaTailor에 함수를 실행할 시기를 알려줍니다.

1. 탐색 창에서 구성을 선택합니다.
2. 업데이트할 재생 구성을 선택합니다.
3. 편집을 선택합니다.
4. 함수 구성 섹션을 확장합니다.
5. 세션 초기화 후크의 경우 드롭다운myFirstFunction에서를 선택합니다.
6. 저장을 선택합니다.

이는 [세션 전 초기화](#) 수명 주기 후크myFirstFunction에 연결됩니다. 결과 함수 매핑은 다음과 같습니다.

```
{
  "FunctionMapping": {
    "PRE_SESSION_INITIALIZATION": "myFirstFunction"
  }
}
```

MediaTailor는이 재생 구성에서 모든 새 세션이 시작될 때 함수를 한 번 실행합니다.

3단계: 세션 시작 및 함수 실행 확인

새 재생 세션을 시작하여 함수를 트리거합니다. 재생 구성의 세션 초기화 엔드포인트에 세션 초기화 요청을 사용합니다.

MediaTailor는 옵트인할 필요 없이 모든 함수 실행에 대한 CloudWatch 지표를 자동으로 게시합니다. 세션을 시작한 후 AWS/MediaTailor 네임스페이스에서 다음 지표를 확인하여 함수가 실행되었는지 확인합니다.

- PreSessionInitHook.Invocations - 후크가 실행되었는지 확인합니다.
- PreSessionInitHook.Errors - 함수가 성공한 경우 0이어야 합니다.
- Function.Invocations - 실행된 개별 함수를 확인합니다. 이 지표에는 FunctionId, FunctionType및 HookType차원이 포함되어 있으므로 로 myFirstFunction 필터링할 수 있습니다.

함수가 실패하면 MediaTailor는 기본적으로 매니페스트 로그에 오류 로그 이벤트를 내보냅니다(구성 필요 없음).

- PRE_SESSION_INIT_HOOK_ERROR - `errorType` 및의 후크 수준 실패cause.
- PRE_SESSION_INIT_FUNCTION_ERROR - 특정 `functionId` 및 오류 세부 정보가 포함된 함수 수준 실패입니다.

다음 예제는 함수 표현식에서 구문 오류에 대한 PRE_SESSION_INIT_FUNCTION_ERROR 이벤트를 보여줍니다.

```
{
  "eventTimestamp": "2024-01-01T12:00:00.076000000Z",
  "eventType": "PRE_SESSION_INIT_FUNCTION_ERROR",
  "eventDescription": "Function execution failed",
  "awsAccountId": "123456789012",
  "originId": "my-config",
  "sessionId": "session-123",
  "requestId": "req-abc",
  "eventId": "5dc6f040-0f72-4e8c-a64e-25eeef62708c",
  "functionId": "myFirstFunction",
  "functionType": "CUSTOM_OUTPUT",
  "executionTimeMs": 2,
  "errorType": "SYNTAX_ERROR",
  "cause": "Expected \"\")\" before end of expression",
  "input": {}
}
```

`eventId` 필드를 사용하여 동일한 실행에 대한 후크 및 함수 오류 이벤트를 상호 연관시킵니다.

`errorType` 필드는 실패 클래스를 알려줍니다. 오류 유형 및 수정 사항의 전체 목록은 [문제 해결 및 모니터링](#) 섹션을 참조하세요.

Note

자세한 성공 로깅을 위해 매니페스트 로그 구성에서 PRE_SESSION_INIT_HOOK_SUMMARY 및 PRE_SESSION_INIT_FUNCTION_COMPLETED 이벤트를 옵트인합니다. 요약 이벤트는 모든 실행에 대한 후크 결과를 표시합니다. 완료된 이벤트에는 각 함수의 입력, 출력 및 HTTP 요청/응답 세부 정보가 표시됩니다. 로그 비용을 최소화하기 위해 기본적으로 비활성화됩니다. 자세한 내용은 [문제 해결 및 모니터링](#) 단원을 참조하십시오.

백그라운드에서 일어나는 일

다음은 방금 생성한 함수에 대한 전체 요청 흐름입니다.

1. 플레이어가 MediaTailor로 세션을 시작합니다.
2. MediaTailor는 PRE_SESSION_INITIALIZATION 수명 주기 후크를 실행하고를 실행합니다myFirstFunction.
3. 함수는 session.user_agent 필드를 평가하고 ctv, mobile또는 desktop에 씁니다player_params.deviceType.
4. MediaTailor는 세션을 생성하고 매니페스트를 플레이어에게 반환합니다.
5. 재생 중에 플레이어가 광고 시간을 경험합니다.
6. MediaTailor는 PRE_ADS_REQUEST 수명 주기 후크를 실행한 다음 ADS 요청을 구성합니다. deviceType는 플레이어 파라미터에 저장되므로 동적 변수 대체를 통해 ADS 요청 URL에 포함할 수 있습니다.
7. ADS는 디바이스 유형을 사용하여 대상 광고 크리에이티브를 반환합니다.
8. MediaTailor는 광고를 매니페스트에 연결하여 플레이어에게 반환합니다.

어떤 이유로든 함수가 실패하면 MediaTailor는 출력을 삭제하고 연결된 함수가 없는 것처럼 진행합니다. 최종 사용자는 디바이스 유형 타겟팅 없이도 여전히 광고를 볼 수 있습니다.

제안된 주제

이제 재생 구성에 작업 함수가 연결되었습니다. 여기에서:

- 각 수명 주기 후크에서 사용할 수 있는 입력 필드 및 출력 네임스페이스에 대한 자세한 내용은 섹션을 참조하세요[수명 주기 후크](#).
- 다양한 함수 유형과 이를 함께 연결하는 방법에 대해 알아보려면 섹션을 참조하세요[함수 유형 및 구성](#).
- 전체 작업 예제를 보려면 섹션을 참조하세요[함수 예제](#).

함수 수명 주기 후크

수명 주기 후크는 MediaTailor가 재생 중에 함수를 실행하는 시기를 정의합니다. 이 페이지는 입력 필드, 출력 네임스페이스 및 각 후크의 데이터 흐름을 관리하는 규칙에 대한 전체 참조입니다.

개요

MediaTailor는 네 가지 수명 주기 후크를 지원합니다.

- **PRE_SESSION_INITIALIZATION** 최종 사용자가 새 세션을 시작하면 한 번 실행됩니다. 대상 세그먼트 가져오기와 같은 일회성 설정 작업에 사용됩니다. 현재 광고 중단이 발생하지 않았으므로 광고 중단 컨텍스트를 사용할 수 없습니다.
- **PRE_ADS_REQUEST**는 모든 광고 결정 서버(ADS) 요청 전에 실행되며 스트림에서 광고 중단당 한 번 실행됩니다. 이를 사용하여 대상 데이터로 ADS 요청을 사용자 지정하거나, ADS URL을 수정하거나, 헤더를 추가할 수 있습니다.
- **POST_ADS_RESPONSE** MediaTailor가 Video Ad Serving Template(VAST) 래퍼 리디렉션의 해상도를 포함하여 ADS 응답을 수신하고 구문 분석한 후 실행됩니다. MediaTailor가 삽입할 광고를 선택하기 전에 이를 사용하여 ADS가 반환한 광고를 필터링, 재정렬, 수정 또는 보완할 수 있습니다.
- **PRE_MANIFEST_INSERTION**는 광고 선택, 트랜스코딩 검사 및 채우기 정책 직후 MediaTailor가 매니페스트에 광고를 쓰기 직전에 광고 시간 개인화가 끝날 때 실행됩니다. 이를 사용하여 과소 채워진 광고 시간에 추가 광고를 삽입하는 등 최종 광고 세트를 검사하거나 수정할 수 있습니다.

후크는 타이밍과 범위가 다릅니다.는 한 번 PRE_SESSION_INITIALIZATION 실행되고 전체 세션 동안 지속되는 데이터를 설정합니다. PRE_ADS_REQUEST는 각 ADS 상호 작용에서 POST_ADS_RESPONSE 실행됩니다.는 발신 요청을 PRE_ADS_REQUEST 구성하고 광고 선택 전에 구문 분석된 응답에 따라 POST_ADS_RESPONSE 작동합니다.는 선택이 완료된 후 PRE_MANIFEST_INSERTION 실행되며 단일 간접 호출에서 새로 개인화된 모든 광고 시간에 대한 최종 광고 세트를 확인합니다.

Important

PRE_ADS_REQUEST, POST_ADS_RESPONSE 및 PRE_MANIFEST_INSERTION 후크는 단일 요청에 대해 2,000ms의 통합 실행 예산을 공유합니다. 이 예산은 각 후크의 자체 2,000ms 제한 시간에 추가됩니다. 후크의 유효 제한 시간은 자체 제한 시간과 나머지 예산 중 작은 값입니다. 이전 후크에서 소요된 시간은 동일한 요청의 이후 후크에서 사용할 수 있는 시간을 줄입니다. 후크가 시작되기 전에 나머지 예산이 소진되면 MediaTailor는 해당 후크를 건너뛰고 해당 후크 없이 요청을 계속 처리합니다.는 세션 초기화 중에 PRE_SESSION_INITIALIZATION 실행되며 결합된 예산의 일부가 아닙니다. 자세한 내용은 [한도](#) 단원을 참조하십시오.

Note

이 설명서에서는 광고 시간을 가용성이라고도 합니다. 입력 필드 이름(예: `avail.availId` 및 `avails.avails`) 및 CloudWatch 지표는 `avail` 양식을 사용합니다.

세션 전 초기화

실행 시

MediaTailor는 새 재생 세션이 시작될 때에 매핑된 함수를 `PRE_SESSION_INITIALIZATION` 한 번 실행합니다. MediaTailor가 초기 매니페스트 응답을 구성하기 전에 함수가 실행됩니다.

Input

`session.*`, `player_params.*` 및 `event.*`. 사용 가능한 모든 필드는 섹션을 참조하세요 [입력 필드 참조](#).

허용된 출력 네임스페이스

네임스페이스	허용되는 유형
<code>player_params.*</code>	문자열, 숫자, 부울

에 기록된 값은 세션에 `player_params.*` 유지됩니다. 다음과 같이 사용할 수 있습니다.

- 를 통한 `PRE_ADS_REQUEST` 수명 주기 후크의 입력으로 `player_params.*`
- 를 통한 ADS 요청 URLs에서 [ADS 요청에 대한 MediaTailor 동적 광고 변수](#)(예: `[player_params.deviceType]`)
- 모든 광고 시간 동안 세션 수명 동안

Note

모든 `player_params` 출력 키 및 값의 직렬화된 총 크기는 1,000자를 초과할 수 없습니다. 합계가 이 제한을 초과하면 함수 출력이 삭제됩니다. 자세한 내용은 [함수 제한](#) 단원을 참조하십시오.

일반적인 사용 사례

- 외부 서비스에서 자격 증명 또는 대상 데이터를 가져와 나중에 ADS 요청에 사용할 수 있도록 플레이어 파라미터에 저장합니다.
- 사용자 에이전트를 기반으로 디바이스 유형을 분류하고 플레이어 파라미터에 분류를 작성합니다.
- 다운스트림 광고 중단 처리가 의존하는 기본 플레이어 파라미터 값을 설정합니다.
- 를 통해 ADS URL에 포함된 플레이어 파라미터에 값을 저장합니다 [ADS 요청에 대한 MediaTailor 동적 광고 변수](#).

실패 특성

어떤 이유로든 연결된 함수가 PRE_SESSION_INITIALIZATION 실패하면 MediaTailor는 함수의 출력을 삭제하고 연결된 함수가 없는 것처럼 진행합니다. 세션은 함수의 플레이어 파라미터 값 없이 정상적으로 시작됩니다.

사전 광고 요청

실행 시

MediaTailor는 ADS에 요청을 보내기 직전에 광고 중단당 PRE_ADS_REQUEST 한 번에 매핑된 함수를 실행합니다. 함수는 매니페스트 처리 중에 광고 기회가 발생할 때마다 실행됩니다.

Input

PRE_SESSION_INITIALIZATION, , avail.*, scte.*, inference.*, asset.*의 모든 필드adsRequest.*(url, 메서드, 헤더, 본문). 사용 가능한 모든 필드는 섹션을 참조하세요 [입력 필드 참조](#).

Elemental Inference 컨텍스트 메타데이터 변수(inference.*)

업스트림 인코더(예: AWS Elemental MediaLive)가 컨텍스트 메타데이터 보강을 위해 구성된 경우 SCTE-35 광고 중단 마커에 Elemental Inference 쿼리 파라미터를 포함합니다. MediaTailor는 이러한 마커를 자동으로 구문 분석하고 inference 네임스페이스에 다음과 같은 읽기 전용 변수를 노출합니다. 이러한 변수는 Elemental Inference를 호출하는 AWS_SERVICE_REQUEST 함수를 동적으로 구성하기 위한 기본 메커니즘입니다.

Note

`inference.*` 변수는 `PRE_ADS_REQUEST` 후크에서만 사용할 수 있습니다. `PRE_SESSION_INITIALIZATION`에서는 사용할 수 없습니다.

Field	유형	설명
<code>inference.enriched</code>	부울	<code>true</code> SCTE-35 마커에서 요소 추론 데이터를 찾아 성공적으로 구문 분석한 경우. 다른 <code>inference.*</code> 필드를 사용하기 전에 이를 확인합니다.
<code>inference.feedId</code>	문자열	Elemental Inference 피드 식별자입니다.
<code>inference.dataEndpoint</code>	문자열	Elemental Inference 피드의 데이터 엔드포인트 URL 접두어입니다. 이를 사용하여 전체 <code>GetMetadata</code> URL을 구성합니다.
<code>inference.pts</code>	Long	광고 중단 신호에 해당하는 프레젠테이션 타임스탬프(PTS)입니다. 쿼리 기간의 끝으로 사용됩니다.
<code>inference.timescale</code>	Long	PTS 값 해석 기간입니다(예: 90kHz의 경우 90000).
<code>inference.region</code>	문자열	쿼리 URL에서 추출된 AWS 리전입니다. 함수의 <code>TargetRegion</code> 로 사용됩니다. <code>AWS_SERVICE_REQUEST</code> .

Field	유형	설명
<code>inference.previousBreakEndPts</code>	Long	이전 광고 시간 이후에 콘텐츠가 재생된 PTS입니다. 이를 쿼리 기간의 시작으로 사용하여 휴식 시간 사이에 재생된 콘텐츠만 분석합니다. 세션의 첫 번째 광고 시간에는 없습니다.
<code>inference.parseError</code>	부울	true Elemental Inference 마커를 찾았지만 구문 분석할 수 없는 경우. 업스트림 인코더의 구성 문제를 나타냅니다.

이러한 변수는 읽기 전용입니다. 함수는 `inference.*` 값을 수정할 수 없습니다.

`inference.enriched`이 false 있거나 없는 경우 나머지 `inference.*` 필드를 사용할 수 없습니다. 이 사례를 정상적으로 처리하려면 JSONata 표현식에 대체 값이 포함되어야 합니다.

이러한 변수를 실제로 사용하는 방법은 단원을 참조하십시오 [Elemental Inference 통합](#).

허용된 출력 네임스페이스

네임스페이스	허용되는 유형	출력 사용 방법
<code>player_params.*</code>	문자열, 숫자, 부울	이 광고 시간에 대한 세션 플레이어 파라미터를 재정의합니다. 를 통해 ADS 요청 URL에 사용할 수 있습니다 ADS 요청에 대한 MediaTailor 동적 광고 변수 .
<code>session.*</code>	문자열, 숫자, 부울	이 광고 시간에 대한 세션 변수를 재정의합니다. 동적 변수 대체를 통해 ADS 요청 URL에 사용할 수 있습니다.

네임스페이스	허용되는 유형	출력 사용 방법
avail.*	문자열, 숫자, 부울	이 광고 시간에 대한 가용 변수를 재정의합니다. 동적 변수 대체를 통해 ADS 요청 URL에 사용할 수 있습니다.
scte.*	문자열, 숫자, 부울	이 광고 시간에 대한 SCTE 변수를 재정의합니다. 동적 변수 대체를 통해 ADS 요청 URL에 사용할 수 있습니다.
asset.*	문자열, 숫자, 부울	이 광고 시간에 대한 자산 메타데이터 변수(EXT-X-ASSET 태그에서)를 재정의합니다. 동적 변수 대체를 통해 ADS 요청 URL에 사용할 수 있습니다. 자세한 내용은 the section called “광고 마커” 단원을 참조하십시오.
adsRequest.*	문자열	이 광고 시간에 대한 ADS 요청만 재정의합니다. 지원되는 필드: url, method, headers.<name> , body.url 값은 템플릿으로 취급되며 함수가 실행된 ADS 요청에 대한 MediaTailor 동적 광고 변수 후를 지원합니다. 지속되지 않습니다.

Note

PRE_ADS_REQUEST 후크의 모든 출력은 일시적 재정의입니다. 이는 현재 광고 시간의 ADS 요청에만 적용되며 세션에는 지속되지 않습니다.

예 - ADS 요청 재작성:

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

이렇게 하면 ADS URL이 대체되고 현재 광고 시간에 대한 사용자 지정 헤더가 추가됩니다.

일반적인 사용 사례

- ADS 요청 URL을 다시 작성하여 A/B 테스트를 위해 서로 다른 광고 서버 간에 트래픽을 라우팅합니다.
- ADS 요청 URL 또는 헤더에 보강 데이터(대상 세그먼트, 자격 증명 토큰)를 추가합니다.
- SCTE-35 신호 데이터 또는 가용 인덱스를 기반으로 ADS 요청 파라미터를 조건부로 수정합니다.
- 콘텐츠 분류 및 브랜드 안전 신호에 대한 Elemental Inference를 쿼리하여 컨텍스트 광고 타겟팅을 활성화합니다.

실패 특성

어떤 이유로든 연결된 함수가 PRE_ADS_REQUEST 실패하면 MediaTailor는 함수의 출력을 삭제하고 연결된 함수가 없는 것처럼 진행합니다. ADS 요청은 수정 없이 원래 세션 및 요청 파라미터를 사용하여 전송됩니다.

광고 후 응답

실행 시

MediaTailor는 VAST 래퍼 리디렉션의 확인을 포함하여 ADS로부터 응답을 수신한 POST_ADS_RESPONSE 후에 매핑된 함수를 실행하고 구문 분석합니다. 이 함수는 MediaTailor가 삽입할 광고를 선택하기 전에 구문 분석된 광고를 표시하므로 여기서 변경한 내용은 광고 시간을 채우는 광고에 영향을 미칩니다.

후크는 모든 스트리밍 프로토콜(HLS 및 DASH) 및 모든 재생 모드(라이브, VOD 및 프리롤)에 대해 실행됩니다. 일반적으로 광고 시간당 한 번 실행됩니다. 한 번에 여러 광고 시간을 포함하는 ADS 응답(예: VMAP 응답)과 미리 가져온 광고 응답의 경우 MediaTailor가 응답을 처리하고 avail 입력 네임스페이스가 없으면 후크가 실행됩니다. \$exists(avail)를 사용하여 가용 필드에 액세스하기 전에 확인합니다.

Input

`event.*`, `adsRequest.*`, 및 `session.*adsResponse.*` - 응답이 단일 광고 시간에 적용되는 `avail.*` 경우. 사용 가능한 모든 필드는 섹션을 참조하세요 [입력 필드 참조](#).

허용된 출력 네임스페이스

네임스페이스	허용되는 유형	출력 사용 방법
<code>adsResponse.ads</code>	광고 객체 배열	이 응답에 대해 구문 분석된 광고 목록을 대체합니다. MediaTailor는 반환된 각 광고를의 원래 광고와 일치 <code>adId</code> 시키고 반환된 목록과 함께 광고 선택을 진행합니다.
<code>temp.*</code>	임의	실행기의 단계 간에 값을 전달하기 위한 임시 데이터입니다. 지속되지 않습니다.

수정된 광고 목록에 다시 씁니다 `adsResponse.ads`. 다음을 수행할 수 있습니다.

- 필터 - 원래 광고의 하위 집합을 반환합니다. 생략한 광고는 삽입 대상으로 간주되지 않으며, 해당 광고가 차지한 기간은 다른 광고가 채울 수 있게 됩니다.
- 재정렬 - 광고를 다른 순서로 반환합니다. 순서는 광고 시간을 먼저 채우는 광고에 영향을 미칩니다.
- 수정 - 광고 `trackingEvents`에서 `durationSeconds`, `adTitle`, `adSystemmediaFiles`, 또는 를 변경합니다. 반환된 광고는 변경되지 않고 원래 VAST 메타데이터를 모두 유지합니다.
- 주입 - 원래 응답에 없는 광고를 추가합니다. 새 항목에는 `adId`, 및 `durationSeconds`가 포함되어야 합니다 `mediaFiles`. 동일한 후크 호출에서 `VAST_REQUEST` 함수로 가져온 광고는 예외입니다. MediaTailor가 광고의 구문 분석된 전체 VAST 데이터를 복원하기 때문에 이러한 광고를 별도로 추가하는 `adId` 것만으로도 충분합니다. 자세한 내용은 [VAST 요청](#) 단원을 참조하십시오.

Important

필터링할 때는 항상 JSONata 배열 생성자 또는 배열 강압을 사용합니다. 와 같은 필터 표현식은 정확히 하나의 광고가 일치할 때 단일 객체를 `adsResponse.ads[adSystem !=`

'BLOCKED'] 생성합니다. 0개의 광고가 일치하면 값이 전혀 생성되지 않습니다. 이 경우 출력키는 생략되고 필터는 효과가 없습니다. 표현식을 대괄호로 묶거나를 추가합니다. []

```
{
  "Output": {
    "adsResponse.ads": "{%adsResponse.ads[adSystem != 'BLOCKED'][%]}"
  }
}
```

반환된 광고 목록에는 다음 규칙이 적용됩니다.

- 두 항목이를 공유하면 adId첫 번째 항목이 성공합니다.
- 잘못된 항목은 개별적으로 거부됩니다. 동일한 출력의 유효한 항목이 여전히 적용됩니다.
- 삽입된 항목이 ADS 응답의 광고와 adId 충돌하는 경우 원래 광고가 기본으로 사용되고 그 위에 필드 변경 사항이 적용됩니다.

이 후크에 표시되지 않는 항목

이 후크는 광고 선택 전에 실행되므로 나중에 다른 MediaTailor 기능이 기여하는 광고를 볼 수 없으며, 나중에 트랜스코딩 상태 및 채우기 결정이 발생하므로 궁극적으로 삽입할 광고를 확인할 수 없습니다. 미리 가져온 광고의 경우 MediaTailor가 미리 가져오기 응답 자체를 처리할 때 후크가 실행됩니다. 나중에 광고 시간에 이전에 미리 가져온 광고는 이 후크가 실행된 후 병합되므로 후크는 해당 광고가 표시되지 않습니다. 최종 광고 세트에 대한 작업을 수행하려면 PRE_MANIFEST_INSERTION을 사용합니다.

일반적인 사용 사례

- 경쟁 광고 시스템에서 광고를 제거하는 것과 같은 메타데이터를 기준으로 광고를 필터링하여 동일한 광고 시간(상대적 분리)에 표시되지 않도록 합니다.
- 기본 응답이 짧을 때 VAST_REQUEST 함수를 사용하여 보조 광고 서버에서 추가 광고를 가져와 광고 목록에 추가합니다.
- 선호하는 광고가 먼저 광고 시간을 채우도록 광고를 재정렬합니다.
- 삽입 전에 추적 이벤트 또는 광고 메타데이터를 수정합니다.

실패 특성

제한 시간, 표현식 오류 또는 잘못된 출력과 같은 이유로에 연결된 함수가 POST_ADS_RESPONSE 실패하면 MediaTailor는 함수의 출력을 삭제합니다. 요청은 연결된 함수가 없는 것처럼 원래 ADS 응답을 진행합니다. 직렬화된 입력이 입력 크기 제한을 초과하면 MediaTailor는 후크 호출을 완전히 건너뛰니다. 적용되는 제한은 섹션을 참조하세요 [한도](#).

사전 매니페스트 삽입

실행 시

MediaTailor는 개인 맞춤 패스당 PRE_MANIFEST_INSERTION 한 번에 매핑된 함수를 실행합니다. 이는 광고 시간 개인화 종료 시, 광고 선택, 트랜스코딩 검사 및 채우기 정책 후, MediaTailor가 매니페스트에 광고를 쓰기 직전에 발생합니다. 함수는 단일 간접 호출로 전달되는에서 새로 개인화된 모든 광고 종단을 수신하므로 교차 종단 로직을 구현할 수 있습니다.

후크는 모든 스트리밍 프로토콜(HLS 및 DASH) 및 모든 재생 모드(라이브, VOD 및 프리롤)에 대해 실행됩니다. 각 광고 시간은 정확히 한 번 개인화됩니다. MediaTailor가 이후 매니페스트 요청에서 동일한 광고 시간을 렌더링할 때 후크가 다시 실행되지 않으며, 새 광고 시간을 개인화하지 않는 요청은 후크를 호출하지 않습니다.

Input

event.* 및 session.* (와 동일한 필드POST_ADS_RESPONSE) + avails.*, 각 광고 시간이 선택한 광고 및 건너뛴 광고를 포함하여 새로 개인화된 광고 시간당 항목이 하나씩 있는 배열입니다. 사용 가능한 모든 필드는 섹션을 참조하세요 [입력 필드 참조](#).

허용된 출력 네임스페이스

네임스페이스	허용되는 유형	출력 사용 방법
avails.avails	광고 종단 객체 배열	반환된 각 광고 시간의 광고 목록을 대체합니다. MediaTailor는 각 광고 시간에 대해와 일치시켜 반환된 ads 배열에서 결정을 유지, 재정렬 및 제거합니다adId. 인식할 수 없는 항목은 주입으로 adId 처리됩니다.

네임스페이스	허용되는 유형	출력 사용 방법
temp.*	임의	실행기의 단계 간에 값을 전달하기 위한 임시 데이터입니다. 지속되지 않습니다.

삽입하려는 ads 배열과 함께 각 광고 시간을 반환합니다. 보관 중인 광고 객체를 에코하고, 제거한 객체를 생략하고, 필요에 따라 재정렬합니다. POST_ADS_RESPONSE 적용되는 것과 동일한 배열-강제 규칙: 항상 배열 생성자에 필터 표현식을 래핑하거나 빈 결과는 광고 시간을 자동으로 변경하지 않고 그대로 둡니다.

광고 삽입 - adId가 원래 광고 시간에 있지 않은 항목은 삽입입니다. 삽입된 항목에서 MediaTailor는 , adId, creativeUrl, durationSeconds, vastAdId 및 필드를 준수합니다 creativeId, trackingEvents. 주입된 크리에이티브를 지정하기 위한 필드는 creativeUrl입니다. 입력에서 읽은 광고는 크리에이티브를 로 표시합니다 mediaUrl. 크리에이티브를 지정하는 방법에는 두 가지가 있습니다.

- 명시적 **creativeUrl** - 삽입할 크리에이티브의 URL을 제공합니다. MediaTailor가 이미 크리에이티브를 트랜스코딩한 경우 주입이 즉시 성공합니다. 이미 트랜스코딩된 크리에이티브의 예로는 이전 광고 시간의 크리에이티브 또는 트랜스코딩과 관련이 없는 skippedAds[].creativeUrl 값이 reason 있습니다. 크리에이티브가 아직 트랜스코딩되지 않은 경우 MediaTailor는 대기하지 않고 현재 광고 시간에 대한 주입을 삭제하지만 시도는 크리에이티브의 트랜스코딩을 시작합니다. 트랜스코딩이 완료된 후 나중에 광고 시간에 동일한 URL을 삽입하면 ADS에서 트랜스코딩되지 않은 광고를 먼저 건너뛰고 나중에 광고 시간에 삽입하는 것과 동일한 방식으로 성공합니다.
- **vastAdId** 자동 수화 - 함수 체인에 동일한 후크 호출에 VAST_REQUEST 호출이 포함된 경우 구문 분석된 광고 vastAdId의 만 사용하여 주입하고를 생략합니다 creativeUrl. 그런 다음 MediaTailor는 일반 광고 삽입 경로와 동일한 선택 로직을 사용하여 광고의 전체 미디어 파일 목록에서 최상의 미디어 파일을 선택합니다. 또한 향후 광고 시간에 사용할 수 있도록 트랜스코딩을 위한 새 스트리밍 크리에이티브를 등록합니다. 둘 다 제공하면 creativeUrl 성공합니다. 일치하는 항목이 없으면 MediaTailor는 주입을 삭제합니다. 자세한 내용은 [구문 분석된 광고가 후크 출력과 통합되는 방법](#) 단원을 참조하십시오.

다음 규칙이 적용됩니다.

- 모든 광고 시간에 걸쳐 호출당 최대 10개의 광고를 삽입할 수 있습니다. MediaTailor는 한도를 초과하여 주입을 삭제합니다.

- 광고 중단이 변경된 경우 MediaTailor는 정상 기간 허용 오차로 채우기 정책을 다시 실행합니다. 광고 시간을 초과하는 광고는 잘립니다.
- 가 로 mutable 설정된 광고 중단false(예: 오버레이 광고 중단, 억제된 광고 중단, 시청자가 중단 중간에 조인한 라이브 광고 중단)은 수정 사항을 무시하고 변경 없이 렌더링합니다. 이 후크는 가용성 억제를 재정의할 수 없습니다.
- 주입된 광고는 ADS의 광고와 동일한 변형 일치 규칙을 거칩니다. 변환이 스트림과 일치하지 않는 삽입된 광고는 삭제됩니다.
- 주입은 매니페스트 요청 간에 멍등성(결과를 복제하지 않고 반복 가능)입니다. MediaTailor는 크리에이티브 자격 증명에 따라 조정되므로 삽입된 광고는 복제하지 않고 유지됩니다.

일반적인 사용 사례

- VAST_REQUEST 함수로 가져온 과소 채워진 광고 시간을 감지fillRate하고 하우스 광고 또는 프로모션을 주입합니다.
- 예를 들어 크리에이티브가 한 번의 광고 시간에만 표시되도록 하는 등 간접 호출의 모든 광고 시간에 걸쳐 광고를 제거하거나 중복 제거합니다.
- 매니페스트에 기록되기 직전에 최종 광고 세트에서 특정 광고를 제거합니다.
- 를 사용하여 이전에 건너뛴 광고를 다시 삽입합니다skippedAds[].creativeUrl.

실패 특성

제한 시간, 표현식 오류 또는 잘못된 출력과 같은 이유로 연결된 함수가

PRE_MANIFEST_INSERTION 실패하면 MediaTailor는 함수의 출력을 삭제합니다. 원래 선택한 광고는 연결된 함수가 없는 것처럼 매니페스트에 기록됩니다. 직렬화된 입력이 입력 크기 제한을 초과하면 MediaTailor는 후크 호출을 완전히 건너뛵니다. 적용되는 제한은 단원을 참조하십시오[한도](#).

입력 필드 참조

다음 표에는 각 수명 주기 후크에서 사용할 수 있는 입력 필드가 나열되어 있습니다. 필드 열에서 parent[].child 표기법은 parent 배열에 있는 각 요소의 필드를 나타냅니다.

Note

POST_ADS_RESPONSE 및 PRE_MANIFEST_INSERTION 후크는 camelCase(예: session.clientIp)의 세션 필드를 노출하는 반면, 처음 두 후크는 snake_case(예:)를 사용

합니다. `session.client_ip`. 플레이어 파라미터는 처음 두 후크의 `player_params` 네임스페이스와 다른 두 후크의 `session.playerParams` 객체를 통해 사용할 수 있습니다.

Field	유형	PRE_SESSION_초기화	PRE_ADS_REQUEST	POST_ADS_RESPONSE	PRE_MANIFEST_INSERTION
<code>session.id</code>	Long	✓	✓	✓	✓
<code>session.uid</code>	문자열	✓	✓	✓	✓
<code>session.client_ip</code>	문자열	✓	✓	✗	✗
<code>session.clientIp</code>	문자열	✗	✗	✓	✓
<code>session.user_agent</code>	문자열	✓	✓	✗	✗
<code>session.userAgent</code>	문자열	✗	✗	✓	✓
<code>session.referrer</code> [*]	문자열	✓	✓	✗	✗
<code>session.avail_duration_secs</code>	Long	✗	✓	✗	✗
<code>session.avail_duration_ms</code>	Long	✗	✓	✗	✗

Field	유형	PRE_SESSI ON_초기화	PRE_ADS_R EQUEST	POST_ADS_ RESPONSE	PRE_MANIF EST_INSERT ION
session.s treamingP rotocol	문자열	X	X	✓	✓
session.p layerPara ms	객체	X	X	✓	✓
player_pa rams.*	문자열	✓	✓	X	X
event.id	문자열	✓	✓	✓	✓
event.hoo k	문자열	✓	✓	✓	✓
event.tim estamp	문자열	✓	✓	✓	✓
avail.ind ex	정수	X	✓	X	X
avail.ran dom	Long	X	✓	X	X
avail.sou rce_conte nt_time_e poch_ms	Long	X	✓	X	X
avail.ava ilId [†]	문자열	X	X	✓	X

Field	유형	PRE_SESSI ON_초기화	PRE_ADS_R EQUEST	POST_ADS_ RESPONSE	PRE_MANIF EST_INSERT ION
avail.dur ationSeco nds [†]	숫자	X	X	✓	X
avail.sta rtTime [†]	문자열	X	X	✓	X
scte.even t_id	정수	X	✓	X	X
scte.avai l_num	정수	X	✓	X	X
scte.segm entation_ event_id	정수	X	✓	X	X
scte.segm entation_ type_id	정수	X	✓	X	X
scte.segm entation_ upid	문자열	X	✓	X	X
scte.segm entation_ upid.asse tId	문자열	X	✓	X	X
scte.segm entation_ upid.cueD ata.key	문자열	X	✓	X	X

Field	유형	PRE_SESSI ON_초기화	PRE_ADS_R EQUEST	POST_ADS_ RESPONSE	PRE_MANIF EST_INSERT ION
scte.segm entation_ upid.cueD ata.value	문자열	X	✓	X	X
scte.uniq ue_progra m_id	정수	X	✓	X	X
scte.arch ive_allow ed_flag	부울	X	✓	X	X
scte.deliv ery_not_ restricte d_flag	부울	X	✓	X	X
scte.devi ce_restri ctions	정수	X	✓	X	X
scte.no_r egional_b lackout_f lag	부울	X	✓	X	X
scte.segm ent_num	정수	X	✓	X	X
scte.segm ents_expe cted	정수	X	✓	X	X

Field	유형	PRE_SESSI ON_초기화	PRE_ADS_R EQUEST	POST_ADS_ RESPONSE	PRE_MANIF EST_INSERT ION
scte.sub_ segment_n um	정수	X	✓	X	X
scte.sub_ segments_ expected	정수	X	✓	X	X
scte.avai ls_expect ed	Long	X	✓	X	X
asset.*	문자열	X	✓	X	X
inference .enriched	부울	X	✓	X	X
inference .feedId	문자열	X	✓	X	X
inference .dataEndp oint	문자열	X	✓	X	X
inference .pts	Long	X	✓	X	X
inference .timescal e	Long	X	✓	X	X
inference .region	문자열	X	✓	X	X

Field	유형	PRE_SESSI ON_초기화	PRE_ADS_R EQUEST	POST_ADS_ RESPONSE	PRE_MANIF EST_INSERT ION
inference. previous BreakEndP ts	Long	X	✓	X	X
inference. parseErr or	부울	X	✓	X	X
adsReques t.url	문자열	X	✓	✓	X
adsReques t.method	문자열	X	✓	✓	X
adsReques t.headers .<key>	문자열	X	✓	✓	X
adsReques t.body	문자열	X	✓	X	X
adsRespon se.respon seType	문자열	X	X	✓	X
adsRespon se.ads	배열	X	X	✓	X
adsRespon se.ads[]. adId [‡]	문자열	X	X	✓	X

Field	유형	PRE_SESSI ON_초기화	PRE_ADS_R EQUEST	POST_ADS_ RESPONSE	PRE_MANIF EST_INSERT ION
adsRespon se.ads[]. durationS econds	숫자	X	X	✓	X
adsRespon se.ads[]. adSystem	문자열	X	X	✓	X
adsRespon se.ads[]. adTitle	문자열	X	X	✓	X
adsRespon se.ads[]. creativeI d	문자열	X	X	✓	X
adsRespon se.ads[]. sequence	정수	X	X	✓	X
adsRespon se.ads[]. mediaFile s	배열	X	X	✓	X
adsRespon se.ads[]. mediaFile s[].url	문자열	X	X	✓	X

Field	유형	PRE_SESSI ON_초기화	PRE_ADS_R EQUEST	POST_ADS_ RESPONSE	PRE_MANIF EST_INSERT ION
adsRespon se.ads[]. mediaFile s[].mimeT ype	문자열	X	X	✓	X
adsRespon se.ads[]. mediaFile s[].width	정수	X	X	✓	X
adsRespon se.ads[]. mediaFile s[].heigh t	정수	X	X	✓	X
adsRespon se.ads[]. mediaFile s[].bitra te	정수	X	X	✓	X
adsRespon se.ads[]. trackingE vents.<ev entType>	문자열 배열	X	X	✓	X
avails.av ails	배열	X	X	X	✓

Field	유형	PRE_SESSI ON_초기화	PRE_ADS_R EQUEST	POST_ADS_ RESPONSE	PRE_MANIF EST_INSERT ION
avails.av ails[].av ailId	문자열	X	X	X	✓
avails.av ails[].du rationSec onds	숫자	X	X	X	✓
avails.av ails[].st artTime	숫자	X	X	X	✓
avails.av ails[].me diaProtoc ol	문자열(HLS DASH)	X	X	X	✓
avails.av ails[].st reamingMo de	문자열(LIVE VOD)	X	X	X	✓
avails.av ails[].fi llDuratio nSeconds	숫자	X	X	X	✓
avails.av ails[].fi llRate	숫자(0~1)	X	X	X	✓

Field	유형	PRE_SESSI ON_초기화	PRE_ADS_R EQUEST	POST_ADS_ RESPONSE	PRE_MANIF EST_INSERT ION
avails.av ails[].mu table	부울	X	X	X	✓
avails.av ails[].ad s	배열	X	X	X	✓
avails.av ails[].ad s[].adId ^s	문자열	X	X	X	✓
avails.av ails[].ad s[].vastA dId	문자열	X	X	X	✓
avails.av ails[].ad s[].creat iveId	문자열	X	X	X	✓
avails.av ails[].ad s[].inser tionUuid	문자열	X	X	X	✓
avails.av ails[].ad s[].seque nceInAvai l	정수	X	X	X	✓

Field	유형	PRE_SESSI ON_초기화	PRE_ADS_R EQUEST	POST_ADS_ RESPONSE	PRE_MANIF EST_INSERT ION
avails.av ails[].ad s[].durat ionSecond s	숫자	X	X	X	✓
avails.av ails[].ad s[].adSys tem ¶	문자열	X	X	X	✓
avails.av ails[].ad s[].adTit le ¶	문자열	X	X	X	✓
avails.av ails[].ad s[].media Url	문자열	X	X	X	✓
avails.av ails[].ad s[].track ingEvents .<eventTy pe>	문자열 배열	X	X	X	✓
avails.av ails[].ad s[].media Files	배열	X	X	X	✓

Field	유형	PRE_SESSI ON_초기화	PRE_ADS_R EQUEST	POST_ADS_ RESPONSE	PRE_MANIF EST_INSERT ION
avails.av ails[].sk ippedAds	배열	X	X	X	✓
avails.av ails[].sk ippedAds[].adId	문자열	X	X	X	✓
avails.av ails[].sk ippedAds[].vastAdI d	문자열	X	X	X	✓
avails.av ails[].sk ippedAds[].duratio nSeconds	숫자	X	X	X	✓
avails.av ails[].sk ippedAds[].creativ eUrl	문자열	X	X	X	✓
avails.av ails[].sk ippedAds[].reason	문자열	X	X	X	✓

* `session.referer`는 세션 초기화 요청에 참조자 헤더가 포함된 경우에만 표시됩니다.
`$exists(session.referer)`를 사용하여 액세스하기 전에 확인합니다.

† `avail` 네임스페이스는 ADS 응답이 단일 광고 시간에 적용되는 경우에만 존재합니다. 다중 중단 (VMAP) 응답 또는 미리 가져온 응답에는 존재하지 않습니다. `$exists(avail)`를 사용하여 액세스하기 전에 확인합니다.

‡ 여기서 `POST_ADS_RESPONSEadId`는 VAST <Ad> 요소의 `id` 속성입니다.

" 여기서 `PRE_MANIFEST_INSERTIONadId`는 내부 배치 식별자입니다. 광고 시간 내에 광고에 대해 안정적이지만 VAST 광고 ID가 아니며 광고 시간 또는 후크 간에 비교할 수 없습니다. 광고 브레이크 또는 후크에 걸쳐 있는 자격 증명 확인의 경우 `vastAdId` (VAST <Ad> 요소의 `id` 속성) 또는 `creativeId`를 사용합니다.

¶ DASH VOD 스트림에서 `adTitle` 및 `adSystem`는 이 후크 `null`에 있습니다. 여기에서 조건자에 `vastAdId` 또는 `creativeId`를 사용합니다.

함수 유형 및 구성

AWS Elemental MediaTailor 는 서로 다른 작업 범주에 맞게 설계된 6가지 함수 유형을 제공합니다. 함수를 생성할 때 유형을 선택하면 해당 유형에 따라 함수가 런타임에 수행할 수 있는 작업이 결정됩니다. 이 페이지에서는 각 유형의 실행 방법, 사용 시기, 함수를 다단계 파이프라인으로 구성하는 방법을 설명합니다.

함수 유형 개요

유형	카테고리	용도
CUSTOM_OUTPUT	데이터 변환	현재 세션 상태를 기준으로 표현식을 평가하고 출력을 생성합니다. 외부 호출이 없습니다.
HTTP_REQUEST	외부 통합	외부 서비스에 대한 HTTP 호출을 수행한 다음 응답을 참조할 수 있는 출력 표현식을 평가합니다.
VAST_REQUEST	외부 통합	VAST 엔드포인트를 호출하고 응답을 VAST(래퍼 해상도 포

유형	카테고리	용도
		함)로 구문 분석한 다음 구문 분석된 광고를 참조할 수 있는 출력 표현식을 평가합니다.
SEQUENTIAL_EXECUTOR	오케스트레이션	함수 시퀀스를 순서대로 실행하여 임시 데이터를 통해 단계 간에 데이터를 전달합니다.
CONCURRENT_EXECUTOR	오케스트레이션	함수 세트를 최대 동시성까지 병렬로 실행한 다음 출력을 결합합니다.
AWS_SERVICE_REQUEST	AWS 서비스 통합	지원되는 AWS 서비스 API에 대해 인증된 요청을 수행한 다음 응답을 참조할 수 있는 출력 표현식을 평가합니다.

각 유형은 고유한 역할을 합니다. 함수를 생성할 때 유형을 선택합니다. 유형에 따라 함수가 런타임에 수행할 수 있는 작업이 결정됩니다.

사용자 지정 출력

사용해야 하는 경우

함수 로직이 순전히 계산되는 CUSTOM_OUTPUT 경우에 사용합니다. 일반적인 사용 사례에는 사용자 에이전트 문자열을 기반으로 디바이스 분류, A/B 테스트 로직을 사용하여 광고 결정 서버 URLs 선택, 광고 결정 서버에 도달하기 전에 플레이어 파라미터 재포밍이 포함됩니다.

CUSTOM_OUTPUT 함수는 네트워크 호출 없이 메모리의 표현식을 평가하기 때문에 가장 빠른 유형입니다. 외부 서비스에 문의하지 않고 목표를 달성할 수 있을 때마다 유형을 선택합니다.

구성 필드

CUSTOM_OUTPUT 함수에는 두 개의 필드가 있습니다.

- 런타임 - 표현식 언어입니다. 이를 로 설정합니다JSONATA.
- 출력 - 키-값 페어의 맵입니다. 각 키는 출력 바인딩 이름(예: `player_params.device_type`)이고, 각 값은 MediaTailor가 런타임 시 평가하는 표현식입니다.

Note

문자열 필드에서 JSONata 표현식을 래핑{%...%}하는 데 사용합니다. 이러한 구분 기호가 없는 값은 정적 문자열로 처리됩니다. 예를 들어는 표현식을 "{%session.id%}" 평가하는 반면 "session.id"는 리터럴 텍스트입니다.

런타임 동작

MediaTailor는 CUSTOM_OUTPUT 함수를 실행할 때 현재 세션 상태를 기준으로 출력 블록의 모든 표현식을 평가합니다. 모든 표현식은 동일한 입력 스냅샷을 볼 수 있습니다. 평가된 결과는 함수의 출력이 됩니다.

표현식이 평가되지 않으면 MediaTailor는 전체 함수의 출력을 삭제하고 연결된 함수가 없는 것처럼 진행합니다. 함수는 결과를 부분적으로 커밋하지 않습니다.

예: A/B 트래픽 분할

다음 함수는 각 광고 시간을 두 광고 결정 서버 URLs. 수명 PRE_ADS_REQUEST 주기 후크용으로 설계되었습니다.

```
{
  "FunctionId": "abTestAdsUrl",
  "FunctionType": "CUSTOM_OUTPUT",
  "CustomOutputConfiguration": {
    "Runtime": "JSONATA",
    "Output": {
      "adsRequest.url": "{%$random() < 0.5 ? 'https://ads.example.com/v1' : 'https://ads.example.com/v2'%}"
    }
  }
}
```

이 예제의 전체 연습은 단원을 참조하십시오 [함수 예제](#).

HTTP 요청**사용해야 하는 경우**

함수가 외부 서비스를 호출해야 할 HTTP_REQUEST 때 사용합니다. 일반적인 사용 사례에는 해결 공급자로부터 자격 증명 데이터 가져오기, 데이터 관리 플랫폼에서 대상 세그먼트 검색, 로깅 엔드포인트로 세션 정보 전송이 포함됩니다.

구성 필드

HTTP_REQUEST 함수에는 다음 필드가 있습니다.

- 런타임 - 표현식 언어입니다. 이를 로 설정합니다JSONATA.
- MethodType - HTTP 메서드입니다. 지원되는 값은 GET 및 POST입니다.
- URL - 요청을 보낼 URL입니다. 정적 URL 또는 URL을 동적으로 빌드하는 JSONata 표현식을 사용할 수 있습니다.
- 헤더 - 요청에 포함할 HTTP 헤더로, 헤더 이름 및 값 페어로 지정됩니다. 동적 헤더 값에 {%...%} 표현식 구문을 사용합니다. 정적 값은 문자열로 직접 지정할 수 있습니다.
- 본문 - 전송할 요청 본문입니다. POST 요청에 사용됩니다. JSONata 표현식을 사용하여 동적으로 본문을 빌드할 수 있습니다.
- RequestTimeoutMilliseconds(필수) - 응답을 기다리는 시간입니다.
- 출력 - HTTP 호출이 완료된 후 생성할 값을 정의합니다. 각 항목은 response 객체를 참조할 수 있는 표현식에 출력 키(예: player_params.envelope_id)를 매핑합니다.

이러한 필드에 적용되는 크기 제한 및 제한은 섹션을 참조하세요[한도](#).

요청 처리 방법

MediaTailor는 다음 두 단계로 HTTP_REQUEST 함수를 처리합니다.

1. 요청 빌드 - MediaTailor는 현재 세션 상태를 기준으로 UrlHeaders, 및 Body 표현식을 평가합니다. 이러한 평가된 값은 아웃바운드 HTTP 요청을 형성합니다.
2. 응답 처리 - HTTP 호출이 완료되면 MediaTailor는 출력 블록의 표현식을 평가합니다. 이러한 표현식은 원래 세션 상태와 호출에서 반환된 response 객체를 모두 참조할 수 있습니다.

응답 필드

HTTP 호출이 완료되면 출력 표현식에서 다음 필드를 참조할 수 있습니다.

Field	유형	설명
response.body	객체 또는 배열	JSON으로 구문 분석된 응답 본문입니다. 본문이 20,000자를 초과하거나 유효한 JSON이

Field	유형	설명
		아닌 null 경우 로 설정합니다.
<code>response.statusCode</code>	Integer	외부 서비스에서 반환한 HTTP 상태 코드입니다. 네트워크 장애 null 시를 로 설정합니다.
<code>response.text</code>	문자열	원시 응답 본문은 문자열로, 20,000자로 잘립니다. 네트워크 장애 "Internal Error" 시를 로 설정합니다.

⚠ Important

`response.body` 필드는 응답이 유효한 JSON인 경우에도 응답이 20,000자를 초과할 null 때 입니다.

ℹ Note

응답 객체는 HTTP_REQUEST 함수의 출력 블록에서만 사용할 수 있습니다. URL, 헤더 또는 본문 필드의 응답 필드는 참조할 수 없습니다. 에서 SEQUENTIAL_EXECUTOR 각 HTTP_REQUEST 함수는 자체 응답에만 액세스할 수 있습니다.

값이 이면 데이터를 사용할 수 없음을 null 의미합니다. 이는 HTTP 호출이 실패하거나(네트워크 오류 또는 제한 시간) 응답 본문이 20,000자를 초과하거나 유효한 JSON이 아닐 때 발생합니다.

네트워크 장애 동작

네트워크 오류 또는 제한 시간으로 인해 HTTP 호출이 실패 `response.statusCode` 하고 `response.body` 가 로 설정 null 되고 `response.text` 가 로 설정된 경우 "Internal Error". 출력 표현식은 여전히 실행되므로 응답 데이터를 사용하기 `response.statusCode` 전에 항상 확인하세요.

i Tip

조건식을 사용하여 실패를 정상적으로 처리합니다. `{%response.statusCode = 200 ? response.body.value : 'default'%}`

예: 자격 증명 데이터 가져오기

다음 함수는 세션 시작 시 자격 증명 확인 API를 호출하고 결과를 플레이어 파라미터에 저장합니다. 수명 PRE_SESSION_INITIALIZATION 주기 후크용으로 설계되었습니다.

```
{
  "FunctionId": "fetchIdentityEnvelope",
  "FunctionType": "HTTP_REQUEST",
  "HttpRequestConfiguration": {
    "Runtime": "JSONATA",
    "MethodType": "GET",
    "Url": "{%'https://identity.example.com/v1/resolve?ip=' &
$encodeURIComponent(session.client_ip)%}",
    "Headers": {
      "Authorization": "{%'Bearer my_api_token'%}",
      "Accept": "application/json"
    },
    "RequestTimeoutMilliseconds": 2000,
    "Output": {
      "player_params.identity_envelope": "{%response.statusCode = 200 ?
response.body.envelope : ''%}"
    }
  }
}
```

유사한 예제에 대한 전체 연습은 단원을 참조하십시오 [함수 예제](#).

VAST 요청

사용해야 하는 경우

함수가 VAST(Video Ad Serving Template) 엔드포인트에서 광고를 가져와야 할 VAST_REQUEST 때 사용됩니다. MediaTailor는 요청을 보내고, 응답을 래퍼 리디렉션이 해결된 VAST로 구문 분석하고, 구문 분석된 광고를 출력 표현식에 사용할 수 있도록 합니다. 일반적인 사용 사례에는 짧은 기본 응답을 보완하기 위해 보조 광고 서버에서 광고를 가져오는 것과 과소 채워진 광고 시간에 삽입할 하우스 광고 또는 프로모션을 검색하는 것이 포함됩니다.

VAST_REQUEST 함수는 자체적으로 또는 실행기 내의 단계로 POST_ADS_RESPONSE 및 PRE_MANIFEST_INSERTION 수명 주기 후크에서 지원됩니다. VAST_REQUEST 함수에 의해 구문 분석된 광고는 완전한 충실도로 해당 후크의 출력과 통합됩니다. 자세한 내용은 [the section called “구문 분석된 광고가 후크 출력과 통합되는 방법”](#)을 참조하세요.

구성 필드

VAST_REQUEST 함수에는 다음 필드가 있습니다.

- 런타임 - 표현식 언어입니다. 이를 로 설정합니다JSONATA.
- MethodType - HTTP 메서드입니다. 지원되는 값은 GET (기본값) 및 입니다POST.
- URL - 요청을 보낼 VAST 엔드포인트입니다. 정적 URL 또는 URL을 동적으로 빌드하는 JSONata 표현식을 사용할 수 있습니다. 정적 URL은 https 스키마를 사용해야 합니다.
- 헤더 - 요청에 포함할 HTTP 헤더로, 헤더 이름 및 값 페어로 지정됩니다. 동적 헤더 값에 {%...%} 표현식 구문을 사용합니다.
- 본문 - 전송할 요청 본문입니다. 예를 들어 OpenRTB 입찰 POST 요청을 보내는 요청과 함께 사용됩니다. JSONata 표현식을 사용하여 동적으로 본문을 빌드할 수 있습니다.
- RequestTimeoutMilliseconds(필수) - 응답을 기다리는 시간입니다.
- 출력 - 호출이 완료된 후 생성할 값을 정의합니다. 각 항목은를 포함하여 response 객체를 참조할 수 있는 표현식에 출력 키를 매핑합니다response.parsedAds.

HTTP_REQUEST 함수에 적용되는 HTTP 요청 제한도 여기에 적용됩니다. 자세한 내용은 [한도](#)을 참조하세요.

요청 처리 방법

1. 요청 빌드 - MediaTailor는 현재 세션 상태를 기준으로 Url, Headers 및 Body 표현식을 평가한 다음 요청을 전송합니다. POST 요청은 application/json 콘텐츠 유형과 함께 전송됩니다.
2. 응답 구문 분석 - MediaTailor는 응답 본문을 VAST로 구문 분석합니다. 광고가 VAST 래퍼인 경우 MediaTailor는 리디렉션 체인을 따라 최종 인라인 광고로 확인합니다. 래퍼 리디렉션 요청은 VAST 사양GET에 따라 항상와 함께 전송됩니다. 엔드포인트가 VAST 자체 대신 VAST URL이 포함된 문서를 반환하는 경우(일부 입찰자처럼) URL을 검색하는 HTTP_REQUEST 단계를 연결한 다음 가져오는 VAST_REQUEST 단계를 연결합니다.
3. 구문 분석된 광고 처리 - MediaTailor는 출력 블록의 표현식을 평가합니다. 이러한 표현식은 원래 세션 상태와 response 객체를 참조할 수 있습니다.

응답 필드

호출이 완료되면 출력 표현식에서 다음 필드를 참조할 수 있습니다.

Field	유형	설명
<code>response.parsedAds</code>	배열	래퍼 리디렉션이 해결된 구문 분석된 광고입니다. 각 광고의 모양은 <code>POST_ADS_RESPONSE</code> 후크의 <code>adsResponse.ads</code> 항목과 동일합니다. <code>trackingEvents.adId</code> , <code>durationSeconds</code> , <code>adSystem</code> , <code>adTitle</code> , <code>creativeId</code> , <code>sequence</code> , 및 <code>mediaFiles</code> . 호출이 실패하거나 응답이 유효한 VAST가 아닌 경우 빈 배열로 설정합니다.
<code>response.statusCode</code>	Integer	엔드포인트에서 반환한 HTTP 상태 코드입니다. 네트워크 장애 <code>null</code> 시를 로 설정합니다.

구문 분석된 광고는와 동일한 형태를 사용하므로 필드 참조를 다시 작성하지 않고도 두 표면 간에 표현식을 이동할 `adsResponse.ads` 수 있습니다.

구문 분석된 광고가 후크 출력과 통합되는 방법

VAST_REQUEST 함수가 구문 분석하는 모든 광고는 MediaTailor에 의해 내부적으로 유지되며, 후크 간접 호출 기간 동안 광고의 VAST 광고 ID로 입력됩니다. 이렇게 하면 파싱된 광고가 후크 출력에서 일급 처리를 제공합니다.

- 예를 들어 `adsResponse.ads` 통해 구문 분석된 광고를 `POST_ADS_RESPONSE` 추가하면 전체 미디어 파일 목록 및 추적 이벤트를 포함하여 광고의 전체 구문 분석된 VAST 데이터가 `adId` 복원됩니다.
- 예를 들어만 광고를 `PRE_MANIFEST_INSERTION` 삽입 `vastAdId` 하면 MediaTailor가 최상의 미디어 파일을 선택하고 트랜스코딩 등록을 자동으로 처리할 수 있습니다.

⚠ Important

보관된 광고는 후크 호출 간에 이월되지 않습니다. 에서 이루어진 VAST_REQUEST 호출 POST_ADS_RESPONSE는 나중에 PRE_MANIFEST_INSERTION 호출할 수 없습니다. 결과를 사용하는 동일한 후크의 함수 체인 VAST_REQUEST에서 호출합니다.

실패 특성

네트워크 오류 또는 제한 시간으로 인해 호출이 실패하거나, 엔드포인트가 HTTP 오류를 반환하거나, 응답을 VAST로 구문 분석할 수 없는 경우 `response.parsedAds`는 빈 배열이며 함수 체인은 계속됩니다. HTTP 실패와 실제로 광고가 포함되지 않은 VAST 응답을 구분 `response.statusCode`하려면 를 확인합니다.

💡 Tip

`RequestTimeoutMilliseconds` 꼭 조이세요. VAST 래퍼 확인에는 여러 네트워크 왕복이 포함될 수 있으며, 모든 네트워크 왕복은 후크의 제한 시간 및 공유 후크 예산에 포함됩니다.

예: 보조 광고 서버에서 광고 가져오기

다음 함수는 백업 광고 서버에서 광고를 가져오고 구문 분석된 광고를 임시 데이터에 저장합니다. 동일한 시퀀스의 나중 CUSTOM_OUTPUT 단계에서 추가할 수 있습니다 `adsResponse.ads`. 수명 POST_ADS_RESPONSE 주기 후크용으로 설계되었습니다.

```
{
  "FunctionId": "fetchBackupAds",
  "FunctionType": "VAST_REQUEST",
  "VastRequestConfiguration": {
    "Runtime": "JSONATA",
    "MethodType": "GET",
    "Url": "{% 'https://backup-ads.example.com/vast?sid=' & session.uuid %}",
    "Headers": {
      "Accept": "application/xml"
    },
    "RequestTimeoutMilliseconds": 1000,
    "Output": {
      "temp.backupAds": "{% response.parsedAds %}"
    }
  }
}
```



```
}
}
```

대신 OpenRTB 입찰 요청을 보내려면 MethodType로 설정하고에서 입찰 요청을 POST 제공합니다Body. 입찰자는 응답 본문에 VAST를 반환해야 합니다.

유사한 예제에 대한 전체 연습은 단원을 참조하십시오[함수 예제](#).

순차 실행기

사용해야 하는 경우

는 순서대로 한 번에 하나씩 일련의 함수를 SEQUENTIAL_EXECUTOR 실행합니다. 각 단계에서는 단계 결과를 사용할 수 있습니다.

로직에 서로의 결과에 따라 여러 단계가 필요한 SEQUENTIAL_EXECUTOR 경우를 사용합니다. 일반적인 사용 사례에는 자격 증명 데이터를 가져온 다음 이를 사용하여 대상 세그먼트를 검색하고, 분류 단계를 실행한 다음 결과에 따라 다양한 외부 서비스를 조건부로 호출하고, 여러 데이터 소스에서 복잡한 광고 요청 URLs 구축하는 것이 포함됩니다.

구성 필드

SEQUENTIAL_EXECUTOR 함수에는 다음 필드가 있습니다.

- 런타임 - 표현식 언어입니다. 이를 로 설정합니다JSONATA.
- FunctionList - 1~10단계로 구성된 정렬된 목록입니다. 각 단계는 실행할 함수FunctionId의를 지정합니다. 선택적으로 RunCondition 표현식을 추가하여 단계가 실행 또는 건너뛰는지 여부를 제어할 수 있습니다.
- 출력(필수) - 모든 단계가 완료된 후 생성할 값을 정의합니다. 각 항목은 출력 키(예: player_params.envelope)를 시퀀스의 모든 단계에서 생성된 데이터를 참조할 수 있는 표현식에 매핑합니다.
- TimeoutMilliseconds(필수) - 전체 시퀀스가 완료되는 최대 시간입니다. 시퀀스가이 제한 시간을 초과하면 MediaTailor는 시퀀스의 모든 출력을 삭제합니다.

정렬된 실행 및 데이터 흐름

MediaTailor는 처음부터 끝까지 시퀀스의 각 단계를 실행합니다. 각 단계가 완료되면 생성되는 값이 실행 중인 결과 집합에 병합됩니다. 후속 단계에서는 원래 세션 데이터와 이전 단계에서 생성된 모든 값에 액세스할 수 있습니다.

임시 데이터는 단계 간에 데이터를 전달하는 기본 메커니즘입니다. 함수가 temp.* 키에 쓰면 다음 단계에서 해당 값을 읽을 수 있습니다. 이전 단계에서 작성한 플레이어 파라미터 및 광고 요청 필드도 이후 단계에서 볼 수 있습니다.

Note

임시 데이터는 객체 및 배열을 포함한 모든 데이터 유형을 허용합니다. 플레이어 파라미터 및 광고 요청 필드는 문자열, 숫자, 부울 및 null만 허용합니다).

단계별 실행 조건

시퀀스의 각 단계에는 선택적 RunCondition 필드가 있습니다. 이 필드에는 true 또는 false를 반환하는 표현식이 포함되어 있습니다. MediaTailor는 해당 단계를 실행하기 직전에 RunCondition 표현식을 평가합니다.

RunCondition 표현식이 false로 평가되면 MediaTailor는 단계를 완전히 건너뛰고 다음 단계로 이동합니다. RunCondition 필드를 생략하면 단계가 항상 실행됩니다.

```
{ "FunctionId": "retryFetch", "RunCondition": "{%temp.statusCode = 500%}" }
```

이 메커니즘을 사용하면 조건부 파이프라인을 구축할 수 있습니다. 예를 들어 1단계에서 자격 증명 가져오기를 실행한 다음 1단계에서 유효한 자격 증명을 반환한 경우에만 2단계에서 세그먼트 조회를 조건부로 실행할 수 있습니다.

출력 블록의 작동 방식

의 출력 블록은 모든 단계가 완료된 후 시퀀스가 생성하는 내용을 SEQUENTIAL_EXECUTOR 제어합니다. MediaTailor는 최종 누적 상태를 기준으로 출력 블록의 표현식을 평가하고 해당 출력만 저장합니다. 순차 출력 블록에서 참조되지 않는 이전 단계에서 생성된 모든 출력은 삭제됩니다.

Tip

출력 블록을 사용하여 누적된 결과를 저장하기 전에 필터링, 이름 바꾸기 또는 변환합니다. 단계는 temp.* 키를 통해 중간 결과를 공유하고 출력 블록이 커밋된 항목을 결정하도록 할 수 있습니다.

제한 시간 구성

TimeoutMilliseconds 필드는 전체 시퀀스의 기한을 설정합니다. 이 제한 시간은 함수에서 수행한 HTTP 호출을 포함하여 모든 단계를 다룹니다. 시퀀스가 제한 시간을 초과하면 MediaTailor는 시퀀스의 모든 출력을 삭제하고 연결된 함수가 없는 것처럼 진행합니다.

개별 HTTP_REQUEST 함수는 여전히 자체 RequestTimeoutMilliseconds 설정을 준수합니다. 시퀀스 제한 시간은 총 실행 시간을 제한하는 외부 경계 역할을 합니다.

예: HTTP 실패 시 재시도

이 예제에서는 자격 증명 API를 호출하고 첫 번째 호출에서 서버 오류가 반환되면 자동으로 재시도합니다. SEQUENTIAL_EXECUTOR에서 오케스트레이션한 두 개의 HTTP_REQUEST 함수를 사용합니다.

1단계 - 기본 가져오기(**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%}"
    }
  }
}
```

2단계 - 실패 시 재시도(**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%}"
      }
    }
  }
}

```

시퀀스(**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%}"
    }
  }
}

```

작동 방식:

1. `fetchIdentity`는 자격 증명 API를 호출하고 상태 코드와 봉투에 씁니다 `temp.*`.
2. 상태 코드가 500 이상이면 2단계 `RunCondition`의가 로 평가 `true`되고 `retryIdentity` 실행됩니다. 재시도 응답 `temp.envelope`으로 `temp.statusCode` 맞을 덮어씁니다.
3. 첫 번째 호출이 성공하면 2단계를 건너뜁니다.

4. 시퀀스 출력 블록은 `temp.envelope`에 씁니다 `player_params.envelope`.

동시 실행기

사용해야 하는 경우

는 최대 동시성까지 하위 함수 세트를 병렬로 `CONCURRENT_EXECUTOR` 실행한 다음 모든 하위 함수가 완료된 후 출력을 결합합니다.

로직에 서로의 결과에 의존하지 않는 여러 독립 작업이 필요한 `CONCURRENT_EXECUTOR` 경우에 사용합니다. 일반적인 사용 사례에는 다양한 보강 서비스에 대한 병렬 HTTP 호출, 독립적인 데이터 조회 동시 실행, 네트워크 요청 중첩으로 총 지연 시간 단축이 포함됩니다.

구성 필드

`CONCURRENT_EXECUTOR` 함수에는 다음 필드가 있습니다.

- 런타임 - 표현식 언어입니다. 이를 로 설정합니다 `JSONATA`.
- `FunctionList` - 병렬로 실행할 1~10개의 하위 항목 목록입니다. 각 항목은 실행할 함수 `FunctionId`의를 지정합니다. 선택적으로 `RunCondition` 표현식을 추가하여 하위 표현식의 실행 또는 건너뛰기를 제어할 수 있습니다. `Alias`를 추가하여 실행기 내에서 하위 항목에 고유한 이름을 지정할 수도 있습니다.
- 출력(필수) - 모든 하위 항목이 완료된 후 생성할 값을 정의합니다. 각 항목은 출력 키(예: `player_params.device_type`)를 실행기의 모든 하위 항목에서 생성된 데이터를 참조할 수 있는 표현식에 매핑합니다.
- `MaxConcurrency` - 동시에 실행할 수 있는 최대 하위 항목 수입니다. 유효한 범위: 1-2. 기본값: 2.
- `TimeoutMilliseconds`(필수) - 전체 실행기가 완료되는 최대 시간입니다. 실행기가이 제한 시간을 초과하면 `MediaTailor`는 실행기의 모든 출력을 삭제합니다. 기본값: 2000.

병렬 실행 및 데이터 흐름

`MediaTailor`는 하위 항목을 동시에 시작합니다. 함수 목록에 보다 많은 항목이 있는 경우 `MaxConcurrency` `MediaTailor`는 실행 중인 하위 항목이 완료되면 추가 하위 항목을 시작합니다. 동시에 실행되는 하위 항목 수가 동시성 한도 이하로 유지됩니다.

각 하위 항목의 결과는 네임스페이스, 즉 명시적 `Alias`에 저장되거나 별칭이 설정되지 않은 `FunctionId` 경우 하위 항목의에 저장됩니다. 모든 하위 항목이 완료되면 실행기의 `Output` 표현식은 경로를 사용하여 하위 항목의 결과를 참조할 수 있습니다 `function.<namespace>.<path>`.

Note

네임스페이스는 실행기의 모든 하위 항목에서 고유해야 합니다. MediaTailor는 작성 시이 요구 사항을 검증합니다.

하위 항목별 실행 조건

실행기의 각 하위 항목에는 선택적 RunCondition 필드가 있습니다. 이 필드에는 true 또는를 반환하는 표현식이 포함되어 있습니다false. MediaTailor는 해당 하위 항목을 시작하기 직전에 RunCondition 표현식을 평가합니다.

RunCondition 표현식이 로 평가되면 false MediaTailor는 하위 항목을 완전히 건너뛵니다. RunCondition 필드를 생략하면 하위 필드가 항상 실행됩니다.

```
{ "FunctionId": "fetchDevice", "RunCondition": "{%session.player_params.device_id != null%}" }
```

이 메커니즘을 사용하면 조건부로 하위 항목을 포함할 수 있습니다. 예를 들어 디바이스 ID가 세션에 없는 경우 디바이스 조회를 건너뛸 수 있습니다.

출력 블록 작동 방식

의 출력 블록은 모든 하위 항목이 완료된 후 실행기가 생성하는 내용을 CONCURRENT_EXECUTOR 제어합니다. 각 항목은를 사용하여 하위 결과를 참조할 수 있는 표현식에 출력 키를 매핑합니다{%function.<namespace>.<path>%}.

예를 들어 하위의 별칭이 fetchUser 이고에 쓰는 경우 temp.user실행기의 출력 표현식은 이를 로 참조할 수 있습니다{%function.fetchUser.temp.user%}.

제한 시간 구성

TimeoutMilliseconds 필드는 전체 실행기의 기한을 설정합니다. 이 제한 시간은 함수에서 수행한 HTTP 호출을 포함하여 모든 하위 항목을 포함합니다. 실행기가 제한 시간을 초과하면 MediaTailor는 실행기의 모든 출력을 삭제하고 연결된 함수가 없는 것처럼 진행합니다.

개별 HTTP_REQUEST 함수는 여전히 자체 RequestTimeoutMilliseconds 설정을 준수합니다. 실행기 제한 시간은 총 실행 시간을 제한하는 외부 경계 역할을 합니다.

예: 병렬 보강 호출

이 예제에서는 두 개의 보강 APIs 병렬로 호출하고 결과를 플레이어 파라미터로 결합합니다. CONCURRENT_EXECUTOR에서 오케스트레이션한 두 개의 HTTP_REQUEST 함수를 사용합니다.

하위 1 - 사용자 프로필(**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%}"
    }
  }
}
```

하위 2 - 디바이스 정보(**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%}"
    }
  }
}
```

실행기(**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%}"
  }
}

```

이 예시는 다음과 같이 설정되어 있습니다.

1. MediaTailor는 두 하위 항목을 동시에 시작합니다(MaxConcurrency는 2).
2. deviceCall 하위는이 있는 경우에만 실행device_id됩니다(RunCondition).
3. 두 호출이 모두 완료된 후(1500ms 이내) 출력 블록은 결합된 결과물에 기록합니다player_params.
4. 1500ms 제한 시간을 초과하면 MediaTailor는 모든 출력을 삭제하고 보강 데이터 없이 진행합니다.

AWS 서비스 요청

사용해야 하는 경우

를 사용하면 매니페스트 개인화 중에 지원되는 AWS 서비스 APIs 호출AWS_SERVICE_REQUEST할 수 있습니다. 현재 상황별 광고 타겟팅을 위해 Elemental Inference GetMetadata를 호출할 수 있습니다.

MediaTailor는 자체 서비스 자격 증명으로 요청에 서명하고 필요한 인증 및 보안 헤더를 자동으로 추가합니다. IAM 역할을 구성하거나 자격 증명을 관리할 필요가 없습니다.

사전 조건, 리소스 정책 및 리전별 고려 사항이 포함된 안내 연습은 섹션을 참조하세요 [Elemental Inference 통합](#).

구성 필드

AWS_SERVICE_REQUEST 함수에는 다음 필드가 있습니다.

- 런타임(필수) - 표현식 언어입니다. 이를 로 설정합니다JSONATA.
- TargetService(필수) - 대상 AWS 서비스입니다. 현재 elemental-inference만 지원됩니다.
- TargetRegion(필수) AWS - 대상 서비스의 리전입니다. 정적 값(예: us-west-2) 또는 동적 해상도에 표현JSONata식(예:)을 사용할 수 있습니다{%inference.region%}.
- MethodType(필수) - HTTP 메서드입니다. 지원되는 값은 대상 서비스 API에 POST따라 GET 및 입니다.
- URL(필수) - AWS 서비스 API의 HTTPS 엔드포인트입니다. 유효한 AWS 엔드포인트 URL이어야 합니다. JSONata 표현식을 사용하여 URL을 동적으로 빌드할 수 있습니다.
- 헤더(선택 사항) - 요청에 포함할 추가 HTTP 헤더입니다. 헤더 제한은 섹션을 참조하세요 [the section called “보안 모델”](#).
- 본문(조건부) - 요청 본문입니다. MethodType가 인 경우 필수입니다POST. JSONata 표현식을 사용하여 세션 상태에서 동적으로 본문을 빌드할 수 있습니다.
- RequestTimeoutMilliseconds(필수) - 응답을 기다리는 시간입니다.
- 출력(선택 사항) - 호출이 완료된 후 생성할 값을 정의합니다. 각 항목은 response 객체를 참조할 수 있는 표현식에 출력 키를 매핑합니다. 출력 표현식은에 설명된 것과 동일한 response 객체를 참조할 수 있습니다 [the section called “응답 필드”](#).

보안 모델

MediaTailor는 자체 서비스 역할을 사용하여 아웃바운드 요청에 서명합니다. MediaTailor가 대상 서비스를 호출하기 위해 IAM 역할을 생성하거나 구성할 필요가 없습니다.

대상 AWS 리소스에는 aws:SourceAccount 및 aws:SourceArn 조건을 사용하여 MediaTailor 서비스 보안 주체에 액세스 권한을 부여하는 리소스 정책이 있어야 합니다. 이러한 조건은 MediaTailor가 계정 및 특정 재생 구성을 대신하여 리소스에 액세스할 수 있도록 하여 교차 서비스 액세스 보호를 제공합니다. 전체 리소스 정책 예제는 섹션을 참조하세요 [MediaTailor에 Elemental Inference 피드에 대한 액세스 권한 부여](#).

헤더 제한: X-Amz- 접두사가 있는 모든 헤더가 예약되어 있습니다. MediaTailor는 이러한 헤더를 필수 인증 및 교차 서비스 액세스 보호 값으로 덮어씁니다. 구성에 X-Amz- 헤더를 포함하지 마십시오.

요청 처리 방법

MediaTailor는 다음 세 단계로 `AWS_SERVICE_REQUEST` 함수를 처리합니다.

1. 요청 빌드 - MediaTailor는 현재 세션 상태를 기준으로 `UrlHeaders`, 및 `Body` 표현식을 평가합니다. URL은 지정된 서비스에 유효한 엔드포인트여야 합니다.
2. 요청 서명 - MediaTailor는 지정된 `TargetService` 및에 대한 자체 서비스 자격 증명을 사용하여 요청을 인증합니다 `TargetRegion`.
3. 응답 처리 - 호출이 완료되면 MediaTailor는 출력 블록의 표현식을 평가합니다. 이러한 표현식은 원래 세션 상태와 호출에서 반환된 `response` 객체를 모두 참조할 수 있습니다.

응답 필드

호출이 완료되면 출력 표현식에서 다음 필드를 참조할 수 있습니다.

Field	유형	설명
<code>response.body</code>	객체 또는 배열	JSON으로 구문 분석된 응답 본문입니다. 본문이 20,000자를 초과하거나 유효한 JSON이 아닌 <code>null</code> 경우로 설정합니다.
<code>response.statusCode</code>	Integer	AWS 서비스에서 반환하는 HTTP 상태 코드입니다. 네트워크 장애 <code>null</code> 시로 설정합니다.
<code>response.text</code>	문자열	원시 응답 본문은 문자열로, 20,000자로 잘립니다. 네트워크 장애 "Internal Error" 시로 설정합니다.

⚠ Important

최대 응답 크기는 20,000자입니다. 이 제한을 초과하는 응답은 `response.body` 설정됩니다 `null`.

오류 처리

다음은 MediaTailor가 `AWS_SERVICE_REQUEST` 함수에 대한 오류를 처리하는 방법을 설명합니다.

- 요청 시간이 초과되면(초과`RequestTimeoutMilliseconds`) `response.statusCode`는 `null`이고는 `response.body`입니다 `null`. 출력 표현식은 여전히 실행됩니다.
- 대상 서비스가 4xx 또는 5xx 오류를 반환하는 경우 `response.statusCode`는 HTTP 상태 코드를 포함하고 오류 응답을 `response.body` 포함합니다(유효한 JSON인 경우 20,000자 미만).
- 응답 본문이 20,000자를 초과하는 경우 응답이 성공했`null`더라도 `response.body`는 입니다. 원시 잘린 콘텐츠 `response.text`에를 사용합니다.
- 함수가 실패하거나 빈 출력을 생성하면 MediaTailor는 함수의 기여도 없이 사용 가능한 값을 사용하여 광고 삽입을 진행합니다. 광고 중단은 차단되지 않습니다.

💡 Tip

오류를 정상적으로 처리하려면 항상 `response.statusCode` 출력 표현식을 확인합니다.

예: Elemental Inference GetMetadata

다음 함수는 Elemental Inference를 호출`GetMetadata`하여 현재 광고 시간 주변의 콘텐츠 창에 대한 컨텍스트 메타데이터를 검색합니다. Body 필드는 JSONata 표현식을 사용하여 `inference.*` 변수에서 동적으로 요청 JSON을 구성합니다. 분류 IAB 범주와 GARM 브랜드 안전 신호를 추출한 다음 ADS 요청에 사용할 플레이어 파라미터로 저장합니다.

```
{
  "FunctionId": "eiContextualMetadata",
  "FunctionType": "AWS_SERVICE_REQUEST",
  "AwsServiceRequestConfiguration": {
    "Runtime": "JSONATA",
    "TargetService": "elemental-inference",
    "TargetRegion": "{%inference.region%}",
  }
}
```

```

        "MethodType": "POST",
        "Url": "{%inference.dataEndpoint & '/v1/feed/' & inference.feedId & '/input/0/
metadata'%}",
        "Headers": {
            "Content-Type": "application/json"
        },
        "Body": "{%'\{"outputName\": \"my-contextual-output\", \"timeSpecification\":
{\\"ptsBased\": {\\"startPts\": ' & $string(($exists(inference.previousBreakEndPts)
and inference.previousBreakEndPts > inference.pts - 30 * inference.timescale ?
inference.previousBreakEndPts : inference.pts - 30 * inference.timescale))
& ', \\"endPts\": ' & $string(inference.pts + 1) & ', \\"timescale\": ' &
$string(inference.timescale) & '}}, \\"parameters\": {\\"contextualMetadata\": {}}}'
%}",
        "RequestTimeoutMilliseconds": 2000,
        "Output": {
            "player_params.iabCategories": "{%response.statusCode = 200 ?
$join($distinct(response.body.items.metadata.contextualMetadata.iabTaxonomy.categories.uniqueI
',') : ''%}",
            "player_params.garmExcluded": "{%response.statusCode = 200 ?
$join($distinct(response.body.items.metadata.contextualMetadata.garm.suitability.categories[fl
= true].category), ',') : ''%}"
        }
    }
}

```

런타임 시 Body 표현식은 다음과 같은 JSON 페이로드를 생성합니다.

```

{
  "outputName": "my-contextual-output",
  "timeSpecification": {
    "ptsBased": {
      "startPts": 162000000,
      "endPts": 163600001,
      "timescale": 90000
    }
  },
  "parameters": {
    "contextualMetadata": {}
  }
}

```

startPts 값은 이전 광고 시간의 끝(inference.previousBreakEndPts) 또는 현재 시간의 30초 전 중 더 나중 값입니다. 값은 현재 브레이크의 PTS에 1을 더한 endPts 값입니다.

Note

를 Elemental Inference 피드의 컨텍스트 메타데이터 출력 이름으로 `my-contextual-output` 바꿉니다.

Note

MediaTailor는 Elemental Inference에 대한 요청에 `x-amzn-elemental-inference-skip-poll` 헤더를 자동으로 포함합니다. 이렇게 하면 광고 중단 시간에 적합한 지연 시간이 짧은 응답이 보장됩니다.

사전 조건 및 리소스 정책을 포함한 전체 설정 가이드는 섹션을 참조하세요 [Elemental Inference 통합](#).

구성 규칙

MediaTailor는 함수 구성에 다음 제한을 적용합니다.

규칙	한도
최대 중첩 깊이	2(실행기에는 함수가 포함될 수 있지만 이러한 하위 항목은 실행기 자체일 수 없음)
시퀀스당 단계	1~10
동시 실행기당 하위 항목	1~10
최대 동시성(동시 실행기)	1 ~ 2
수명 주기 후크당 총 함수 실행 수	20
순환 참조	허용되지 않음. 함수는 직접 또는 간접적으로 자신을 참조할 수 없습니다.
함수 존재 여부	상위 실행기를 생성하거나 업데이트하기 전에에서 참조되는 모든 함수가 존재FunctionL ist 해야 합니다.

규칙	한도
고유 네임스페이스(동시 실행기)	각 하위 항목은 실행기 내에 고유한 별칭 또는 함수 ID가 있어야 합니다.

함수를 생성할 때 MediaTailor는 표현식 구문을 검증하고, 제한된 함수를 확인하고, 참조된 모든 함수가 존재하는지 확인하고, 순환 참조를 감지합니다. 함수를 재생 구성에 연결하면 MediaTailor는 전체 함수 트리의 모든 출력 키가 할당된 수명 주기 후크와 호환되는지 추가로 검증합니다.

Elemental Inference를 사용한 컨텍스트 광고 타겟팅

Elemental Inference는 비디오 콘텐츠를 분석하고 각 샷에 대해 IAB 콘텐츠 분류 분류 및 GARM 브랜드 안전 신호를 생성합니다. MediaTailor를 Elemental Inference와 통합하면 컨텍스트 메타데이터로 광고 요청을 보강할 수 있습니다. 그러면 광고 결정 서버가 콘텐츠에서 일어나는 일을 기반으로 광고를 대상으로 지정할 수 있습니다.

이 통합에서는 AWS_SERVICE_REQUEST 함수 유형을 사용하여 광고 시간 동안 Elemental Inference GetMetadata API를 호출합니다. 함수 유형에 대한 자세한 내용은 섹션을 참조하세요 [AWS 서비스 요청](#).

다음은 통합 흐름에 대한 설명입니다.

1. Elemental Inference는 콘텐츠를 분석하고 샷 수준 분류를 생성합니다.
2. AWS Elemental MediaLive는 Elemental Inference 쿼리 파라미터를 사용하여 SCTE-35 마커를 장식합니다.
3. 각 광고 시간에 MediaTailor는 콘텐츠 기간의 분류를 검색GetMetadata하기 위해 인증된 요청을 보냅니다.
4. 함수의 출력 표현식은 IAB 및 GARM 신호를 추출하여 광고 결정 서버에 전달합니다.

사전 조건

이 통합을 구성하기 전에 다음이 필요합니다.

- 컨텍스트 메타데이터 출력이 있는 Elemental Inference 피드입니다. 피드 생성에 대한 지침은 Elemental Inference 사용 설명서 [의 피드 생성](#)을 참조하세요.
- MediaTailor 액세스 권한을 부여하는 피드의 리소스 정책입니다. [the section called “MediaTailor에 Elemental Inference 피드에 대한 액세스 권한 부여”](#)을(를) 참조하세요.

- 컨텍스트 메타데이터 보강이 활성화된 AWS Elemental MediaLive 채널입니다. Elemental Inference 쿼리에 정확한 PTS 타이밍 값을 제공하려면 MediaLive 보강을 활성화합니다. 컨텍스트 메타데이터 보강 구성에 대한 지침은 AWS Elemental MediaLive 사용 설명서의 [컨텍스트 메타데이터 보강 설정](#)을 참조하세요.

MediaTailor와 Elemental Inference가 동일한 AWS 리전에 있을 필요는 없습니다. 지원되는 조합은 각 서비스가 옵트인 리전에 있는지 여부에 따라 달라집니다. 자세한 내용은 [the section called “MediaTailor에 Elemental Inference 피드에 대한 액세스 권한 부여”](#)을 참조하세요. Elemental Inference를 사용할 수 있는 리전은 [Elemental Inference 엔드포인트 및 할당량을 참조하세요](#).

MediaTailor에 Elemental Inference 피드에 대한 액세스 권한 부여

MediaTailor는 자체 서비스 자격 증명을 GetMetadata 사용하여 호출합니다. 대상 Elemental Inference 피드에는 MediaTailor 서비스 보안 주체를 신뢰하는 리소스 정책이 있어야 합니다.

MediaTailor는 계정 및 재생 구성을 식별하는 보안 헤더를 자동으로 추가하여 Elemental Inference 피드가 요청이 MediaTailor 리소스에서 시작되었는지 확인할 수 있도록 합니다.

다음 표에는 배포 시나리오에 따라 사용할 서비스 보안 주체가 나와 있습니다.

시나리오	서비스 보안 주체
동일한 non-opt-in 리전 또는 다른 non-opt-in 리전의 MediaTailor 및 Elemental Inference	mediatailor.amazonaws.com
비옵트인 리전에서 Elemental Inference를 호출하는 non-opt-in 리전의 MediaTailor	mediatailor. region .amazonaws.com (예: mediatailor.af-south-1.amazonaws.com)
단일 MediaTailor 리전에 대한 액세스 제한	mediatailor. region .amazonaws.com

Note

MediaTailor는 현재 cross-opt-in-Region 액세스(다른 옵트인에 옵트인 또는 non-opt-in 않음)를 지원하지 않습니다.

다음 예제 리소스 정책은 MediaTailor에 피드GetMetadata에서 호출할 수 있는 권한을 부여합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "mediatailor.amazonaws.com"
      },
      "Action": "elemental-inference:GetMetadata",
      "Resource": "arn:aws:elemental-inference:us-west-2:123456789012:feed/my-
contextual-feed",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:mediatailor:us-
west-2:123456789012:playbackConfiguration/*"
        }
      }
    }
  ]
}
```

us-west-2, 123456789012 및 리전, 계정 ID 및 피드 ID my-contextual-feed로 바꿉니다.

이 aws:SourceAccount 조건은 계정을 대신하여 작동하는 MediaTailor만 피드에 액세스할 수 있도록 합니다. aws:SourceArn 조건은 특정 재생 구성에 대한 액세스를 추가로 제한합니다. 더 엄격한 범위를 지정하려면 와일드카드를 특정 재생 구성 이름으로 바꿉니다.

다음 예제에서는 서비스 보안 주체에 리전 이름이 포함된 옵트인 리전에 대한 리소스 정책을 보여줍니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "mediatailor.af-south-1.amazonaws.com"
      },
      "Action": "elemental-inference:GetMetadata",
```



```

    "Resource": "arn:aws:elemental-inference:us-west-2:123456789012:feed/my-
contextual-feed",
    "Condition": {
        "StringEquals": {
            "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
            "aws:SourceArn": "arn:aws:mediatailor:af-
south-1:123456789012:playbackConfiguration/*"
        }
    }
}
]
}

```

MediaTailor 콘솔을 사용하여 구성

MediaTailor 콘솔은 Elemental Inference 컨텍스트 메타데이터 함수를 구성하기 위한 안내 레시피를 제공합니다. 레시피는 URL을 자동으로 구성하고, 요청 본문을 템플릿화하고, 출력 표현식을 생성합니다.

시작하기 전 준비 사항

[the section called “사전 조건”](#)의 단계를 수행하세요. 컨텍스트 메타데이터 출력이 있는 Elemental Inference 피드, MediaTailor 액세스 권한을 부여하는 리소스 정책, 컨텍스트 메타데이터 보강이 활성화된 MediaLive 채널이 필요합니다.

함수 생성

1. [MediaTailor 콘솔](#)을 엽니다. 탐색 창에서 함수를 선택합니다.
2. 함수 생성을 선택합니다.
3. 함수 유형에서 AWS_SERVICE_REQUEST를 선택합니다.
4. Elemental Inference - 컨텍스트 메타데이터 레시피를 선택합니다.
5. 피드 소스에서 다음 중 하나를 선택합니다.
 - SCTE-35 마커에서(권장) - MediaTailor는 MediaLive로 데코레이션된 SCTE-35 마커에서 피드 엔드포인트, 리전 및 타이밍 정보를 추출합니다. 다른 구성은 필요하지 않습니다.
 - 내 계정의 피드 찾아보기 - 콘솔에 선택한 리전의 Elemental Inference 피드가 나열됩니다. 피드를 선택하여 구성을 자동으로 채웁니다. Elemental Inference 피드가 MediaTailor 리소스와 동일한 계정에 있는 경우이 옵션을 사용합니다.

- 출력 이름에서 Elemental Inference 피드에 구성된 컨텍스트 메타데이터 출력의 이름을 선택하거나 입력합니다.

Note

피드 브라우저를 사용하여 피드를 선택한 경우 콘솔은 피드 구성에서 사용 가능한 출력 이름을 자동으로 채웁니다.

- 출력 구성에서 추출할 컨텍스트 신호를 선택합니다.

- IAB 콘텐츠 범주 - IAB 분류 범주 IDs 심포로 구분된 문자열로 추출합니다.
- GARM 브랜드 안전 - 플래그가 지정된 GARM 브랜드 안전 범주를 추출합니다.

광고 결정 서버 파라미터 이름과 일치하도록 출력 키 이름(예: `player_params.iabCategories`)을 사용자 지정할 수 있습니다.

- 제한 시간에 응답을 기다리는 최대 시간을 입력합니다. 기본값은 2000밀리초입니다.
- (선택 사항) 리소스 정책에서 생성된 정책 코드 조각을 검토합니다. Elemental Inference 피드에 대한 액세스를 아직 구성하지 않은 경우 정책을 복사하고 Elemental Inference 콘솔 또는 `PutFeedPolicy` API를 사용하여 적용합니다.
- 생성(Create)을 선택합니다.

재생 구성에 연결

함수를 생성한 후 `PRE_ADS_REQUEST` 수명 주기 후크의 재생 구성에 연결합니다. 직접 연결하거나 `SEQUENTIAL_EXECUTOR` 함수 내에서 단계로 연결할 수 있습니다.

- 탐색 창에서 재생 구성을 선택하고 구성을 선택합니다.
- 함수 매핑에서 편집을 선택합니다.
- `PRE_ADS_REQUEST` 후크에서 생성한 함수를 선택합니다.
- 구성을 저장합니다.

Important

스트림에 Elemental Inference 마커가 없는 광고 중단이 포함된 경우 컨텍스트 메타데이터 함수를 호출 `inference.enriched` 하기 전에 확인하는 실행 조건과 `SEQUENTIAL_EXECUTOR` 함께를 사용합니다. [the section called “다른 함수와의 체인”](#)을(를) 참조하세요.

API 또는 CLI를 사용하여 구성

Elemental Inference 쿼리에 정확한 타이밍 정보를 제공하려면 MediaLive 컨텍스트 메타데이터 보강을 활성화합니다. 다음 섹션에서는 MediaLive의 SCTE-35 마커를 사용하는 권장 프로덕션 구성과 테스트용 수동 구성 옵션에 대해 설명합니다.

MediaLive의 SCTE-35 마커 사용(프로덕션)

MediaLive는 컨텍스트 메타데이터 보강을 위해 구성되면 Elemental Inference 쿼리 파라미터를 SCTE-35 광고 중단 마커에 직접 포함합니다. MediaTailor는 이를 자동으로 구문 분석하고 `inference.*` 변수로 노출합니다. 사용 가능한 변수의 전체 목록은 섹션을 참조하세요 [the section called “Elemental Inference 컨텍스트 메타데이터 변수\(`inference.\*`\)”](#).

다음과 같은 이유로 프로덕션 배포에이 접근 방식을 사용합니다.

- 하드 코딩된 엔드포인트는 필요하지 않습니다.
- 리전 및 피드는 스트림에서 자동으로 확인됩니다.
- PTS 정렬은 인코더에서 처리됩니다.

MediaLive 채널에서 컨텍스트 메타데이터 보강을 활성화하는 방법에 대한 지침은 AWS Elemental MediaLive 사용 설명서의 [컨텍스트 메타데이터 보강 설정](#)을 참조하세요.

다음 예제에서는 SCTE-35 보강을 사용한 전체 함수 구성을 보여줍니다.

```
{
  "FunctionId": "eiContextualMetadata",
  "FunctionType": "AWS_SERVICE_REQUEST",
  "AwsServiceRequestConfiguration": {
    "Runtime": "JSONATA",
    "TargetService": "elemental-inference",
    "TargetRegion": "${inference.region}",
    "MethodType": "POST",
    "Url": "${inference.dataEndpoint} & '/v1/feed/' & inference.feedId & '/input/0/metadata'${inference.feedId}",
    "Headers": {
      "Content-Type": "application/json"
    },
    "Body": "${inference.outputName}: \"my-contextual-output\", \"timeSpecification\": {\"ptsBased\": {\"startPts\": ' & $string(($exists(inference.previousBreakEndPts) and inference.previousBreakEndPts > inference.pts - 30 * inference.timescale ?
```

```

inference.previousBreakEndPts : inference.pts - 30 * inference.timescale))
& ', \"endPts\": ' & $string(inference.pts + 1) & ', \"timescale\": ' &
$string(inference.timescale) & '}}, \"parameters\": {\"contextualMetadata\": {}}}'
%}",
    "RequestTimeoutMilliseconds": 2000,
    "Output": {
        "player_params.iabCategories": "{%response.statusCode = 200 ?
$join($distinct(response.body.items.metadata.contextualMetadata.iabTaxonomy.categories.uniqueI
',') : ''%}",
        "player_params.garmExcluded": "{%response.statusCode = 200 ?
$join($distinct(response.body.items.metadata.contextualMetadata.garm.suitability.categories[fl
= true].category), ',') : ''%}"
    }
}
}

```

다음 목록은 이전 예제의 키 필드를 설명합니다.

- TargetRegion는 inference.region를 사용하여 올바른 리전에 동적으로 서명합니다.
- URL은 inference.dataEndpoint 및 로 구성됩니다inference.feedId.
- 본문은 이전 광고 시간의 끝(또는 최대 30초 룩백)부터 현재 시간으로 쿼리합니다.
- 출력은 IAB 범주 IDs와 플래그가 지정된 GARM 범주를 추출합니다.

Note

를 Elemental Inference 피드의 컨텍스트 메타데이터 출력의 이름으로 my-contextual-output 바꿉니다.

Note

URL 경로에서 /v1/feed/{feed-id}/input/0/metadata는 Elemental Inference 피드의 입력 인덱스를 0 나타냅니다. 대부분의 피드는 인덱스에서 단일 입력을 갖습니다0. 피드에 여러 입력이 있는 경우를 적절한 인덱스0로 바꿉니다.

Note

MediaTailor는 Elemental Inference에 대한 요청에 `x-amzn-elemental-inference-skip-poll` 헤더를 자동으로 포함합니다. 이렇게 하면 광고 중단 시간에 적합한 지연 시간이 짧은 응답이 보장됩니다. 이 헤더를 구성할 필요가 없습니다.

Important

스트림에 Elemental Inference 마커(예: 혼합 콘텐츠 소스)가 없는 광고 시간이 포함된 경우 `inference.enriched`를 `RunCondition` 확인하는 `SEQUENTIAL_EXECUTOR` 로이 함수를 래핑합니다.

```
{ "FunctionId": "eiContextualMetadata", "RunCondition": "{%inference.enriched = true%}" }
```

`inference.enriched`이 `false` 있거나 없는 경우 `inference.*` 변수를 사용할 수 없으며 변수를 참조하는 표현식은 유효한 결과를 생성하지 않습니다.

수동 구성(테스트 전용)

이 구성은 테스트에만 사용됩니다. MediaLive의 SCTE-35 보강이 없으면 PTS 값을 수동으로 결정해야 하며 동적 타이밍 정렬의 이점을 누릴 수 없습니다. 이 모드에는 제한된 기능이 있습니다. 프로덕션 워크플로에는 지원하지 않습니다.

```
{
  "FunctionId": "eiContextualMetadataManual",
  "FunctionType": "AWS_SERVICE_REQUEST",
  "AwsServiceRequestConfiguration": {
    "Runtime": "JSONATA",
    "TargetService": "elemental-inference",
    "TargetRegion": "us-west-2",
    "MethodType": "POST",
    "Url": "https://abc123.elemental-inference-data.us-west-2.amazonaws.com/v1/feed/my-feed-id/input/0/metadata",
    "Headers": {
      "Content-Type": "application/json"
    }
  },
}
```

```

    "Body": "{%'\\"outputName\\": \"my-contextual-output\\", \\"timeSpecification\\":
    {\\"ptsBased\\": {\\"startPts\\": 6300001, \\"endPts\\": 9000001, \\"timescale\\": 90000}},
    \\"parameters\\": {\\"contextualMetadata\\": {}}}' %}",
    "RequestTimeoutMilliseconds": 2000,
    "Output": {
        "player_params.iabCategories": "{%response.statusCode = 200 ?
    $join($distinct(response.body.items.metadata.contextualMetadata.iabTaxonomy.categories.uniqueI
    ',') : ''%}"
    }
}
}

```

Note

my-contextual-output를 Elemental Inference 피드의 컨텍스트 메타데이터 출력 이름으로 바꿉니다.

Note

startPts 및 endPts 값을 콘텐츠에 적합한 PTS 값으로 바꿉니다. 이 예제에서 값은 90kHz 시간대의 30초 기간을 나타냅니다. SCTE-35 마커 보강이 없으면 콘텐츠 파이프라인에서 PTS 값을 결정해야 합니다.

Note

abc123.elemental-inference-data.us-west-2.amazonaws.com, my-feed-id 및 를 피드의 데이터 엔드포인트, 피드 ID 및 리전us-west-2으로 바꿉니다. 데이터 엔드포인트는 Elemental Inference GetFeed API 응답 또는 콘솔에서 찾을 수 있습니다.

재생 구성에 연결

함수를 생성한 후 PutPlaybackConfiguration API를 사용하여 PRE_ADS_REQUEST 수명 주기 후크의 재생 구성에 연결합니다. 함수 매핑에 함수 ID를 포함합니다.

```

{
    "FunctionMappings": {

```

```

    "PRE_ADS_REQUEST": {
        "FunctionId": "eiContextualMetadata"
    }
}
}

```

에 래핑 `SEQUENTIAL_EXECUTOR` 하여 다른 함수와 연결할 수도 있습니다. 자세한 내용은 [the section called “다른 함수와의 체인”](#) 단원을 참조하십시오.

기간 이해

본문 표현식은 `GetMetadata` 요청에 대한 PTS 기반 기간을 빌드합니다. 다음 목록은 록백 로직을 설명합니다.

- `endPts` = 현재 광고 시간의 PTS(에서 `inference.pts`) + 1(포함 끝).
- `startPts` = 다음 중 나중 날짜: 이전 광고 시간 종료(`inference.previousBreakEndPts`) 또는 현재 시간 30초 전.
- 이렇게 하면 광고 중단 사이에 재생된 콘텐츠만 쿼리하여 이전 광고 포드 내의 콘텐츠를 피할 수 있습니다.
- 세션의 첫 번째 광고 시간에는 `inference.previousBreakEndPts`를 사용할 수 없으므로 전체 30초 록백이 사용됩니다.
- 30초 기간은 Elemental Inference가 지원하는 최대 록백입니다.
`inference.previousBreakEndPts`를 사용할 수 있는 경우 표현식은 해당 값의 이후 또는 30초 록백을 사용합니다. 이렇게 하면 이전 광고 중단 이후의 콘텐츠만 쿼리에 포함됩니다.

광고 요청에 응답 사용

다음 표에서는 사용 가능한 출력 대상 접두사와 해당 동작을 설명합니다.

대상 접두사	동작
<code>player_params.*</code>	[<code>player_params.key</code>] 대체를 통해 현재 광고 시간의 광고 결정 서버 URL에서 사용할 수 있습니다. 광고 시간 동안 지속되지 않습니다.
<code>temp.*</code>	의 후속 함수에 대한 중간 값입니다 <code>SEQUENTIAL_EXECUTOR</code> . 지속되지 않습니다.

대상 접두사	동작
<code>adsRequest.headers.*</code>	광고 결정 서버 요청에서 HTTP 헤더로 전송됩니다.
<code>adsRequest.url</code>	광고 결정 서버 URL을 완전히 재정의합니다.

출력 표현식 레시피에 대한 자세한 내용은 섹션을 참조하세요 [예제 3: 컨텍스트 메타데이터](#).

다른 함수와의 체인

Elemental Inference 호출을 다른 함수와 결합하려면 래핑합니다 `SEQUENTIAL_EXECUTOR`. 다음 예제는 Elemental Inference가 범주를 성공적으로 반환한 경우에만 두 번째 함수가 실행되는 파이프라인을 보여줍니다.

```
{
  "FunctionId": "contextualAdPipeline",
  "FunctionType": "SEQUENTIAL_EXECUTOR",
  "SequentialExecutorConfiguration": {
    "Runtime": "JSONATA",
    "FunctionList": [
      { "FunctionId": "eiContextualMetadata" },
      {
        "FunctionId": "buildEnrichedAdsRequest",
        "RunCondition": "{%$exists(player_params.iabCategories) and
player_params.iabCategories != ''}"
      }
    ],
    "TimeoutMilliseconds": 3000
  }
}
```

는 Elemental Inference 함수가 데이터를 생성할 때 `buildEnrichedAdsRequest`만 실행 `RunCondition`되도록 합니다. 그런 다음 컨텍스트 데이터를 사용하여 광고 결정 서버 요청을 빌드할 수 있습니다.

문제 해결

다음 표에서는 일반적인 문제와 해결 방법을 설명합니다.

증상	원인	해결 방법
403 함수 로그에서 AccessDenied	리소스 정책 누락 또는 부정확	피드 정책에 MediaTailor 서비스 보안 주체가 포함되어 있고 <code>aws:SourceAccount</code> 조건이 계정 ID와 일치하는지 확인합니다. 함수 실행 로그 보기에 대한 자세한 내용은 섹션을 참조하세요 문제 해결 및 모니터링 .
<code>inference.enriched</code> 은 (는) false	SCTE-35 마커에는 Elemental Inference 데이터가 포함되지 않습니다.	MediaLive 채널에서 컨텍스트 메타데이터 보강이 활성화되어 있는지 확인합니다.
<code>inference.parseError</code> 은(는) true	SCTE-35 마커의 잘못된 Elemental Inference 데이터	MediaLive 채널 구성 및 인코더 소프트웨어 버전을 확인합니다.
빈 응답(항목 없음)	쿼리 기간에 분석된 콘텐츠가 없음	Elemental Inference 피드가 콘텐츠를 능동적으로 처리하고 있는지, 기간이 분석된 세그먼트를 포함하는지 확인합니다.
응답 잘림	응답이 20,000자 제한을 초과함	분석된 샷을 더 적게 반환하려면 기간을 줄입니다.

Note

어떤 이유로든(제한 시간, 권한 부여 오류 또는 빈 응답) 함수가 실패하면 MediaTailor는 함수의 기여도 없이 사용 가능한 값을 사용하여 광고 삽입을 진행합니다. 광고 중단은 차단되지 않습니다.

함수 생성 및 관리

이 페이지에서는 MediaTailor 콘솔을 사용하여 함수를 생성, 연결, 업데이트 및 삭제하는 방법을 안내합니다. 함수를 사용하면 재생 중에 키 포인트에서 로직을 실행하여 세션 동작 및 광고 요청을 사용자 지정할 수 있습니다.

함수 설정

콘솔을 사용하여 함수 생성

1. [MediaTailor 콘솔](#)을 엽니다.
2. 탐색 창에서 함수를 선택합니다.
3. 함수 생성을 선택합니다.
4. 함수 ID에 함수의 고유한 이름(예: fetchIdentity)을 입력합니다.
5. 함수 유형에서, CUSTOM_OUTPUT, HTTP_REQUESTVAST_REQUEST, 또는 SEQUENTIAL_EXECUTOR를 선택합니다CONCURRENT_EXECUTOR. 콘솔에 선택한 유형의 구성 필드가 표시됩니다.
6. 유형별 구성 필드를 입력합니다. 각 함수 유형 및 해당 필드에 대한 설명은 섹션을 참조하세요[함수 유형 및 구성](#).
7. 함수 생성을 선택합니다.

재생 구성에 함수 연결

함수 매핑(수명 주기 후크와 함수 간의 링크)을 통해 재생 구성에 연결할 때까지 함수는 실행되지 않습니다. 함수 매핑은 함수를 트리거하는 수명 주기 후크를 지정합니다.

1. MediaTailor 콘솔을 엽니다.
2. 탐색 창에서 구성을 선택합니다.
3. 업데이트할 재생 구성을 선택합니다.
4. 함수 매핑 섹션에서 편집을 선택합니다.
5. 각 수명 주기 후크에 대해 연결할 함수를 선택합니다.
 - 세션 초기화 후크 - 세션 시작 시 한 번 실행할 함수를 선택합니다.
 - 광고 요청 후크 - 각 ADS 요청 전에 실행할 함수를 선택합니다.
 - 광고 게시 응답 후크 - 각 ADS 응답을 수신하고 구문 분석한 후 실행할 함수를 선택합니다.

- 매니페스트 삽입 전 후크 - 매니페스트에 광고가 삽입되기 직전에 최종 광고 세트에서 실행할 함수를 선택합니다.

6. 저장을 선택합니다.

함수 관리

함수 보기

함수로 이동하여 계정의 모든 함수를 확인합니다. 함수 이름을 선택하여 구성을 확인합니다.

기존 함수 업데이트

함수를 업데이트하려면 함수로 이동하여 함수를 선택하고 구성을 수정합니다. 업데이트는 전체 함수 정의를 대체합니다.

Tip

변경하기 전에 함수 구성의 사본을 저장합니다. 함수에는 기본 제공 버전 관리 또는 롤백이 없습니다.

함수 제거

재생 구성에서 함수 분리

함수를 삭제하기 전에 함수를 참조하는 모든 재생 구성에서 함수를 제거합니다.

1. MediaTailor 콘솔을 엽니다.
2. 탐색 창에서 구성을 선택합니다.
3. 재생 구성을 선택합니다.
4. 함수 매핑 섹션에서 편집을 선택합니다.
5. 수명 주기 후크에서 함수를 제거합니다.
6. 저장을 선택합니다.

함수 삭제

함수로 이동하여 함수를 선택한 다음 삭제를 선택합니다.

삭제 차단 규칙

MediaTailor는 아직 사용 중인 함수를 삭제하지 못하도록 합니다.

조건	결과
함수는 함수 매핑을 통해 재생 구성에 연결됩니다.	삭제가 차단되었습니다. 먼저 함수를 분리합니다.
함수는 SEQUENTIAL_EXECUTOR 또는에서 참조됩니다. CONCURRENT_EXECUTOR	삭제가 차단되었습니다. 먼저 상위 함수에서 참조를 제거합니다.
함수는 어디에서도 참조되지 않음	삭제에 성공합니다.

검증 규칙

MediaTailor는 함수를 생성하거나 업데이트할 때 함수를 검증합니다. 다음 검사가 수행됩니다.

- 표현식 구문 - 모든 표현식은 유효한 JSONata여야 합니다.
- 제한된 함수 - 표현식은 제한된 JSONata 함수를 호출할 수 없습니다. 전체 목록은 섹션을 참조 [JSONata 표현식 참조](#)하세요.
- 출력 키 접두사 - 모든 출력 키는 인식된 네임스페이스 접두사로 시작해야 합니다. 허용되는 접두사 목록은 섹션을 참조하세요 [수명 주기 후크](#).
- 함수 참조 -의 모든 함수 IDs는 기존 함수를 참조FunctionList해야 합니다.
- 순환 참조 - 함수는 직접 또는 간접적으로 자신을 참조할 수 없습니다.
- 중첩 깊이 -는 다른 함수를 호출할 SEQUENTIAL_EXECUTOR 수 있지만 해당 함수 자체는 실행기가 될 수 없습니다.

특정 값 및 크기 제한은 섹션을 참조하세요 [한도](#).

API 참조

프로그래밍 방식으로 함수를 관리하려면 [AWS Elemental MediaTailor API 참조](#)를 참조하세요.

JSONata 함수에 대한 표현식 참조

이 페이지는 함수에서 사용할 수 있는 표현식 구문, 연산자 및 함수에 대한 전체 참조입니다. 출력 블록, URL 필드, 헤더 값, 본문 템플릿 및 실행 조건에 대한 표현식을 작성할 때 사용합니다.

표현식 구분 기호

함수에서 정의하는 모든 값은 상수 또는 표현식이며 둘 다의 혼합이 아닙니다. MediaTailor는 구분 기호를 기준으로 둘을 구분합니다.

구문	유형	평가
<code>https://ads.example.com/vast</code>	상수	평가 없이 있는 그대로 반환됩니다.
<code>{%session.client_ip%}</code>	표현식	런타임에 평가됩니다. 결과는 전체 값을 대체합니다.
<code>GET</code>	상수	있는 그대로 반환됩니다.
<code>{%'https://ads.example.com/vast?ip=' & session.client_ip%'}</code>	표현식	런타임에 평가됩니다.

Important

값은 완전히 상수이거나 완전히 표현식입니다. 두 값을 단일 값으로 혼합할 수 없습니다. 예를 들어, `hello {% 'world' %}`은 유효하지 않습니다. 정적 텍스트를 동적 값과 결합하려면 표현식 내에서 문자열 연결을 사용합니다 `{%'hello ' & 'world'%'}`.

언어 기본 사항

경로 탐색을 위한 점 표기법

점 표기법을 사용하여 입력 데이터를 통과합니다. 각 점은 객체 계층 구조로 한 수준 내려갑니다.

```
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
```

누락된 필드는 오류를 발생시키지 않고 반환됩니다.

```
temp.nonExistent           # null
```

```
temp.nonExistent.deeply.nested # null
```

와 문자열 연결 &

& 연산자는 두 문자열 값을 조인합니다. 비문자열 값은 자동으로 문자열로 변환됩니다.

```
'https://ads.example.com/vast?ip=' & session.client_ip
# "https://ads.example.com/vast?ip=192.0.2.1"

'duration=' & 30
# "duration=30"
```

조건부(3진수) 표현식

삼원 연산자를 사용하여 조건에 따라 두 값 중 하나를 반환합니다.

```
condition ? value_if_true : value_if_false
```

예시:

```
$exists(player_params.env) ? player_params.env : 'prod'
response.statusCode = 200 ? response.body.id : 'unknown'
$random() > 0.5 ? 'groupA' : 'groupB'
```

다방향 분기를 위해 삼원 표현식을 중첩할 수 있습니다.

```
$contains(session.user_agent, 'CTV') ? 'ctv'
: $contains(session.user_agent, 'Mobile') ? 'mobile'
: 'desktop'
```

를 사용한 변수 바인딩 :=

괄호 안의 := 연산자를 사용하여 표현식 내에서 중간 값을 할당합니다. 바인딩 변수는 묶는 괄호로 범위가 지정되며 표현식 외부에서 지속되지 않습니다.

```
(
  $base := 'https://ads.example.com';
  $base & '/vast?ip=' & session.client_ip
)
```

세미콜론은 괄호 안에 문을 구분합니다. 마지막 문은 표현식의 반환 값입니다.

```
(
  $code := response.statusCode;
  $code != null and $code >= 200 and $code < 300
    ? response.body.value
    : 'fallback'
)
```

연산자

Arithmetic

연산자	설명	예제	결과
+	덧셈	5 + 3	8
-	뺄셈	10 - 4	6
*	곱셈	6 * 7	42
/	나눗셈	15 / 4	3.75
%	모듈로	17 % 5	2

Important

플레이어 파라미터 및 세션 데이터의 입력 값은 문자열로 도착합니다. `$number()`를 사용하여 숫자 비교 또는 산술 전에 변환합니다. 문자열을 숫자와 비교하면 예상치 못한 결과가 발생합니다.

비교

연산자	설명	예제	결과
=	같음	response. statusCode = 200	true

연산자	설명	예제	결과
<code>!=</code>	같지 않음	<code>player_params.region != 'us-east-1'</code>	true us-east-1이 아닌 경우
<code><</code>	보다 작음	<code>avail.index < 3</code>	true 3 미만인 경우
<code>></code>	보다 큼	<code>\$number(player_params.age) > 18</code>	true 18세를 초과하는 경우
<code><=</code>	작거나 같음	<code>\$count(items) <= 10</code>	true 10 이하인 경우
<code>>=</code>	크거나 같음	<code>response.statusCode >= 400</code>	true 오류 상태인 경우

부울

연산자	설명	예제
<code>and</code>	Logical AND	<code>response.statusCode = 200 and \$exists(response.body.id)</code>
<code>or</code>	Logical OR	<code>player_params.region = 'us-east-1' or player_params.region = 'us-west-2'</code>

우선 순위에는 괄호를 사용합니다.

```
score > 0.5 and (tier = 'premium' or tier = 'gold')
```


Note

논리적 부정\$not()에 사용합니다. not 키워드 연산자는 없습니다.

멤버십(in)

연in산자는 값이 배열에 존재하는지 테스트합니다.

```
'premium' in segments      # true if segments contains 'premium'
player_params.region in ['us-east-1', 'us-west-2'] # true
```

연결(~>)

체인 연산자는 왼쪽 표현식의 결과를 오른쪽의 함수에 첫 번째 인수로 전달합니다.

```
session.user_agent ~> $lowercase ~> $trim
# equivalent to $trim($lowercase(session.user_agent))
```

허용된 함수

MediaTailor는 다음과 같은 내장 함수를 지원합니다. 여기에 나열되지 않은 모든 함수는 차단되며 함수를 생성하거나 업데이트할 때 검증 오류가 발생합니다.

유형 변환(3)

함수	설명	예제	결과
\$string(value)	문자열로 변환	\$string(200)	"200"
\$number(value)	숫자로 변환	\$number('42')	42
\$boolean(value)	부울로 변환	\$boolean(1)	true

내부 검사(6)

함수	설명	예제	결과
<code>\$length(string)</code>	문자열 길이	<code>\$length('hello')</code>	5
<code>\$count(array)</code>	배열 요소 수	<code>\$count([1, 2, 3])</code>	3
<code>\$exists(value)</code>	값이 존재하는지 확인 (정의되지 않음)	<code>\$exists(temp.id)</code>	true 또는 false
<code>\$keys(object)</code>	객체 키 이름 가져오기	<code>\$keys(response.body)</code>	<code>["id", "name"]</code>
<code>\$lookup(object, key)</code>	키 이름으로 객체에서 값 가져오기	<code>\$lookup(response.body, 'status')</code>	status 필드 값
<code>\$each(object, func)</code>	객체의 각 키-값 페어 에 함수 적용	<code>\$each({"a":1,"b":2}, function(\$v,\$k) {\$k & '=' & \$v})</code>	<code>["a=1", "b=2"]</code>

숫자(7)

함수	설명	예제	결과
<code>\$sum(array)</code>	배열의 합계	<code>\$sum([1, 2, 3])</code>	6
<code>\$max(array)</code>	최대값	<code>\$max([10, 5, 20])</code>	20
<code>\$min(array)</code>	최소값	<code>\$min([10, 5, 20])</code>	5

함수	설명	예제	결과
<code>\$average(array)</code>	산술 평균	<code>\$average([10, 20, 30])</code>	20
<code>\$abs(number)</code>	절대값	<code>\$abs(-7)</code>	7
<code>\$floor(number)</code>	반내림	<code>\$floor(3.9)</code>	3
<code>\$round(number, precision)</code>	정밀도로 반올림	<code>\$round(3.456, 2)</code>	3.46

문자열(9)

함수	설명	예제	결과
<code>\$uppercase(string)</code>	대문자로	<code>\$uppercase('hello')</code>	"HELLO"
<code>\$lowercase(string)</code>	소문자로 설정하려면	<code>\$lowercase('Hello')</code>	"hello"
<code>\$trim(string)</code>	선행/후행 공백 제거	<code>\$trim(' hi ')</code>	"hi"
<code>\$substring(string, start, length)</code>	하위 문자열 추출(0 기반)	<code>\$substring('abcdef', 2, 3)</code>	"cde"
<code>\$contains(string, pattern)</code>	문자열에 패턴이 포함되어 있는지 확인	<code>\$contains(session.user_agent, 'CTV')</code>	true 또는 false
<code>\$match(string, pattern)</code>	정규식 패턴과 문자열 일치	<code>\$match('abc-123', /[0-9]+/)</code>	{"match": "123", ...}

함수	설명	예제	결과
<code>\$replace(string, pattern, replacement)</code>	일치하는 패턴 바꾸기	<code>\$replace('hello', 'l', 'r')</code>	"herro"
<code>\$join(array, separator)</code>	배열 요소를 문자열에 조인	<code>\$join(['a', 'b', 'c'], ',')</code>	"a,b,c"
<code>\$split(string, separator)</code>	문자열을 배열로 분할	<code>\$split('a,b,c', ',')</code>	["a", "b", "c"]

배열(7)

함수	설명	예제	결과
<code>\$append(arr1, arr2)</code>	배열 연결	<code>\$append([1, 2], [3, 4])</code>	[1, 2, 3, 4]
<code>\$reverse(array)</code>	역순	<code>\$reverse([1, 2, 3])</code>	[3, 2, 1]
<code>\$sort(array)</code>	정렬 배열	<code>\$sort([3, 1, 2])</code>	[1, 2, 3]
<code>\$distinct(array)</code>	중복값 제거	<code>\$distinct([1, 2, 2, 3])</code>	[1, 2, 3]
<code>\$map(array, func)</code>	각 요소에 함수 적용	<code>\$map([1,2,3], function(\$v) {\$v*2})</code>	[2, 4, 6]
<code>\$filter(array, func)</code>	함수가 true를 반환하는 요소 반환	<code>\$filter([1,2,3,4], function(\$v){\$v > 2})</code>	[3, 4]

함수	설명	예제	결과
<code>\$reduce(array, func, init)</code>	배열을 단일 값으로 축소	<code>\$reduce([1,2,3], function(\$prev, \$val){\$prev+\$val}, 0)</code>	6

부울(1)

함수	설명	예제	결과
<code>\$not(value)</code>	논리적 NOT	<code>\$not(false)</code>	true

임의(1)

함수	설명	예제	결과
<code>\$random()</code>	0(포함)에서 1(제외) 사이의 난수	<code>\$random() > 0.5 ? 'A' : 'B'</code>	"A" 또는 "B"

 Note

`$random()`는 각 평가에 대해 새 값을 생성합니다. 여러 출력 키에 동일한 임의 값이 필요한 경우 먼저 변수에 바인딩합니다(`$r := $random(); ...`).

날짜/시간(4)

함수	설명	예제	결과
<code>\$now()</code>	ISO 8601 문자열로서의 현재 타임스탬프	<code>\$now()</code>	"2024-01-15T12:00:00.000Z"

함수	설명	예제	결과
\$millis()	epoch 이후 밀리초 단위의 현재 타임스탬프	\$millis()	1705320000000
\$toMillis(string)	ISO 8601 문자열을 밀리초로 변환	\$toMillis('2024-01-15T12:00:00.000Z')	1705320000000
\$fromMillis(number)	밀리초를 ISO 8601 문자열로 변환	\$fromMillis(1705320000000)	"2024-01-15T12:00:00.000Z"

인코딩(6)

함수	설명	예제
\$encodeURIComponent(string)	URL 인코딩(/, ?와 같은 구조 문자 보존&)	\$encodeURIComponent('https://example.com/path?q=hello world')
\$encodeURIComponentComponent(string)	단일 구성 요소 URL 인코딩(모든 특수 문자 인코딩)	\$encodeURIComponent('a&b=c') → "a%26b%3Dc"
\$decodeUrl(string)	URL 인코딩 문자열 디코딩	\$decodeUrl('hello%20world') → "hello world"
\$decodeUrlComponent(string)	URL 인코딩 구성 요소 디코딩	\$decodeUrlComponent('a%26b') → "a&b"
\$base64encode(string)	Base64로 인코딩	\$base64encode('hello') → "aGVsbG8="
\$base64decode(string)	Base64에서 디코딩	\$base64decode('aGVsbG8=') → "hello"

i Tip

개별 쿼리 파라미터 값에 `$encodeURIComponent()`를 사용합니다. 구조를 유지하면서 전체 URL을 인코딩해야 하는 `$encodeUrl()` 경우에만 사용합니다.

일반적인 패턴

폴백 값

값이 없을 때 기본값을 제공합니다.

```
{%$exists(player_params.region) ? player_params.region : 'us-east-1'%}
```

동적 URL 구성

여러 입력에서 광고 결정 서버 URL을 빌드합니다.

```
{%'https://ads.example.com/v1/vast?ip=' & $encodeURIComponent(session.client_ip) &
'&ua=' & $encodeURIComponent(session.user_agent) & '&sid=' & session.id%}
```

HTTP_REQUEST 출력에 대한 상태 코드 확인

HTTP 실패에 대한 출력 값을 보호합니다.

```
{%response.statusCode != null and response.statusCode = 200 ? response.body.envelope :
'default-envelope'%}
```

플레이어 파라미터에서 숫자 변환

플레이어 파라미터는 문자열로 도착합니다. 산술 또는 숫자 비교 전에 변환합니다.

```
{%$number(player_params.max_duration) > 30 ? 'long' : 'short'%}
```

A Important

가 숫자가 아닌 문자열을 `$number()` 수신하면 `undefined`를 반환합니다. 파라미터가 누락되거나 유효하지 않은 `$exists()` 경우와 결합합니다(`$val :=`

```
$number(player_params.max_duration); $exists($val) and $val > 30 ?
'long' : 'short').
```

임의 트래픽 분할

를 사용하여 실험 그룹에 뷰어를 할당합니다\$random().

```
{% $random() > 0.5 ? 'https://ads.example.com/v1/vast-a' : 'https://ads.example.com/v1/
vast-b'%}
```

디바이스 유형 분류

사용자 에이전트 문자열을 기준으로 디바이스를 분류합니다.

```
{% $contains(session.user_agent, 'CTV') ? 'ctv' : $contains(session.user_agent,
'Mobile') ? 'mobile' : 'desktop'%}
```

함수 예제

이러한 예제를 살펴보기 전에 다음 사항을 숙지해야 합니다.

- [수명 주기 후크](#) - 각 수명 주기 후크에서 사용할 수 있는 입력 필드 및 출력 네임스페이스입니다.
- [함수 유형 및 구성](#) - 각 함수 유형의 작동 방식 및 구성 필드.
- [생성 및 관리](#) - 함수를 생성하고 재생 구성에 연결하는 방법입니다.

이 페이지에서는 일반적인 사용 사례에 대한 완전한 작동 함수 구성을 제공합니다. 각 예제에는 시나리오, 전체 구성, 함수 매핑, 함수 실행 시 발생하는 상황에 대한 설명이 포함되어 있습니다.

예제	시나리오 설명
the section called “예제 1: 데이터 보강”	세션 시작 시 LiveRamp 자격 증명 봉투를 가져와 플레이어 파라미터에 저장합니다.
the section called “예제 2: A/B 트래픽 분할”	A/B 테스트를 위해 두 광고 결정 서버 URLs 간에 광고 요청 트래픽을 균등하게 분할합니다.

예제	시나리오 설명
the section called “예제 3: 컨텍스트 메타데이터”	IAB 콘텐츠 분류 및 GARM 브랜드 안전 신호에 대한 Elemental Inference를 쿼리하여 컨텍스트 메타데이터로 광고 요청을 강화합니다.
the section called “예제 4: 보조 광고 서버 대체”	기본 ADS 응답이 짧을 때 보조 광고 서버에서 광고를 가져와 광고 목록에 추가합니다.

예제 1: 데이터 보강

시나리오

스트리밍 서비스는 세션 시작 시 외부 서비스(이 경우 LiveRamp)에서 대상 ID 데이터를 가져와 세션 뒷부분의 광고 대상 지정을 위한 플레이어 파라미터에 저장하려고 합니다. 자격 증명 봉투는 광고 시스템이 개인 정보를 노출하지 않고 디바이스 및 플랫폼에서 최종 사용자를 인식할 수 있도록 자격 증명 확인 서비스에서 제공하는 암호화된 식별자입니다.

구성

LiveRamp 봉투 가져오기(HTTP_REQUEST):

```
{
  "FunctionId": "fetchLiveRamp",
  "FunctionType": "HTTP_REQUEST",
  "HttpRequestConfiguration": {
    "Runtime": "JSONATA",
    "MethodType": "GET",
    "Url": "{%'https://api.example-identity.com/v2/envelope?pid=12345&iv=' & player_params.hashId%}'",
    "Headers": {
      "Authorization": "{%'Bearer ramp-token-789'%}"
    },
    "RequestTimeoutMilliseconds": 2000,
    "Output": {
      "player_params.envelope_id": "{%response.body.envelopes[0].value%}"
    }
  }
}
```

Note

를 자체 API 자격 증명 Bearer ramp-token-789으로 바꿉니다. 프로덕션의 함수 구성에 민감한 토큰을 직접 저장하지 마십시오. 토큰 교체 전략을 사용하는 것이 좋습니다.

외부 서비스의 예상 응답 시간을 RequestTimeoutMilliseconds 기준으로 설정합니다. 값이 낮을수록 지연 시간이 줄어들지만 제한 시간 오류의 가능성이 높아집니다.

함수 매핑

```
{
  "FunctionMapping": {
    "PRE_SESSION_INITIALIZATION": "fetchLiveRamp"
  }
}
```

함수 실행 시 발생하는 일

1. 뷰어가 재생 세션을 시작합니다.
2. MediaTailor는 PRE_SESSION_INITIALIZATION 수명 주기 후크를 실행하고 이를 실행합니다 fetchLiveRamp.
3. 함수는 이를 사용하여 요청 URL을 빌드 player_params.hashsetId하고 LiveRamp API를 호출합니다. URL에서 pid는 파트너 ID이고 iv는 해시된 최종 사용자 식별자입니다.
4. 출력 표현식은 응답 배열(response.body.envelopes[0].value)의 첫 번째 봉투에서 값을 추출하여 씁니다 player_params.envelope_id.

Tip

오류를 명시적으로 처리하려면 응답 데이터에 액세스 response.statusCode하기 전에 다음을 확인하세요. {%response.statusCode = 200 ? response.body.envelopes[0].value : ''%}

자세한 내용은 [HTTP 요청](#), [세션 전 초기화](#), [문제 해결 및 모니터링](#) 및 [한도](#) 부분을 참조하세요.

예제 2: A/B 트래픽 분할

시나리오

스트리밍 서비스는 CUSTOM_OUTPUT 함수를 사용하여 A/B 테스트를 위해 두 광고 결정 서버(ADS) URLs 간에 광고 요청 트래픽을 무작위로 분할하려고 합니다. 광고 요청의 절반은 v1 엔드포인트로 이동하고 절반은 v2 엔드포인트로 이동합니다.

구성

A/B 트래픽 분할(CUSTOM_OUTPUT):

```
{
  "FunctionId": "abTestAdsUrl",
  "FunctionType": "CUSTOM_OUTPUT",
  "CustomOutputConfiguration": {
    "Runtime": "JSONATA",
    "Output": {
      "adsRequest.url": "{%$random() < 0.5 ? 'https://ads.example.com/v1/decision?session=' & session.id : 'https://ads.example.com/v2/decision?session=' & session.id%}"
    }
  }
}
```

이 구성의 경우:

- FunctionId -이 함수의 고유한 이름입니다.
- FunctionType - HTTP 호출 없이 표현식을 CUSTOM_OUTPUT 평가합니다.
- Output - 두 ADS URL 중 하나를 임의로 선택하는 표현식adsRequest.url에 매핑됩니다. URLs

함수 매핑

```
{
  "FunctionMapping": {
    "PRE_ADS_REQUEST": "abTestAdsUrl"
  }
}
```

함수 실행 시 발생하는 일

1. 재생 중에 광고가 중단됩니다.
2. MediaTailor는 PRE_ADS_REQUEST 수명 주기 후크를 실행하고를 실행합니다abTestAdsUrl.
3. \$random() 함수는 0에서 1 사이의 값을 반환합니다. 값이 0.5 미만인 경우 함수는 ADS URL을 v1 엔드포인트로 설정합니다. 그렇지 않으면 URL을 v2 엔드포인트로 설정합니다.

\$random() 함수는 각 광고 시간에 평가되므로 분할은 세션이 아닌 광고 요청당입니다.

Tip

트래픽 분할 비율을 조정하려면 임계값을 변경합니다. 예를 들어는 트래픽의 80%를 첫 번째 URL로 보내고 20%를 두 번째 URL로 \$random() < 0.8 보냅니다.

Note

에 쓰면 현재 광고 시간에 대한 재생 구성에 구성된 기본 ADS URL이 adsRequest.url 재정의됩니다.

자세한 내용은 [사용자 지정 출력](#), [사전 광고 요청](#), [JSONata 표현식 참조](#) 섹션을 참조하세요.

예제 3: 컨텍스트 메타데이터

이 섹션에서는 Elemental Inference GetMetadata 응답으로 작업할 때 일반적인 패턴에 대한 JSONata 표현식을 제공합니다. AWS_SERVICE_REQUEST 함수의 Output 블록에서 이를 사용합니다.

Note

GetMetadata는 각 항목이 IAB 분류 범주 및 GARM 브랜드 안전성 평가가 포함된 분석된 샷을 나타내는 items 배열을 반환합니다. 쿼리 기간이 여러 샷에 걸쳐 있는 경우 응답에는 여러 항목이 포함됩니다. 다음 섹션의 표현식은 모든 샷에 걸쳐 집계하여 이를 처리합니다. 전체 Elemental Inference GetMetadata 응답 스키마는 Elemental Inference API 참조를 참조하세요.

쿼리 기간 빌드

AWS_SERVICE_REQUEST 함수의 Body 필드에 다음 표현식을 사용합니다.

```
{%'{"outputName": "my-contextual-output", "timeSpecification": {"ptsBased":
{"startPts": ' & $string(($exists(inference.previousBreakEndPts) and
inference.previousBreakEndPts > inference.pts - 30 * inference.timescale ?
inference.previousBreakEndPts : inference.pts - 30 * inference.timescale))
& ', "endPts": ' & $string(inference.pts + 1) & ', "timescale": ' &
$string(inference.timescale) & '}}, "parameters": {"contextualMetadata": {}}}' %}
```

이 표현식은의 최신 inference.previousBreakEndPts 및 30초 록백을 선택하여 쿼리 기간이 30 초를 초과하지 않도록 합니다. inference.previousBreakEndPts를 사용할 수 없는 경우(예: 첫 번째 광고 시간) 표현식은 기본적으로 30초 록백으로 설정됩니다.

Note

를 Elemental Inference 피드의 컨텍스트 메타데이터 출력의 이름으로 my-contextual-output 바꿉니다.

IAB 범주 IDs 추출

```
{%response.statusCode = 200 ?
$join($distinct(response.body.items.metadata.contextualMetadata.iabTaxonomy.categories.uniqueI
',') : ''%}
```

결과: "641,645,324"- 쿼리 파라미터(예:)로 전달하는 데 적합한 IAB 콘텐츠 분류 고유 IDs입니다. iab_cats=641,645,324.

IAB 범주 경로 추출

```
{%response.statusCode = 200 ?
$join($distinct(response.body.items.metadata.contextualMetadata.iabTaxonomy.categories.
($join(path, ' > '))), '|') : ''%}
```

결과: "Genres > Animation & Anime|Genres > Family/Children|Entertainment > Movies"- 사람이 읽을 수 있는 분류 경로입니다.

플래그가 지정된 GARM 범주 추출

```
{%response.statusCode = 200 ?
  $join($distinct(response.body.items.metadata.contextualMetadata.garm.suitability.categories[fl
    = true].category), ',') : ''%}
```

결과: "ILLEGAL_DRUGS_TOBACCO_ALCOHOL"- 광고 대상 지정에서 제외하거나 브랜드 안전 문제를 광고 결정 서버에 알리기 위한 범주입니다.

가장 높은 GARM 위험 수준 결정

```
{%response.statusCode = 200 and $exists(response.body.items) ?
  ($names := ['NONE','LOW','MEDIUM','HIGH']; $flagged :=
    response.body.items.metadata.contextualMetadata.garm.suitability.categories[flagged =
    true].risk; $scores := $map($flagged, function($r){ $r = 'HIGH' ? 3 : $r = 'MEDIUM' ?
    2 : $r = 'LOW' ? 1 : 0 }); $count($scores) > 0 ? $names[$max($scores)] : 'NONE') :
  ''%}
```

결과: "MEDIUM" - 분석된 모든 샷 및 GARM 범주에서 최악의 위험 수준입니다. 요청이 실패하거나 응답 본문이 누락된 경우 빈 문자열을 반환하여 누락된 데이터가 로 보고되지 않도록 합니다NONE.

브랜드 안전 확인

```
{%response.statusCode = 200 and $exists(response.body.items) ?
  $string($not($exists(response.body.items.metadata.contextualMetadata.garm.suitability.categories
    = true)))) : ''%}
```

결과: GARM 카테고리에 플래그가 지정된 "false" 경우, 콘텐츠가 브랜드 안전인 "true" 경우. 요청이 실패하거나 응답 본문이 누락된 경우 빈 문자열을 반환합니다.

Important

`$exists(response.body.items)` 가드가 필요합니다. 예를 null 들어 본
문response.body이 20,000자를 초과하거나 유효한 JSON이 아닌 경우 응답은 상태 코드
200을 반환할 수 있습니다. 가드가 없으면 GARM 분류가 수신되지 않았더라도 표현식은 콘텐
츠를 반환"true"하고 브랜드 안전으로 보고합니다.

결합된 컨텍스트 신호

```
{%response.statusCode = 200 ? $string({'categories':
  $distinct(response.body.items.metadata.contextualMetadata.iabTaxonomy.categories.
($join(path, ' > '))), 'category_ids':
  $distinct(response.body.items.metadata.contextualMetadata.iabTaxonomy.categories.uniqueId),
  'garm_flagged':
  $distinct(response.body.items.metadata.contextualMetadata.garm.suitability.categories[flagged
= true].category))} : ''%}
```

결과: 모든 컨텍스트 신호를 단일 구조화된 페이로드로 결합하는 JSON 문자열입니다.

샷별 세부 정보

```
{%response.statusCode = 200 ? $string(response.body.items.{'pts': pts, 'categories':
  metadata.contextualMetadata.iabTaxonomy.categories.($join(path, ' > ')),
  'garm_flagged': metadata.contextualMetadata.garm.suitability.categories[flagged =
true].category)} : ''%}
```

결과: PTS가 분류를 트리거한 콘텐츠를 보존하는 샷당 객체 배열입니다. 광고 결정 로직에 시간적 세분성이 필요한 경우에 사용합니다.

팁과 모범 사례

Elemental Inference 응답에 대한 출력 표현식을 작성할 때 다음 팁을 사용합니다.

- 본문을 처리하기 `response.statusCode` 전에 항상 확인하세요. 호출이 실패하면 `response.body`입니다 `null`.
- `$distinct()`를 사용하여 여러 샷에서 범주를 중복 제거하세요.
- Elemental Inference는 샷당 여러 IAB 범주와 GARM 브랜드 안전 범주 세트를 반환합니다. 현재 제한은 Elemental Inference 설명서를 참조하세요.
- 기간이 여러 샷에 걸쳐 있는 경우 응답에는 `items` 배열에 여러 항목이 포함됩니다.
- 최대 응답 크기는 20,000자입니다. 잘린 응답을 받으면 기간을 줄입니다.
- 의 다른 함수와 연결할 때 `SEQUENTIAL_EXECUTOR`와 광고 결정 서버 URL에 직접 값을 전달할 `player_params.*` 때 `temp.*` 출력 키를 사용합니다.
- 지원되는 JSONata 함수 및 연산자의 전체 목록은 섹션을 참조하세요 [the section called “JSONata 표현식 참조”](#).

사전 조건 및 리소스 정책을 포함한 전체 설정 가이드는 섹션을 참조하세요 [Elemental Inference 통합](#). 함수 유형 참조는 섹션을 참조하세요 [AWS 서비스 요청](#).

예제 4: 보조 광고 서버 대체

시나리오

기본 ADS가 광고 시간보다 적은 수의 광고를 반환하는 경우가 있습니다. 이 경우 보조 광고 서버에서 추가 광고를 가져와 광고 목록에 추가하면 추가 광고가 남은 시간을 채울 수 있습니다.

이 예제에서는 POST_ADS_RESPONSE 수명 주기 후크를 사용합니다. VAST_REQUEST 함수는 보조 광고를 가져오고 프로세스를 SEQUENTIAL_EXECUTOR 오케스트레이션합니다. 실행기의 출력 블록은 가져온 광고를 광고 목록에 추가하고 기본 응답에 이미 충분한 광고가 있는 경우 보조 가져오기를 완전히 RunCondition 건너뛵니다.

함수 구성

가져오기 함수(**fetchBackupAds**):

```
{
  "FunctionId": "fetchBackupAds",
  "FunctionType": "VAST_REQUEST",
  "VastRequestConfiguration": {
    "Runtime": "JSONATA",
    "MethodType": "GET",
    "Url": "{%'https://backup-ads.example.com/vast?sid=' & session.uuid%}'",
    "Headers": {
      "Accept": "application/xml"
    },
    "RequestTimeoutMilliseconds": 1000,
    "Output": {
      "temp.backupAds": "{%response.parsedAds%}"
    }
  }
}
```

시퀀스(**backupAdsFallback**):

```
{
  "FunctionId": "backupAdsFallback",
  "FunctionType": "SEQUENTIAL_EXECUTOR",
```



```

    "SequentialExecutorConfiguration": {
      "Runtime": "JSONATA",
      "TimeoutMilliseconds": 2000,
      "FunctionList": [
        { "FunctionId": "fetchBackupAds", "RunCondition": "{%
$count(adsResponse.ads) < 3%}" }
      ],
      "Output": {
        "adsResponse.ads": "{% $count(adsResponse.ads) < 3 ?
$append(adsResponse.ads, temp.backupAds)[] : adsResponse.ads[%]}"
      }
    }
  }
}

```

함수 매핑

```

{
  "FunctionMapping": {
    "POST_ADS_RESPONSE": "backupAdsFallback"
  }
}

```

실행 시간 작동

1. MediaTailor는 기본 ADS 응답을 수신하고 구문 분석한 다음 POST_ADS_RESPONSE 수명 주기 후크를 실행합니다.
2. 기본 응답에 3개 미만의 광고가 포함된 경우는 보조 광고 서버를 RunCondition 평가하고 true fetchBackupAds 호출합니다. 그런 다음 MediaTailor는 응답을 VAST로 구문 분석하고 래퍼 리더렉션을 확인하며 구문 분석된 광고를 기록합니다 temp.backupAds.
3. 기본 응답에 이미 3개 이상의 광고가 있는 경우 가져오기는 건너뛰고 설정 temp.backupAds되지 않습니다.
4. 시퀀스의 출력 블록은 결합된 광고 목록에 씁니다 adsResponse.ads. 가져오기를 건너뛰면 조건은 else브랜치를 가져와서 원래 광고를 변경 없이 반환합니다. 보조 광고는 동일한 후크 호출에서 VAST_REQUEST 함수에 의해 구문 분석되었으므로 MediaTailor는 광고 선택 전에 추가된 각 광고의 전체 VAST 데이터를 복원합니다.
5. 광고 선택은 결합된 목록으로 진행됩니다. 보조 가져오기에 실패한 경우 temp.backupAds는 빈 배열이며 원래 광고 목록은 변경되지 않고 사용됩니다.

Note

출력 표현식[]의 후행은 필수입니다. 배열 강제가 없으면 0개 또는 1개의 광고를 생성하는 표현식은 배열을 생성하지 않으며 출력이 삭제되거나 잘못 적용됩니다. 자세한 내용은 [광고 후 응답](#) 단원을 참조하십시오.

이 예제에서 사용되는 필드에 대한 자세한 내용은 [VAST 요청](#) 및 섹션을 참조하세요 [광고 후 응답](#).

함수 문제 해결 및 모니터링

이 페이지에서는 일반적인 함수 오류를 진단하고 프로덕션 환경에서 함수 성능을 모니터링할 수 있습니다. 문제 해결 섹션은 증상별로 구성되어 있습니다. 관찰한 내용부터 시작한 다음 원인을 따르고 수정합니다.

모니터링

CloudWatch Metrics

MediaTailor는 함수 실행에 대한 지표를 Amazon CloudWatch에 게시합니다. 옵트인이 필요하지 않습니다.

후크 수준 지표 - 수명 주기 후크 실행당 하나의 데이터 포인트:

지표	설명	측정 기준
PreSessionInitHook. Invocations	후크 실행 수	ConfigurationName
PreSessionInitHook. Errors	후크 오류 수	ConfigurationName
PreSessionInitHook. Latency	후크 실행 시간(ms)	ConfigurationName
PreAdsRequestHook. Invocations	후크 실행 수	ConfigurationName
PreAdsRequestHook. Errors	후크 오류 수	ConfigurationName

지표	설명	측정 기준
PreAdsRequestHook.Latency	후크 실행 시간(ms)	ConfigurationName
PostAdsResponseHook.Invocations	후크 실행 수	ConfigurationName
PostAdsResponseHook.Errors	후크 오류 수	ConfigurationName
PostAdsResponseHook.Latency	후크 실행 시간(ms)	ConfigurationName
PreManifestInsertionHook.Invocations	후크 실행 수	ConfigurationName
PreManifestInsertionHook.Errors	후크 오류 수	ConfigurationName
PreManifestInsertionHook.Latency	후크 실행 시간(ms)	ConfigurationName

함수 수준 지표 - 개별 함수 실행당 하나의 데이터 포인트:

지표	설명	측정 기준
Function.Invocations	함수 실행 수	ConfigurationName , FunctionId , FunctionType , HookType
Function.Errors	함수 오류 수	ConfigurationName , FunctionId , FunctionType , HookType
Function.Latency	함수 실행 시간(ms)	ConfigurationName , FunctionId , FunctionType , HookType

경보 설정 및 이러한 지표 작업에 대한 자세한 내용은 섹션을 참조하세요 [Amazon CloudWatch 지표 AWS Elemental MediaTailor 를 사용한 모니터링](#).

로그 이벤트

MediaTailor는 함수 실행을 위해 로그 이벤트를 내보냅니다. 오류 이벤트는 기본적으로 내보내집니다. 완료된 이벤트와 요약 이벤트는 옵트인됩니다.

이벤트 유형	기본 / 옵트인	로그 그룹:	설명
PRE_SESSION_INIT_HOOK_SUMMARY	옵트인	매니페스트 로그	후크 실행 요약(성공/오류)
PRE_SESSION_INIT_HOOK_ERROR	기본값	매니페스트 로그	errorType 및 원인과 함께 후크 실패
PRE_SESSION_INIT_FUNCTION_COMPLETED	옵트인	매니페스트 로그	입력/출력으로 완료된 개별 함수
PRE_SESSION_INIT_FUNCTION_ERROR	기본값	매니페스트 로그	개별 함수 실패
PRE_ADS_REQUEST_HOOK_SUMMARY	옵트인	ADS 상호 작용 로그	후크 실행 요약(성공/오류)
PRE_ADS_REQUEST_HOOK_ERROR	기본값	ADS 상호 작용 로그	errorType 및 원인과 함께 후크 실패
PRE_ADS_REQUEST_FUNCTION_COMPLETED	옵트인	ADS 상호 작용 로그	입력/출력으로 완료된 개별 함수

이벤트 유형	기본 / 옵트인	로그 그룹:	설명
PRE_ADS_REQUEST_FUNCTION_ERROR	기본값	ADS 상호 작용 로그	개별 함수 실패
POST_ADS_RESPONSE_HOOK_SUMMARY	옵트인	ADS 상호 작용 로그	후크 실행 요약(성공/오류)
POST_ADS_RESPONSE_HOOK_ERROR	기본값	ADS 상호 작용 로그	errorType 및 원인과 함께 후크 실패
POST_ADS_RESPONSE_FUNCTION_COMPLETED	옵트인	ADS 상호 작용 로그	입력/출력으로 완료된 개별 함수
POST_ADS_RESPONSE_FUNCTION_ERROR	기본값	ADS 상호 작용 로그	개별 함수 실패
PRE_MANIFEST_INSERTION_HOOK_SUMMARY	옵트인	ADS 상호 작용 로그	후크 실행 요약(성공/오류)
PRE_MANIFEST_INSERTION_HOOK_ERROR	기본값	ADS 상호 작용 로그	errorType 및 원인과 함께 후크 실패
PRE_MANIFEST_INSERTION_FUNCTION_COMPLETED	옵트인	ADS 상호 작용 로그	입력/출력으로 완료된 개별 함수

이벤트 유형	기본 / 옵트인	로그 그룹:	설명
PRE_MANIFEST_INSERTION_FUNCTION_ERROR	기본값	ADS 상호 작용 로그	개별 함수 실패

옵트인 로그 이벤트를 활성화하려면 섹션을 참조하세요 [Amazon CloudWatch 지표 AWS Elemental MediaTailor 를 사용한 모니터링](#).

eventId 필드를 사용하여 동일한 실행에 대한 후크 수준 이벤트와 함수 수준 이벤트를 상호 연관시킵니다.

다음 Amazon CloudWatch Logs Insights 쿼리는 함수 오류 이벤트를 로 필터링eventId하여 단일 실행을 추적합니다.

```
fields @timestamp, eventType, functionId, errorType, cause
| filter eventId = "5dc6f040-0f72-4e8c-a64e-25eeef62708c"
| sort @timestamp asc
```

문제 해결

함수가 실패하면 MediaTailor는 오류 이벤트에 errorType 필드를 기록합니다. 이 필드를 사용하여 실패 클래스를 식별합니다.

오류 유형	설명
SYNTAX_ERROR	표현식이 컴파일되지 않았거나 런타임 유형 오류가 발생했습니다.
RESOURCE_LIMIT_ERROR	표현식이 CPU 시간, 메모리 또는 스택 깊이 제한을 초과함
RESTRICTION_ERROR	표현식이 차단된 함수를 사용했거나 입력 페이로드 크기 제한을 초과했습니다.
TIMEOUT_ERROR	함수 실행이 시간 제한을 초과했습니다.

오류 유형	설명
VALIDATION_ERROR	출력 경로는 현재 후크의 범위에서 쓸 수 없는 필드를 대상으로 합니다.
INTERNAL_ERROR	함수와 관련이 없는 인프라 장애

항목은 증상별로 구성되며 해당하는 경우 이러한 오류 유형을 참조합니다.

표현식이 여기치 않게 null을 반환합니다.

증상: 채워질 것으로 예상되는 출력 값이 플레이어 파라미터에서 null 이거나 누락되었습니다.

가능한 원인:

원인	식별 방법	수정
이 수명 주기 후크에는 입력 필드가 없습니다.	PRE_SESSION_INITIALIZATION 함수adsRequest.url 에서를 참조했습니다. 세션 시작 시 ADS 요청 데이터를 사용할 수 없습니다.	함수를 PRE_ADS_REQUEST 수명 주기 후크로 이동하거나 다른 입력 필드를 사용합니다. 수명 주기 후크 을(를) 참조하세요.
세션 데이터에서 입력 필드가 누락되었습니다.	를 참조player_params.campaign_id 했지만 플레이어가 세션 초기화 시 해당 파라미터를 전달하지 않았습니다.	\$exists() 를 사용하여 액세스하기 전에 확인합니다{%\$exists(player_params.campaign_id) ? player_params.campaign_id : 'default'%} .
플레이어 파라미터 또는 ADS 요청에 객체 또는 배열을 작성했습니다.	이러한 네임스페이스는 문자열, 숫자 및 부울만 허용합니다. 객체와 배열이 필터링됩니다.	복잡한 데이터를 저장temp.*하고 후속 단계에서 문자열, 숫자 또는 부울을 추출합니다.

RESOURCE_LIMIT_ERROR: 스택 오버플로

증상: `errorType: "RESOURCE_LIMIT_ERROR"` 및에서 함수가 실패합니다 `cause: "Stack overflow error"`.

원인: 표현식이 레벨 100의 최대 스택 깊이를 초과했습니다. 이는 일반적으로 깊이 중첩된 조건부(if/then/else) 표현식 또는 복잡한 변수 할당에서 발생합니다.

즉, 표현식의 중첩 수준이 너무 많아 MediaTailor에서 처리할 수 없습니다.

수정: 표현식을 간소화합니다. 복잡한 로직을 순차적 실행기에서 여러 출력 항목 또는 여러 단계로 나눕니다.

RESOURCE_LIMIT_ERROR: CPU 제한 시간

증상: `errorType: "RESOURCE_LIMIT_ERROR"` 및에서 함수가 실패합니다 `cause: "Expression evaluation timeout: Check for infinite loop"`.

원인: 표현식이 100ms CPU 시간 제한을 초과했습니다. 이는 대규모 데이터 구조에 대해 비용이 많이 드는 계산을 수행하는 표현식에서 발생할 수 있습니다.

수정: 표현식의 복잡성을 줄입니다. 큰 배열을 처리하는 경우 해당 로직을 외부 서비스로 이동하고 HTTP_REQUEST 함수를 사용하여 호출하는 것이 좋습니다.

RESTRICTION_ERROR: 함수가 허용되지 않음

증상: `errorType: "RESTRICTION_ERROR"` 및에서 함수가 실패합니다 `cause: "Function '<name>' is not allowed"`.

원인: 표현식은 허용되는 44개 함수 목록에 없는 JSONata 함수를 호출합니다. 일반적인 예로는 \$eval, \$assert, \$error, 등이 있습니다 \$sift.

수정: `cause` 필드에 차단된 함수 이름이 있는지 확인합니다. 이를 허용된 대안으로 바꿉니다. 허용되는 함수 44개의 전체 목록은 섹션을 참조 [JSONata 표현식 참조](#) 하세요.

일반적으로 사용되는 허용되는 함수는 \$string, \$number, \$substring, 및 \$contains입니다 \$encodeURIComponent.

RESTRICTION_ERROR: 표현식이 너무 길다.

증상: 함수를 사용하여 생성하거나 업데이트하지 못합니다 `cause: "Expression length <actual> exceeds limit <limit>"`.

원인: 단일 표현식이 1,000자를 초과합니다.

수정: 표현식을 더 작은 부분으로 나눕니다. 여러 출력 항목을 사용하거나 순차적 실행기에서 여러 단계로 로직을 분할합니다. 변수 바인딩(=)을 사용하여 긴 하위 표현식이 반복되지 않도록 합니다.

HTTP 오류: 상태 코드가 null입니다.

증상: HTTP_REQUEST 함수의 출력에서는 `response.statusCode`입니다null.

원인: 외부 API에 연결할 수 없거나, 연결 시간이 초과되었거나, 네트워크 오류가 발생했습니다. 이 경우 MediaTailor는 `response.statusCode`를 null, `response.body`를 null, `response.text`로 설정합니다"Internal Error".

수정: 응답 데이터에 액세스`response.statusCode`하기 전에 항상 확인하세요.

```
{%response.statusCode = 200 ? response.body.value : 'default'%}
```

이 표현식은 HTTP 호출이 200 상태 코드를 반환했는지 확인합니다. 그렇다면 응답 값을 사용합니다. 그렇지 않으면 기본값으로 돌아갑니다.

이 문제가 자주 발생하면 외부 API가 정상인지 확인합니다. API가 느린 `RequestTimeoutMilliseconds` 경우 늘리고 빠른 실패를 원하는 경우 줄이는 것이 좋습니다.

HTTP 오류: 응답 본문이 null임

증상: `response.statusCode`는 200이지만 `response.body`입니다null.

원인: 응답 본문이 유효한 JSON이 아니거나 20,000자를 초과합니다. MediaTailor는 최대 20,000자의 JSON 응답만에 구문 분석합니다`response.body`.

수정:를 대체`response.text`로 사용합니다. `response.text` 필드에는 20,000자로 잘린 원시 응답 본문이 포함됩니다.

```
{%response.statusCode = 200 ? ($exists(response.body.id) ? response.body.id :  
  $substring(response.text, 0, 100)) : 'error'%}
```

필요한 데이터가 20,000자 제한을 초과하는 경우 외부 API에 더 작은 응답을 반환하도록 요청하는 것이 좋습니다(예: 특정 필드 요청).

HTTP 오류: URL 검증 실패

증상: URL의 형식이 잘못되었거나 잘못된 체계를 사용하거나 최대 길이를 초과했다는 메시지와 함께 함수가 런타임에 실패합니다.

가능한 원인:

원인	수정
URL은 http 또는 https를 사용하지 않습니다.	URL 표현식이 http:// 또는 https://로 시작하는 URL을 생성하는지 확인합니다.
표현식 평가 후 URL이 2,048자를 초과합니다.	URL을 줄입니다. POST 메서드를 사용하여 큰 파라미터 값을 요청 본문으로 이동합니다.
URL의 형식이 잘못되었습니다(유효한 URI가 아닙니다).	표현식에 누락된 문자나 추가 문자가 있는지 확인합니다. 특수 문자를 포함할 수 있는 쿼리 파라미터 값에 <code>\$encodeURIComponent()</code> 를 사용합니다.

VALIDATION_ERROR

증상: 함수가에서 실패합니다. `errorType: "VALIDATION_ERROR"`. 이 오류는 작성 시(함수를 생성하거나 업데이트할 때) 또는 실행 시(세션 중에 함수가 실행될 때) 발생할 수 있습니다.

가능한 원인:

원인	예제	수정
출력 키는 현재 후크에서 쓸 수 없는 네임스페이스를 대상으로 합니다.	에 연결된 함수 <code>adsRequest.url</code> 에서에 쓰기 <code>PRE_SESSION_INITIALIZATION</code> .	수명 주기 후크에서 허용되는 출력 네임스페이스를 확인합니다. <code>PRE_SESSION_INITIALIZATION</code> 만 허용합니다. <code>player_params.*</code> . 함수를 올바른 후크로 이동하거나 출력 키를 변경합니다. 수명 주기 후크 을(를) 참조하세요.
출력 키는 잘못된 문자를 사용합니다.	와 같은 출력 키 <code>player_params.device_type</code> (공백 포함). 문자, 숫자, 밑줄 및 하이픈만 허용됩니다.	유효한 문자만 사용하도록 출력 키의 이름을 바꿉니다. 예를 들어 <code>player_params.device_type</code> 대신 <code>player_params.device_type</code> 를 사용합니다.

원인	예제	수정
출력 키는 유효한 접두사로 시작하지 않습니다.	<code>player_params.myValue</code> 또는 <code>custom.myValue</code> 대신과 같은 출력 키입니다 <code>temp.myValue</code> .	유효한 출력 접두사 <code>player_params.*</code> , 또는 <code>temp.*</code> 를 사용합니다 <code>adsRequest.*</code> .
JSONata 표현식에 구문 오류가 있습니다.	누락된 종료 따옴표 또는 불완전한 조건부: <code>{%session.id & %}</code> .	표현식에서 누락된 따옴표, 일치하지 않는 괄호 또는 <code>??</code> 또는와 같은 지원되지 않는 연산자를 검토합니다?:.
HTTP_REQUEST 함수에 필수 필드가 누락되었습니다.	URL 필드가 비어 있거나 메시드가 지정되지 않았습니다.	URL 및 메서드 필드가 설정되어 있는지 확인합니다. 메서드는 GET 또는 여야 합니다 POST.
표현식으로 빌드된 URL이 잘못되었습니다.	평가된 URL은와 같은 지원되지 않는 체계를 사용하거나 <code>ftp://</code> , 2,048자를 초과하거나, 형식이 잘못되었습니다.	URL 표현식이 유효한 <code>http://</code> 또는 <code>https://</code> URL을 생성하는지 확인합니다. 특수 문자를 포함할 수 있는 쿼리 파라미터 값에 <code>\$encodeURIComponent()</code> 를 사용합니다.
HTTP 헤더에 잘못된 문자가 포함되어 있거나 제한된 이름을 사용합니다.	헤더 값에 줄 바꿈이 포함되거나 헤더 이름이 <code>host</code> 또는 <code>transfer-encoding</code> .	헤더 값에서 잘못된 문자를 제거합니다. 제한된 헤더 이름을 사용하지 마세요. 헤더 제한은 섹션을 참조 HTTP 요청 하세요.

오류 로그 이벤트의 `cause` 필드를 확인합니다. 즉, 검증에 실패한 필드 또는 표현식을 식별합니다.

INTERNAL_ERROR

증상: 함수가에서 실패합니다 `errorType: "INTERNAL_ERROR"`.

원인: 함수 구성과 관련이 없는 인프라 장애가 발생했습니다.

수정: 요청을 다시 시도합니다. 오류가 지속되면 AWS Support에 문의하십시오.

광고 목록 변경에는 영향이 없습니다.

증상: 함수가 POST_ADS_RESPONSE 또는에서 오류 없이 실행PRE_MANIFEST_INSERTION되지만 스트림의 광고는 변경되지 않습니다. 필터, 재정렬 또는 수정은 무시되는 것으로 보입니다.

가능한 원인:

원인	식별 방법	수정
필터 표현식은 배열 강제 없이 0개 또는 1개의 결과를 생성했으므로 출력 키가 생략되거나 잘못 적용되었습니다.	표현식은 후행 [] 또는 둘러싸는 배열 생성자가 adsResponse.ads[adSystem != 'BLOCKED'] 없는 등의 조건자 필터입니다. 일치 항목 하나가 있으면 단일 객체가 생성되고 일치 항목이 0이면 값이 생성되지 않으며 출력 키가 생략되어 시스템에서 "변경 없음"으로 처리됩니다.	항상 필터 결과를 또는 배열로 강제 적용합니다adsResponse.ads[adSystem != 'BLOCKED'] [] [adsResponse.ads[predicate]] .
광고 시간은 수정할 수 없습니다(PRE_MANIFEST_INSERTION).	광고 시간의 mutable 입력 필드는입니다false. 이러한 광고 시간에 대한 수정 사항은 삭제됩니다.	표현식을 mutable 확인하고 인 광고 시간만 수정합니다true.
공유 후크 예산이 소진되었거나 입력이 크기 제한을 초과했기 때문에 후크를 건너뛰었습니다.	해당 요청에 대한 ADS 상호 작용 로그에 후크에 대한 HOOK_SUMMARY 이벤트가 없습니다.	이전 후크에 소요되는 시간을 줄이거나 입력 크기를 줄입니다. 한도 을(를) 참조하세요.

함수가 실제로 반환한 내용을 확인하려면 후크의 FUNCTION_COMPLETED 이벤트 유형을 선택하고 ADS 상호 작용 로그에서 함수 출력을 검사합니다.

주입된 광고가 매니페스트에 표시되지 않음

증상: PRE_MANIFEST_INSERTION 함수가 삽입된 광고를 반환하고 성공적으로 완료되지만 광고가 렌더링된 매니페스트에 없습니다.

주입 검사는 함수가 반환된 후 실행되므로 주입이 삭제된 경우에도 함수의 로그 이벤트가 성공을 보고합니다. 다음 원인을 순서대로 확인합니다.

원인	수정
<code>creativeUrl</code> 크리에이티브가 아직 트랜스코딩되지 않았습니다. MediaTailor는 대기하지 않고 현재 광고 시간에 대한 주입을 삭제하지만 시도는 트랜스코딩을 시작합니다.	트랜스코딩이 완료된 후 나중에 광고 시간에 동일한 URL을 삽입하는 작업이 성공합니다. 즉시 삽입하려면 이미 트랜스코딩된 크리에이티브 MediaTailor(이전 광고 시간의 크리에이티브 또는 트랜스코딩과 관련이 없는 <code>skippedAds[].creativeUrl</code> 값)를 사용하거나 MediaTailorreason가 미디어 선택 및 트랜스코딩 등록을 처리하도록 동일한 체인의 <code>VAST_REQUEST</code> 함수와 <code>vastAdId</code> 함께를 주입합니다.
항목에는 사용 가능한 <code>creativeUrl</code> 도 없고 동일한 후크 호출에서 <code>VAST_REQUEST</code> 호출에 의해 구문 분석된 광고와 <code>vastAdId</code> 일치하는도 없습니다.	주입과 동일한 함수 체인 <code>VAST_REQUEST</code> 에서 호출하고 구문 분석된 광고의 <code>adId</code> 값을 사용하여 주입합니다 <code>vastAdId</code> .
호출이 광고 10개의 주입 한도를 초과했습니다.	한도를 초과하는 주입은 삭제됩니다. 호출당 주입된 광고 수를 줄입니다.
대상 광고 중단은 수정할 수 없습니다 (<code>mutable: false</code>).	<code>mutable</code> 가 인 광고 시간에만 주입합니다 <code>true</code> .
삽입된 광고의 변환이 스트림의 변형과 일치하지 않거나 추가된 지속 시간이 광고 시간을 초과합니다.	주입된 광고는 ADS의 광고와 동일한 변형 일치 및 채우기 정책을 거칩니다. 크리에이티브의 변환이 스트림과 일치하는지 확인하고 광고 시간 내에 총 광고 기간을 유지합니다.

함수 제한

이 페이지에는 함수에 적용되는 제한이 나열되어 있습니다. 이러한 제한은 작성 시간(함수를 생성하거나 업데이트할 때) 및 런타임(함수가 실행될 때)에 적용됩니다.

후크 실행 제한

후크	한도	값
PRE_SESSION_INITIALIZATION	총 제한 시간	2,000ms
PRE_SESSION_INITIALIZATION	총 player_params 크기	1,000자
PRE_ADS_REQUEST	총 제한 시간	2,000ms
POST_ADS_RESPONSE	총 제한 시간	2,000ms
POST_ADS_RESPONSE	입력 페이로드 크기	128KB
POST_ADS_RESPONSE	출력 크기	256KB
PRE_MANIFEST_INSERTION	총 제한 시간	2,000ms
PRE_MANIFEST_INSERTION	입력 페이로드 크기	128KB
PRE_MANIFEST_INSERTION	출력 크기	256KB
PRE_MANIFEST_INSERTION	호출당 주입된 광고	10
요청 경로 후크 결합(요청당)	공유 실행 예산	2,000ms

이러한 제한 시간은 모든 함수 단계 및 HTTP 호출을 포함하여 전체 후크 실행을 포함합니다. 후크가 제한 시간을 초과하면 MediaTailor는 모든 출력을 삭제하고 연결된 함수가 없는 것처럼 진행합니다.

각 후크의 자체 제한 시간 외에도 단일 요청 내에서 실행되는

PRE_ADS_REQUEST, POST_ADS_RESPONSE, 및 PRE_MANIFEST_INSERTION 후크는 2,000ms의 통합 실행 예산을 공유합니다. 이전 후크에서 소요된 시간은 동일한 요청의 이후 후크에 사용할 수 있는 예산을 줄이고 후크의 유효 제한 시간은 자체 제한 시간과 나머지 예산 중 작은 값입니다. 후크가 시작되

기 전에 예산이 소진되면 MediaTailor는 해당 후크를 완전히 건너뛰고 요청을 계속 처리합니다. 세션 초기화 중에 PRE_SESSION_INITIALIZATION 실행되며 결합된 예산의 일부가 아닙니다.

입력 페이로드 크기 제한은 MediaTailor가 POST_ADS_RESPONSE 및 PRE_MANIFEST_INSERTION 후크에 대해 빌드하는 직렬화된 JSON 입력에 적용됩니다. 입력이 128KB를 초과하면 MediaTailor는 해당 요청에 대한 후크 호출을 건너뜁니다. 출력 크기 제한은 함수가 후크의 쓰기 가능한 네임스페이스에 쓸 수 있는 직렬화된 총 크기를 제한합니다. 256KB를 초과하는 출력은 삭제됩니다.

개별 함수 제한 시간(예: HTTP_REQUEST 함수RequestTimeoutMilliseconds의 경우)은 후크의 총 제한 시간 내에 있어야 합니다. 예를 들어 후크 제한 시간이 2,000ms이고 HTTP_REQUEST 함수가 2,000msRequestTimeoutMilliseconds로 설정된 경우 HTTP 호출 후처리에 처리가 발생하는 경우 함수가 완료되기 전에 시간 초과될 수 있습니다.

player_params 총 크기에는 모든 출력 키 이름 및 값의 결합된 문자 수가 포함됩니다. 예를 들어 맞출 출력하면 player_params.deviceType = "mobile" player_params.region = "us-east-1" deviceType, region, mobile 및 1,000자 합계에 us-east-1 포함됩니다.

표현식 제한

한도	값	에 적용됨
표현식당 CPU 시간	100ms	런타임
스택 깊이	100	런타임
표현식 길이	1,000자	작성

표현식이 런타임 시 제한을 초과하면 MediaTailor는 표현식을 중지하고 오류를 기록합니다. 특정 오류 유형에 대한 자세한 내용은 섹션을 참조하세요 [문제 해결 및 모니터링](#).

함수 구성 제한

한도	값
순차 실행기당 단계	1~10
동시 실행기당 하위 항목	1~10

한도	값
최대 동시성(동시 실행기)	1~2
기본 제한 시간(동시 실행기)	2,000ms
동시 실행기당 고유한 네임스페이스	필수(작성 시 검증됨)
최대 중첩 깊이	2
후크당 총 함수 실행 수	20
함수당 출력 항목	20
순환 참조	허용되지 않음

이러한 제한은 작성 시 적용됩니다.

HTTP 요청 제한

한도	값
URL 표현식 길이	25,000자
URL 길이(평가 후)	2,048자
본문 표현식 길이	100,000자
요청 본문 크기(평가 후)	64KB
헤더 수	50
헤더 이름 길이	256자
헤더 값 길이	8,192자
요청 제한 시간	100~2,000ms(를 통해 고객 구성 가능RequestTimeoutMilliseconds)
허용된 URL 체계	https, http

한도	값
제한된 헤더	Host, Transfer-Encoding , Content-Length , Connection

표현식 길이 제한 및 헤더 제한은 함수를 생성하거나 업데이트할 때 작성 시 적용됩니다. MediaTailor가 표현식을 평가한 후 런타임 시 평가된 URL 및 본문 크기 제한, 요청 제한 시간 및 제한된 헤더가 적용됩니다. 작성 시 수락되는 URL 표현식은 런타임 시 평가된 URL 제한을 여전히 초과할 수 있으며,이 경우에도 요청이 전송되지 않습니다.

함수가 제한된 헤더를 설정하는 경우 MediaTailor는 작성 시 함수 구성을 수락하지만 실행 중에 HTTP 요청이 전송될 때 헤더를 삭제합니다.

응답 제한

한도	값
response.body 최대 크기	20,000자
response.text 최대 길이	20,000자

이러한 제한은 런타임에 적용됩니다.

출력 제한

한도	값
출력 값 최대 길이(키당)	1,000자

이러한 제한은 런타임에 적용됩니다.

허용된 JSONata 함수

허용되는 함수의 전체 목록은 섹션을 참조하세요 [JSONata 표현식 참조](#).

이러한 제한과 관련된 오류를 진단하는 데 도움이 필요하면 섹션을 참조하세요 [문제 해결 및 모니터링](#).

MediaTailor 수율 최적화 작업

수율 최적화는 라이브스트림에 대한 서버 측 광고 삽입(SSAI) 중에 Amazon Ads 수요로 미사용 광고 인벤토리를 자동으로 수익화하는 기능입니다. 출시 시 Amazon Publisher Services(APS) 스트리밍 TV 프로그램에 등록된 게시자는 수율 최적화를 사용할 수 있으며, 시간이 지남에 따라 추가 수요 소스가 제공됩니다. Yield Optimization은 추가 인프라 없이를 활성화하기 위해 채워지지 않은 광고 시간을 0의 비용으로 수익으로 전환합니다.

다음과 같은 경우 수율 최적화를 사용합니다.

- 광고 수익을 극대화합니다. 기본 광고 결정 서버(ADS)가 채워지지 않은 상태로 두는 남은 인벤토리를 채워 Amazon DSP 및 후원 TV를 통해 프로그래밍 방식의 수요로 인한 수익을 캡처합니다.
- 최종 사용자 경험을 개선합니다. 반복적인 주택 광고 또는 슬레이트 콘텐츠를 광고 시간의 채워지지 않은 부분에서 관련 프로그래밍 방식의 광고로 바꿉니다.

수율 최적화가 MediaTailor 워크플로에 어떻게 적용되는지

재생 중에 최종 사용자가 광고 시간을 발견하면 MediaTailor는 다음 순서를 따릅니다.

1. MediaTailor는 오리진 콘텐츠에서 광고 중단 신호(SCTE 마커)를 감지합니다.
2. MediaTailor는 기본 광고 결정 서버(ADS)에서 광고를 요청합니다.
3. 기본 ADS는 광고를 반환합니다. MediaTailor는 채워지지 않은 나머지 기간을 계산합니다.
4. 수율 최적화가 구성되어 있고 채워지지 않은 지속 시간이 최소 임계값인 15초를 초과하며 충분한 개입 시간이 남아 있는 경우 MediaTailor는 APS에 OpenRTB 입찰 요청을 보냅니다. 수율 최적화는 OpenRTB 입찰 요청을 통해 VAST 광고 범주 신호를 전달하여 브랜드 안전을 지원합니다. 이렇게 하면 경쟁하거나 제한된 광고 범주가 응답에서 제외되어 충돌하는 광고가 같은 시간에 나타나는 것을 방지할 수 있습니다.
5. APS는 프로그래밍 방식의 광고를 반환합니다. MediaTailor는 이를 기본 광고와 병합하고 중복 제거를 적용하여 동일한 광고가 휴식 시간 내에 두 번 이상 삽입되지 않도록 합니다.
6. MediaTailor는 완전한 개인 맞춤형 광고 시간을 비디오 플레이어에 반환합니다.

APS 요청이 실패하거나 시간이 초과되면 MediaTailor는 기본 ADS 광고만 진행합니다. 수율 최적화는 항상 열리지 않습니다. 최종 사용자의 재생 환경을 차단하거나 지연시키지 않습니다.

사전 조건

수율 최적화를 활성화하기 전에 다음을 수행해야 합니다.

- 활성 Amazon Publisher Services(APS) 스트리밍 TV 게시자 계정이 있어야 수율 최적화를 활성화할 수 있습니다. APS 게시자 ID를 사용하여 MediaTailor 콘솔 또는 API를 통해 수율 최적화를 활성화합니다. [APS와 통합](#)을(를) 참조하세요.
- 기존 MediaTailor 재생 구성이 있어야 합니다.
- SCTE 광고 브레이크 마커가 있는 라이브 스트리밍 콘텐츠 소스가 있어야 합니다.

지원되는 콘텐츠 유형

수율 최적화는 현재 다음을 지원합니다.

- HLS 및 DASH 미드를 광고 브레이크를 사용한 라이브 스트리밍(SCTE 브레이크 기간을 사용할 수 있어야 함).

현재 릴리스에서는 다음이 지원되지 않습니다.

- Video-on-Demand(VOD) 콘텐츠
- 광고 시간 사전 롤링
- 워크플로 미리 가져오기
- 오버레이 광고 요청

Tip

수율 최적화를 처음 사용하는 경우 로 시작한 다음에 따라 APS에 [빠른 시작 설명서](#) 등록합니다. [APS와 통합](#). 타이밍 및 페일 오픈 동작을 이해하려면 섹션을 참조하세요. [수율 최적화 작동 방식](#).

주제

- [수율 최적화 빠른 시작 안내서](#)
- [APS와 통합](#)
- [수율 최적화 작동 방식](#)

- [입찰 구성 및 템플릿 지정](#)
- [ORTB 필드 참조](#)
- [템플릿 예제](#)
- [모범 사례](#)
- [문제 해결 및 모니터링](#)

수율 최적화 빠른 시작 안내서

이 가이드에서는 재생 구성에서 수율 최적화를 활성화하고 작동하는지 확인하는 방법을 안내합니다.

목표: 이 가이드가 끝나면 APS의 프로그래밍 방식 광고로 채워지지 않은 광고 인벤토리를 자동으로 채우는 재생 구성을 갖게 됩니다.

사전 조건

- 라이브 콘텐츠 소스가 있는 기존 MediaTailor 재생 구성입니다.
- APS 게시자 ID입니다. 없는 [APS와 통합](#) 경우를 참조하세요.

1단계: 콘솔에서 수율 최적화 활성화

1. [MediaTailor 콘솔](#)을 엽니다.
2. 왼쪽 탐색 창에서 구성을 선택합니다.
3. 업데이트할 재생 구성을 선택합니다.
4. 편집을 선택합니다.
5. 수율 최적화 설정 섹션을 확장합니다(시작 시 새 배지로 표시됨). 섹션 설명에는 "게시자 ID, 광고 기간 제한 및 리전별 타겟팅을 포함한 수율 최적화 설정 구성"이 나와 있습니다.
6. 활성화 드롭다운을 활성화됨으로 설정합니다.
7. 다음 필드를 입력합니다.
 - APS 게시자 ID: 게시자 계정과 연결된 APS 게시자 ID(예: 8caf35b3-402b-4d75-bab2-f2aef63a66d3). 이를 얻는 방법은 [APS와 통합](#) 단원을 참조하십시오.
 - 최소 미충진 기간: 수율 최적화가 채우려고 시도하는 초 단위의 최소 미충진 기간입니다. 이 값을 15 이상으로 설정합니다. APS 광고 크리에이티브는 일반적으로 15초 이상이므로 15초 미만의 값은 광고를 반환하지 않습니다.

- Amazon Ads 리전: 입찰 요청을 라우팅할 Amazon Ads 리전입니다. 기본 대상에 Asia Pacific 따라 AmericasEurope, 또는 중 하나를 선택합니다. 이는 YieldOptimizationConfiguration API 모델의 Region 필드에 매핑되며 리전 이름은에 설명된 APS 리전에 해당합니다[APS와 통합](#).
- OpenRTB 템플릿 구성: JSON 템플릿. 시작점은 섹션을 참조 [템플릿 예제](#)하세요. 콘솔은 JSON 구문을 검증하고 오류 및 경고를 인라인으로 표시합니다. 기본 템플릿은 콘솔의 스타터 템플릿으로 미리 채워집니다. [입찰 구성 및 템플릿 지정](#)을(를) 참조하세요.

8. 저장을 선택합니다.

2단계: OpenRTB 템플릿 구성

콘솔은 OpenRTB 템플릿에 대한 JSON 편집기를 제공합니다. 템플릿에는 최소한 다음이 포함되어야 합니다.

```
{
  "app": {
    "bundle": "your.app.bundle.id",
    "storeurl": "https://your-app-store-url",
    "content": {}
  },
  "device": {
    "ua": "{{session.user_agent}}",
    "ip": "{{session.client_ip}}"
  },
  "imp": [
    {
      "video": {
        "mimes": ["video/mp4"],
        "protocols": [2, 3, 5, 6],
        "ext": {
          "slotId": "your-aps-slot-id"
        }
      },
      "bidfloor": 1.0
    }
  ]
}
```

필수 필드가 누락된 경우 콘솔에서 경고를 표시합니다.

3단계: 세션 시작 및 확인

1. 플레이어 파라미터(사용자 에이전트, IP 주소 등)가 포함된 재생 세션을 시작합니다.
2. 채워지지 않은 기간의 광고 시간을 기다립니다.
3. `YieldOptimization.BidRequest` 및에 대한 CloudWatch 지표를 확인합니다
`다YieldOptimization.AdsInserted`.

`BidRequest`가를 보내지만 `AdsInserted`가 0인 경우에서 `RAW_BID_REQUEST` 이벤트를 확인하여 템플릿이 유효한 입찰 요청을 생성하고 `ads_interaction_log` 있는지 확인합니다.

2단계(대안): API를 사용하여 구성

`YieldOptimizationConfiguration` 파라미터와 함께 `PutPlaybackConfiguration` API를 사용합니다.

```
aws mediatailor put-playback-configuration \
  --name "MyStreamingService" \
  --ad-decision-server-url "https://ads.example.com/vast" \
  --video-content-source-url "https://origin.example.com/hls/" \
  --yield-optimization-configuration '{
    "MinimumUnfilledDuration": 15,
    "PublisherId": "your-aps-publisher-id",
    "Region": "AMERICAS",
    "OpenRtbTemplate": "{ \"app\": { \"bundle\": \"your.app.bundle\", \"storeurl\": \"https://store-url\", \"content\": {} }, \"device\": { \"ua\": \"{{session.user_agent}}\", \"ip\": \"{{session.client_ip}}\", \"imp\": [ { \"video\": { \"mimes\": [ \"video/mp4\" ], \"protocols\": [ 2, 3, 5, 6 ], \"ext\": { \"slotId\": \"your-slot-id\" }, \"bidfloor\": 1.0 } ] } } }
```

다음 단계

- APS에 등록하고 게시자 ID를 가져오려면 섹션을 참조하세요 [APS와 통합](#).
- MediaTailor가 입찰 요청 시기를 결정하는 방법을 알아보려면 섹션을 참조하세요 [수율 최적화 작동 방식](#).
- 전체 템플릿 예제를 보려면 섹션을 참조하세요 [템플릿 예제](#).

APS와 통합

출시 시 Yield Optimization은 Amazon Publisher Services(APS)를 프로그래밍 방식의 광고 공급자로 사용하며 시간이 지남에 따라 추가 수요 소스가 제공됩니다. 수율 최적화를 활성화하려면 활성 APS 스트리밍 TV 게시자 계정의 APS 게시자 ID가 필요합니다.

APS란 무엇입니까?

Amazon Publisher Services는 프로그래밍 방식의 광고를 통해 게시자를 광고자와 연결하는 Amazon의 광고 기술 플랫폼입니다. 게시자는 APS를 통해 실시간 입찰을 통해 Amazon DSP 및 후원 TV 광고주가 인벤토리를 사용할 수 있도록 합니다.

활성화 프로세스

1. 게시자 ID를 가져옵니다. APS 계정 팀에 문의하여 예제와 유사한 범용 고유 식별자(UUID)인 게시자 ID를 가져옵니다. `8caf35b3-402b-4d75-bab2-f2aef63a66d3`.
2. 수율 최적화를 활성화합니다. MediaTailor 콘솔 또는 API를 통해 게시자 ID를 입력하여 기능을 활성화합니다.

APS 리전

APS는 세 리전에서 작동합니다. 기본 대상과 가장 가까운 리전을 선택합니다.

리전	식별자	적용 범위
북남미	AMERICAS	북미, 남아메리카
유럽	EUROPE	유럽, 중동, 아프리카
아시아 태평양	ASIA_PACIFIC	아시아, 호주, 태평양

Important

APS 리전 선택은 재생 구성당입니다. 단일 재생 구성에서 다중 리전 트래픽을 제공하는 경우 APS 리전을 하나 선택해야 합니다. 다중 리전 지원을 위해 리전별로 별도의 재생 구성을 생성합니다.

속도 제한

Amazon Publisher Services는 대규모 게시자를 수용하도록 확장됩니다. 고객 대면 요금 제한은 적용되지 않습니다.

OpenRTB 프로토콜

MediaTailor는 [OpenRTB 2.6 사양을 사용하여](#) APS와 통신합니다. 입찰 요청은 OpenRTB BidRequest 스키마를 준수하는 JSON 본문이 있는 HTTP POST입니다. APS는 전체 OpenRTB 사양에 대해 요청을 검증하고 적절한 오류 코드를 반환합니다.

다음 단계

- MediaTailor가 입찰 요청 시기를 결정하는 방법을 알아보려면 섹션을 참조하세요 [수율 최적화 작동 방식](#).
- 템플릿을 구성하려면 섹션을 참조하세요 [입찰 구성 및 템플릿 지정](#).

수율 최적화 작동 방식

이 페이지에서는 Yield Optimization의 내부 메커니즘, 즉 트리거 시기, 타이밍 작동 방식 및 다른 MediaTailor 기능과 상호 작용하는 방식을 설명합니다.

수율 최적화가 트리거되는 경우

MediaTailor는 APS에 입찰 요청을 하기 전에 세 가지 조건을 평가합니다. 세 가지 모두 true여야 합니다.

1. 수율 최적화가 구성되어 있습니다. 재생 구성에는 유효한 게시자 ID, 리전 및 템플릿이 `YieldOptimizationConfiguration` 있는가 있습니다.
2. 채우지 않은 기간이 충분합니다. 기본 ADS 응답 후 광고 시간에 채워지지 않은 나머지 시간이 구성된 `MinimumUnfilledDuration` 임계값을 초과합니다.
3. 충분한 개인화 시간이 남아 있습니다. APS를 요청하고 응답을 처리하기에 충분한 시간이 개인화 예산에 남아 있습니다.

조건이 충족되지 않으면 수율 최적화를 건너뛰고 MediaTailor는 기본 ADS 광고만 진행합니다.

채우기 부족 계산 방법

1. MediaTailor는 SCTE 신호에서 총 광고 중단 기간을 읽습니다.

2. MediaTailor는 기본 ADS에서 반환한 모든 광고의 기간을 합산합니다(VAST 기간부터).
3. 채워지지 않은 기간은 총 휴식 기간에서 기본 광고 기간의 합계를 뺀 값과 같습니다.
4. 채워지지 않은 기간이 보다 크거나 같으면 MinimumUnfilledDuration 수율 최적화가 진행됩니다.

예: 기본 ADS가 30초의 광고를 반환하고 30초가 채워지지 않은 상태로 유지되는 60초 광고 시간입니다. MinimumUnfilledDuration이 15로 설정된 경우 Yield Optimization은 나머지 30초 동안 광고를 트리거하고 요청합니다.

개인화 시간 관리

MediaTailor는 광고 개인화를 위한 엄격한 타이밍 예산 내에서 운영됩니다. 수율 최적화는 기본 ADS 요청이 완료된 후 남은 개인화 시간을 사용합니다.

타이밍에 영향을 미치는 주요 SDC(안전 동적 구성) 플래그:

플래그	설명
maximumLivePersonalizationLatencyMs	라이브 콘텐츠에 대한 모든 광고 개인화(기본 ADS + YO)의 최대 총 시간입니다.
adsRequestTimeoutMilliseconds	기본 ADS 요청에 대한 제한 시간입니다.

타이밍 동작:

- APS 요청을 시작하는 데 시간이 충분하지 않으면 YO를 완전히 건너뛰니다 (SkipPersonalizationTimeoutExceeded 지표 생략).
- APS 요청이 시작되지만 제시간에 완료되지 않으면 시간 초과(Timeouts 지표 생략)됩니다.
- MediaTailor는 APS HTTP 요청 및 VAST 처리 모두에 엄격한 제한 시간을 적용하여 제한 내에서 개인화가 완료되도록 합니다.

Fail-open 동작

수율 최적화는 항상 열리지 않습니다. 다음 중 하나가 발생하면 MediaTailor는 기본 ADS 광고만 진행합니다.

- APS는 오류(4xx, 5xx)를 반환합니다.

- 요청 시간이 초과되었습니다.
- 입찰 응답에는 유효한 광고가 포함되어 있지 않습니다.
- 반환된 모든 입찰에 대해 VAST 구문 분석이 실패합니다.

수율 최적화 실패로 인해 최종 사용자의 재생 환경이 중단되지 않습니다.

APS 응답 코드

코드	의미
200	광고를 통한 성공적인 입찰 응답.
204	입찰 없음. 요청이 유효하지만 사용 가능한 입찰이 없습니다.
400	잘못된 요청(잘못된 JSON 또는 필수 필드 누락).
500	APS 서버 오류입니다.
503	APS 오버로드(역압 신호).
429	APS 제한. APS가 요청을 수동으로 제한하기 시작했습니다.

다른 MediaTailor 기능과의 상호 작용

기능	상호 작용
광고 결정 서버 구성(ADSC)	상호 작용이 없습니다. ADSC는 기본 ADS 요청을 제어합니다. YO는 독립적입니다.
미리 가져오기	현재 릴리스에서는 YO에서 지원되지 않습니다.
오버레이 광고	현재 릴리스에서는 YO에서 지원되지 않습니다.

다음 단계

- 템플릿에서 입찰 요청을 작성하는 방법을 알아보려면 섹션을 참조하세요 [입찰 구성 및 템플릿 지정](#).
- 전체 필드 참조는 섹션을 참조하세요 [ORTB 필드 참조](#).

입찰 구성 및 템플릿 지정

이 페이지에서는 MediaTailor가 템플릿에서 OpenRTB 입찰 요청을 작성하는 방법, 변수 보간 작동 방식, MediaTailor가 자동으로 설정하는 값에 대해 설명합니다.

구성 필드

재생 구성에서 수율 최적화를 활성화하면 네 개의 필드를 제공합니다.

필드	설명	필수
PublisherId	APS 게시자 ID(APS 등록에서 획득). 모든 입찰 요청에 대해 <code>app.publisher.id</code> 에 주입됩니다.	예
Region	APS 리전: AMERICAS, EUROPE 또는 ASIA_PACIFIC . APS 엔드포인트를 결정합니다.	예
MinimumUnfilledDuration	MediaTailor가 입찰 요청을 트리거하기 전에 필요한 최소 채워지지 않은 초입니다.로도 사용됩니다 <code>imp.video.minduration</code> .	예
OpenRtbTemplate	ORTB JSON 템플릿(최대 100KB). 입찰 요청 본문을 정의합니다.	예

입찰 요청 구성 방법

광고 시간이 채워지지 않은 기간이 남아 있는 경우 MediaTailor는 다음 단계에서 입찰 요청을 구성합니다.

1. 템플릿 보간. MediaTailor는 표준 [Mustache 템플릿 지정 구문을 사용하여 템플릿의](#) , , `{{session.*}}`, `{{player_params.*}}`, `{{avail.*}}`, `{{request.*}}`, `{{asset.*}}`, 및 `{{scte.*}}` 자리 표시자를 확인합니다. Mustache는 ADS URL 변수 대체에 [] 사용되는 대괄호와 `{{ }}` 다른 이중 중괄호를 사용합니다.
2. JSON 구문 분석. 보간된 템플릿은 JSON으로 구문 분석됩니다.
3. 필드 추출 및 검증. MediaTailor는 필수 필드를 추출하고 누락되거나 비어 있는 `BidRequestConfigurationError` 경우를 발생시킵니다.
4. 하드코딩된 필드 주입. MediaTailor는 콘솔 구성 및 현재 가용 컨텍스트의 값(예: 구성된 게시자 `IDapp.publisher.id`, 계산된 채워지지 않은 기간 `imp.video.maxduration`)을 사용하여 자동으로 관리하는 필드를 재정의하거나 주입합니다.
5. 직렬화 및 제출. 최종 `BidRequest`는 JSON으로 직렬화되고 HTTP POST로 APS 엔드포인트로 전송됩니다.

Important

보간 후 필수 필드가 누락되거나 비어 있으면 입찰 요청이 실패 `BidRequestConfigurationError` 하고 요청이 APS로 전송되지 않습니다. 오류가 로깅되지만 재생은 정상적으로 계속됩니다(실패-열림).

콘솔 기본 템플릿

MediaTailor 콘솔에서 수율 최적화를 활성화하면 다음과 같은 사전 채워진 템플릿이 시작점으로 제공됩니다. 애플리케이션의 값을 사용자 지정해야 합니다.

```
{
  "imp": [
    {
      "bidfloor": 5
    }
  ],
  "app": {
    "id": "{{player_params.app_id}}",
```

```

    "name": "{{player_params.app_name}}",
    "bundle": "{{player_params.bundle}}",
    "storeurl": "{{player_params.storeurl}}",
    "domain": "{{player_params.domain}}",
    "content": {
      "genre": "{{asset.genre}}",
      "contentrating": "{{asset.content_rating}}"
    }
  },
  "device": {
    "dnt": "{{player_params.dnt}}",
    "ua": "{{session.user_agent}}",
    "ip": "{{session.client_ip}}",
    "ifa": "{{player_params.device_ifa}}",
    "w": "{{player_params.device_width}}",
    "h": "{{player_params.device_height}}",
    "language": "{{player_params.language}}",
    "model": "{{player_params.model}}",
    "os": "{{player_params.os}}",
    "osv": "{{player_params.osv}}",
    "devicetype": "{{player_params.devicetype}}",
    "make": "{{player_params.make}}"
  },
  "user": {
    "consent": "{{player_params.consent}}"
  },
  "regs": {
    "gdpr": "{{player_params.gdpr}}",
    "us_privacy": "{{player_params.us_privacy}}",
    "gpp": "{{player_params.gpp_consent}}",
    "gpp_sid": "{{player_params.gpp_sid}}"
  }
}

```

Field-by-field 설명:

Field	카테고리	수행할 작업
imp[0].bidfloor	선택 사항(기본값: 1.0)	최소 CPM을 설정합니다. 콘솔 기본값은 5 USD입니다. 테스트 중에 더 나은 채우기 속도를 위해를 낮춥니다.

Field	카테고리	수행할 작업
app.id	APS 권장	앱 식별자입니다. 플레이어는를 전송해야 합니다playerParams.app_id .
app.name	APS 권장	앱 이름입니다. 플레이어는를 전송해야 합니다playerParams.app_name .
app.bundle	필수	앱 번들 ID입니다. 플레이어는를 전송playerParams.bundle 하거나를 정적 값으로 바꿔야 합니다(권장).
app.storeurl	필수	앱 스토어 URL입니다. 플레이어는를 전송playerParams.storeurl 하거나를 정적 값으로 바꿔야 합니다(권장).
app.domain	APS 권장	앱 도메인. 플레이어는를 전송해야 합니다playerParams.domain .
app.content.genre	APS 권장	콘텐츠 소스 자산 메타데이터에서 채워집니다.
app.content.contentrating	APS 권장	콘텐츠 소스 자산 메타데이터에서 채워집니다.
device.dnt	APS 권장	추적 안 함 플래그. 플레이어는를 전송해야 합니다playerParams.dnt .

Field	카테고리	수행할 작업
device.ua	필수	를 통해 최종 사용자의 사용자-에이전트 헤더에서 자동으로 채워집니다{{session.user_agent}} . 플레이어 파라미터가 필요하지 않습니다.
device.ip	필수	를 통해 최종 사용자의 IP에서 자동으로 채워집니다{{session.client_ip}} . 플레이어 파라미터가 필요하지 않습니다.
device.ifa	APS 권장	광고 ID. 플레이어는 전송해야 합니다playerParams.device_ifa .
device.w / device.h	APS 권장	화면 차원입니다. 플레이어는 playerParams.device_width /를 전송해야 합니다device_height .
device.language	APS 권장	디바이스 언어입니다. 플레이어는 전송해야 합니다playerParams.language .
device.model	APS 권장	디바이스 모델. 플레이어는 전송해야 합니다playerParams.model .
device.os / device.osv	APS 권장	OS 및 버전. 플레이어는 playerParams.os /를 전송해야 합니다osv.

Field	카테고리	수행할 작업
<code>device.devicetype</code>	재정의 가능(기본값: 3)	IAB 디바이스 유형. 플레이어는 전송해야 합니다 <code>playerParams.devicetype</code> . 비어 있는 경우 기본값은 3(CTV)입니다.
<code>device.make</code>	APS 권장	디바이스 제조업체. 플레이어는 전송해야 합니다 <code>playerParams.make</code> .
<code>user.consent</code>	APS 권장(EU)	GDPR 동의 문자열(TCF). 플레이어는 전송해야 합니다 <code>playerParams.consent</code> .
<code>regs.gdpr</code>	APS 권장(EU)	GDPR 플래그(0 또는 1). 플레이어는 전송해야 합니다 <code>playerParams.gdpr</code> .
<code>regs.us_privacy</code>	APS 권장(미국)	CCPA 문자열(예: 1YNY). 플레이어는 전송해야 합니다 <code>playerParams.us_privacy</code> .
<code>regs.gpp</code>	APS 권장	글로벌 프라이버시 플랫폼 동의 문자열입니다. 플레이어는 전송해야 합니다 <code>playerParams.gpp_consent</code> . 더 나은 채우기 속도를 <code>us_privacy</code> 위해 대신 선호됩니다.
<code>regs.gpp_sid</code>	APS 권장	GPP 섹션 IDs(예: 미국 국적[7]의 경우). 플레이어는 전송해야 합니다 <code>playerParams.gpp_sid</code> .

⚠ Important

콘솔 기본값은 `app.bundle` 및와 같은 필수 필드에 `{{player_params.*}}`를 사용합니다. `app.storeurl`. 플레이어가 이러한 파라미터를 보내지 않으면 입찰 요청이와 함께 실패합니다. `BidRequestConfigurationError`. 모든 세션에 대해 동일한 필드(일반적으로 `app.bundle`, `app.storeurl`, `app.name`, 및 기타 앱 수준 식별자)의 경우 자리 표시자를 정적 값으로 바꾸는 것이 좋습니다(예: `"bundle": "com.yourcompany.app"`). , , `device.ua`, `device.ip`, `device.ifa`, GDPR 및 GPP 동의 문자열 또는 모든 `reg` 또는 사용자 객체 필드와 같이 최종 사용자별로 다른 필드를 하드코딩하지 마십시오. 또는 `{{session.*}}`에서 가져와야 합니다 `{{player_params.*}}`.

ℹ Note

콘솔 기본값은 템플릿 `video.protocols`에 `video.mimes` 또는를 포함하지 않습니다. MediaTailor는 하드코딩된 폴백 값(`["video/mp4"]` `mimes`의 경우 , 프로토콜 `[1,2,3,4,5,6,7,8]`의 경우)이 제공되지 않은 경우 이를 사용합니다. 지원되는 형식을 제한하려는 경우 템플릿에 추가할 수 있습니다.

템플릿 보간(Mustache 템플릿 지정)

ORTB 템플릿은 동적 값 대체에 [Mustache 템플릿 구문을](#) 사용합니다. 자리 표시자는 이중 중괄호를 사용합니다 `{{ }}`.

사용 가능한 템플릿 변수

ORTB 템플릿은 ADS URL 템플릿에서 사용할 수 있는 것과 동일한 동적 변수를 사용할 수 있습니다. 여기에는 세션 변수(예: , `{{session.user_agent}}` `{{session.client_ip}}`), 플레이어 파라미터(`{{player_params.*}}`), 자산 메타데이터(`{{asset.*}}`), 가용 컨텍스트(`{{avail.*}}`) 및 SCTE 신호 데이터(`{{scte.*}}`)가 포함됩니다.

사용 가능한 변수의 전체 목록과 설명은 섹션을 참조하세요 [ADS 요청에 대한 MediaTailor 동적 광고 변수](#). MediaTailor URL이 플레이어 파라미터 값을 템플릿으로 대체하기 전에 이를 디코딩하는 방법은 이 페이지의 [인코딩 및 디코딩 동작](#) 섹션을 참조하세요.

ORTB 템플릿에서 가장 일반적으로 사용되는 변수는 다음과 같습니다.

- `{{session.user_agent}}`: 최종 사용자의 사용자-에이전트 헤더(에 사용 `device.ua`).

- `{{session.client_ip}}`: 최종 사용자의 IP 주소(에 사용 `device.ip`).
- `{{player_params.*}}`: 세션 초기화 중에 플레이어가 전달하는 사용자 지정 값입니다.
- `{{asset.*}}`: 오리진의 콘텐츠 메타데이터(예: , `asset.genreasset.content_rating`).

Warning

dnt 또는와 같은 숫자 필드의 경우에도 항상 자리 표시자를 따옴표로 묶습니다. `device.type`. 자리 표시자가 빈 문자열로 확인되고 따옴표로 묶이지 않은 경우 (예: `"dnt": {{player_params.dnt}}`) 결과는 유효하지 않은 JSON이며 전체 입찰 요청이와 함께 실패합니다 `BidRequestConfigurationError`. 대신 `"dnt": "{{player_params.dnt}}"`을 사용하세요. APS는 문자열로 전달된 숫자 필드를 허용합니다. 이 동작은 자동 인용 빈 문자열 대체가 보간기의 다른 곳에서 사용되는 원시 JSON 모드와 충돌하기 때문에 설계에 따른 것입니다. 제약 조건은 적용되지 않고 문서화됩니다. MediaTailor [인코딩 및 디코딩 동작](#)가 보간 전에 플레이어 파라미터 값을 처리하는 방법은 단원을 참조하십시오.

플레이어 파라미터를 전달하는 방법

세션 초기화 중에 플레이어 파라미터를 전달하는 방법에는 두 가지가 있습니다.

방법 1: JSON 본문을 사용한 POST 요청(권장)

POST `https://<mediatailor-endpoint>/v1/session/<config-hash>/<origin-id>/master.m3u8`
Content-Type: `application/json`

```
{
  "playerParams": {
    "app_id": "558775_example",
    "slot_id": "608109df-2378-4bc5-8da5-24bc355f01a4",
    "os": "Tizen",
    "make": "Samsung",
    "model": "Tizen TV",
    "language": "en",
    "us_privacy": "1YNY"
  }
}
```

이 메서드는 사용자 에이전트 문자열과 같은 복잡한 값의 URL 인코딩 문제를 방지하기 때문에 선호됩니다.

방법 2: URL 쿼리 파라미터

```
GET https://<mediatailor-endpoint>/v1/session/<config-hash>/<origin-id>/master.m3u8
?playerParams.app_id=558775_example
```

이 방법을 사용하면 값이 URL로 인코딩되어야 합니다. MediaTailor URL은 보간 전에 이를 디코딩합니다.

인코딩 및 디코딩 동작

MediaTailor는 플레이어 파라미터 값을 템플릿으로 보간하기 전에 플레이어 파라미터 값에 대해 한 수준의 URL 디코딩을 수행합니다. 이는 파라미터 전달 방법(POST 본문 또는 URL 쿼리 파라미터)에 관계없이 적용됩니다.

디코딩된 값의 예:

전송된 값	디코딩 후 값(템플릿에 사용됨)
Test%200s	Test 0s
https%3A%2F%2Flocalhost	https://localhost
foo%20bar	foo bar
Mozilla%2F5.0%20(SMART-TV)	Mozilla/5.0 (SMART-TV)

요점:

- MediaTailor는 한 레벨로만 디코딩합니다. 이중 인코딩된 값(예: %2520)은 공백이 %20아닌 로 디코딩됩니다.
- POST 본문 값도 디코딩됩니다. POST 본문에 URL로 인코딩된 값이 포함된 경우 보간 전에 디코딩됩니다.
- 일반 텍스트 값(인코딩 없음)은 변경되지 않습니다.
- 디코딩 및 보간 후 전체 템플릿은 JSON으로 구문 분석됩니다. 보간된 값의 특수 문자(예: 이스케이프 처리되지 않은 따옴표 또는 백슬래시)는 JSON 구문 분석이 중단될 수 있습니다.

i Tip

플레이어가 이미 일반 텍스트(URL 인코딩되지 않음)인 값을 보내면 그대로 작동합니다. 플레이어가 값을 보내기 전에 URL 인코딩하는 경우에만 디코딩에 유의해야 합니다.

변수가 누락되면 어떻게 되나요?

플레이어가 템플릿에 참조된 파라미터(예: 플레이어 파라미터 `app_name`에는 `{{player_params.app_name}}` 없음)를 보내지 않으면 자리 표시자는 빈 문자열로 확인됩니다"".

영향: 빈 문자열이 필수 필드(예: `app.bundle`)에 있는 경우 입찰 요청에서 실패합니다 `BidRequestConfigurationError`. 선택적 필드에 있는 경우 APS는 입찰 없음(HTTP 204) 응답을 일으킬 수 있는 빈 값을 수신합니다.

MediaTailor가 자동으로 설정하는 항목

다음 필드는 항상 콘솔 구성 및 현재 가용 컨텍스트의 값을 사용하여 MediaTailor에서 설정합니다. 템플릿에 이를 포함하면 값이 재정의됩니다.

Field	소스
<code>id</code>	자동 생성: <code>{sessionId}_{availId}</code> 형식 (예: <code>abc123-def456_78901</code>). 입찰 요청을 세션 로그와 상호 연관시키는 데 유용합니다.
<code>app.publisher.id</code>	콘솔 구성 <code>PublisherId</code> 의.
<code>imp[0].video.minduration</code>	콘솔 구성 <code>MinimumUnfilledDuration</code> 의.
<code>imp[0].video.maxduration</code>	계산됨:이 가용 구간의 실제 채워지지 않은 초입니다.
<code>imp[0].video.maxseq</code>	계산된 값: <code>floor(unfilled_duration / 6)</code> .
<code>imp[0].video.poddur</code>	계산됨:이 가용 구간의 실제 채워지지 않은 초입니다.
<code>ext.integrationType</code>	항상 <code>EMT</code> 로 설정합니다.

다음 단계

- field-by-field 참조는 섹션을 참조하세요 [ORTB 필드 참조](#).
- copy-paste 템플릿은 섹션을 참조하세요 [템플릿 예제](#).

ORTB 필드 참조

이 페이지는 OpenRTB 입찰 요청의 모든 필드에 대한 전체 참조입니다. 필드는 MediaTailor가 필드를 처리하는 방법과 APS가 최적의 광고 채우기를 위해 필드를 필요로 하는지 또는 추천하는지에 따라 분류됩니다.

전체 OpenRTB 2.6 사양은 [IAB OpenRTB 2.6 사양](#)을 참조하세요.

필드 범주

카테고리	의미	작업
필수	템플릿에 있어야 합니다. MediaTailor가 없으면 오류가 발생합니다.	이를 제공해야 합니다.
EMT에서 하드코딩	MediaTailor는 항상 콘솔 구성 또는 현재 가용 컨텍스트의 값을 사용하여 이러한 값을 설정합니다. 템플릿 값이 재정의됩니다.	템플릿에서 이를 설정하지 마십시오.
APS 권장	MediaTailor에서 반드시 필요한 것은 아니지만 APS는 더 나은 광고 채우기 및 타겟팅을 위해 이를 권장합니다. 권장 필드가 누락되면 채우기 속도가 낮거나 입찰 없는 응답이 발생할 수 있습니다.	강력히 권장됩니다.
선택 사항	입찰 요청을 제공하는 경우에만 입찰 요청에 포함됩니다. 기본값이 없습니다.	사용 사례와 관련된 경우를 제공합니다.

app 객체

Field	카테고리	참고
app	필수	app 객체는 템플릿에 있어야 합니다.
app.bundle	필수	앱의 번들 ID(예: G15147002586). 비어 있으면 안 됩니다.
app.storeurl	필수	앱 스토어 URL(예: https://apps.apple.com/us/app/yourapp/id123).
app.publisher.id	EMT에서 하드코딩	콘솔 구성PublisherId 에서 항상를 로 설정합니다.
app.content	필수(객체)	이를 생략하면 MediaTailor가 빈 Content 객체를 전송합니다. 더 나은 타겟팅을 위해 콘텐츠 메타데이터를 제공합니다.
app.name	APS 권장	식별을 위한 앱 이름입니다.
app.id	APS 권장	APS의 앱 식별자입니다.
app.domain	APS 권장	앱 도메인.
app.content.genre	APS 권장	콘텐츠 장르(예: , "Sport""News").
app.content.contentrating	APS 권장	콘텐츠 등급(예: , "TV-G""TV-14").
app.content.language	APS 권장	콘텐츠 언어(ISO-639-1).
app.content.id	APS 권장	콘텐츠 식별자입니다.
app.content.title	APS 권장	콘텐츠 제목.

Field	카테고리	참고
app.content.episode	APS 권장	에피소드 번호입니다.
app.content.len	APS 권장	초 단위의 콘텐츠 길이입니다.
app.cat	선택 사항	콘텐츠 범주(IAB 분류).
app.sectioncat	선택 사항	섹션 범주.
app.pagecat	선택 사항	페이지 범주.
app.ver	선택 사항	앱 버전.
app.privacypolicy	선택 사항	1 = 개인 정보 보호 정책이 있습니다.
app.paid	선택 사항	1 = 유료 앱.
app.keywords	선택 사항	쉼표로 구분된 키워드입니다.
app.kwarray	선택 사항	키워드를 배열로 사용합니다.
app.inventorypartnerdomain	선택 사항	인벤토리 파트너 도메인.
app.ext	선택 사항	확장 객체(패스스루).

device 객체

Field	카테고리	참고
device	필수	device 객체는 템플릿에 있어야 합니다.
device.ua	필수	사용자 에이전트 문자열입니다. 비어 있으면 안 됩니다. {{session.user_agent}} 를 사용합니다.

Field	카테고리	참고
device.ip	필수	IPv4 주소입니다. 비어 있으면 안 됩니다. {{session.client_ip}} 를 사용합니다.
device.lmt	선택 사항(기본값: 0)	광고 추적 제한 플래그. 0 = 추적 허용됨, 1 = 추적 제한. 템플릿에 제공되지 않은 경우 기본값은 0입니다.
device.devicetype	EMT로 하드코딩(기본값: 3)	제공되지 않은 경우 MediaTailor 세트3(연결된 TV)입니다. 값이 있는 경우 사용됩니다.
device.make	APS 권장	디바이스 제조업체(예: , "Samsung" "Apple").
device.model	APS 권장	디바이스 모델(예: , "Tizen TV""iPhone").
device.os	APS 권장	운영 체제(예: , "Tizen""iOS").
device.osv	APS 권장	OS 버전.
device.h	APS 권장	픽셀 단위의 화면 높이입니다.
device.w	APS 권장	픽셀 단위의 화면 너비입니다.
device.language	APS 권장	디바이스 언어(ISO-639-1).
device.ifa	APS 권장	디바이스 광고 ID입니다. 모든 디바이스에 하나(예: 웹 브라우저)가 있는 것은 아닙니다.
device.dnt	APS 권장	추적 안 함 플래그.

Field	카테고리	참고
device.geo	선택 사항	country, lat,가 있는 Geo 객체입니다lon.
device.sua	선택 사항	구조화된 사용자 에이전트(UA-CH).
device.ipv6	선택 사항	IPv6 주소입니다.
device.hwv	선택 사항	하드웨어 버전.
device.connectiontype	선택 사항	연결 유형(WiFi, 셀룰러 등).
device.carrier	선택 사항	통신 사업자 이름입니다.
device.mccmnc	선택 사항	모바일 국가/네트워크 코드.
device.ext	선택 사항	확장 객체(패스스루).

imp (억제) 객체

템플릿은 배열에 정확히 하나의 imp 객체를 포함해야 합니다.

Field	카테고리	참고
imp[0].video	필수	video 객체가 있어야 합니다.
imp[0].video.mimes	필수	지원되는 MIME 유형(예: ["video/mp4"]).
imp[0].video.protocols	선택 사항	지원되는 VAST 프로토콜. 생략하면 MediaTailor는 하드 코딩된 폴백을 사용합니다[1, 2, 3, 4, 5, 6, 7, 8]. 아래 프로토콜 열거형 참조를 참조하세요.

Field	카테고리	참고
<code>imp[0].video.minduration</code>	EMT에서 하드코딩	콘솔 <code>MinimumUnfilledDuration</code> 에서를 로 설정합니다.
<code>imp[0].video.maxduration</code>	EMT에서 하드코딩	를 실제 채워지지 않은 기간(초)으로 설정합니다.
<code>imp[0].video.maxseq</code>	EMT에서 하드코딩	계산된 값: <code>floor(unfilled_duration / 6)</code> . APS는 <code>maxseq</code> 또는를 사용하여 여러 광고를 반환 <code>poddur</code> 할 것을 권장합니다.
<code>imp[0].video.poddur</code>	EMT에서 하드코딩	채우지 않은 기간(초)으로 설정합니다.
<code>imp[0].video.startdelay</code>	선택 사항	시작 지연. 0 = 프리롤, -1 = 일반 미드롤, -2 = 포스트롤, > 0 = 초 단위 미드롤 오프셋. 라이브 미드롤 브레이크-1에 사용합니다. 하드 코딩된 대체: 6.
<code>imp[0].video.w</code>	APS 권장	픽셀 단위의 비디오 너비입니다.
<code>imp[0].video.h</code>	APS 권장	픽셀 단위의 비디오 높이입니다.
<code>imp[0].video.plcmt</code>	APS 권장	배치 유형(1 = 인스트림).
<code>imp[0].video.podid</code>	APS 권장	포드 식별자입니다.
<code>imp[0].video.slotindex</code>	APS 권장	포드의 슬롯 위치입니다.

Field	카테고리	참고
<code>imp[0].video.ext.slotId</code>	선택 사항(기본값: "default_slot_id")	APS 콘솔의 APS 슬롯 ID입니다. 기본값은 작동하지만 실제 슬롯 ID를 제공하면 APS가 슬롯별 대상 지정 및 최저 요금을 적용할 수 있습니다.
<code>imp[0].bidfloor</code>	선택 사항(기본값: 1.0)	CPM의 최소 입찰 가격입니다.
<code>imp[0].secure</code>	선택 사항(기본값: 1)	1 = HTTPS 크리에이티브 URLs 필요합니다.
<code>imp[0].id</code>	선택 사항	노출 ID. 제공되지 않은 경우 자동 생성됩니다.
<code>imp[0].tagid</code>	선택 사항	태그/배치 ID.
<code>imp[0].bidfloorcur</code>	선택 사항	입찰 기준 통화(기본 USD).
<code>imp[0].ssai</code>	선택 사항	SSAI 표시기입니다.
<code>imp[0].exp</code>	선택 사항	노출의 만료 시간입니다.
<code>imp[0].rqddurs</code>	선택 사항	요청된 기간 배열입니다.
<code>imp[0].mincpmpers</code>	선택 사항	초당 최소 CPM입니다.
<code>imp[0].pmp</code>	선택 사항	프라이빗 마켓플레이스 객체.
<code>imp[0].ext</code>	선택 사항	확장 객체(패스스루).

프로토콜 열거형 참조

값	프로토콜
1	VAST 1.0
2	VAST 2.0

값	프로토콜
3	VAST 3.0
4	VAST 1.0 래퍼
5	VAST 2.0 래퍼
6	VAST 3.0 래퍼
7	VAST 4.0
8	VAST 4.0 래퍼
9+	대스트

최상위 필드

Field	카테고리	참고
id	EMT에서 하드코딩	자동 생성: {sessionId}_{availId} 형식(예: abc123-def456_78901). 입찰 요청을 세션 로그와 상호 연관시키는 데 유용합니다.
cur	선택 사항(기본값: ["USD"])	허용된 통화입니다.
ext.integrationType	EMT에서 하드코딩	항상를 로 설정합니다"EMT".
bcat	선택 사항	차단된 광고 범주(IAB 분류). 템플릿 값은 그대로 APS로 전송됩니다.
badv	선택 사항	차단된 광고주 도메인.
bapp	선택 사항	차단된 앱 번들 IDs.
test	선택 사항	테스트 모드 플래그입니다.

Field	카테고리	참고
at	선택 사항	경매 유형.
tmax	선택 사항	입찰 응답을 기다리는 최대 시간(ms)입니다.
wseat	선택 사항	허용된 구매자 좌석.
bseat	선택 사항	구매자 좌석이 차단되었습니다.
wlang	선택 사항	허용된 크리에이티브 언어입니다.
source	선택 사항	공급망의 소스 객체입니다.
ext	선택 사항	사용자 지정 확장 필드는 EMT의와 병합됩니다integrationType .

user 객체

Field	카테고리	참고
user	APS 권장(EU)	GDPR 동의가 있는 EU 트래픽에 필요합니다.
user.consent	APS 권장(EU)	GDPR 동의 문자열(TCF).
user.id	선택 사항	사용자 ID입니다.
user.buyeruid	선택 사항	구매자별 사용자 ID입니다.
user.eids	선택 사항	확장 식별자(LiveRamp, UID2 등).
user.data	선택 사항	사용자 데이터 세그먼트.

Field	카테고리	참고
<code>user.ext</code>	선택 사항	확장 객체.

regs 객체

Field	카테고리	참고
<code>regs</code>	APS 권장	규정 준수를 위해를 포함합니다.
<code>regs.coppa</code>	APS 권장(미국)	COPPA 플래그(1 = 하위 대상).
<code>regs.us_privacy</code>	APS 권장(미국)	미국 개인 정보 보호/CCPA 문자열입니다.
<code>regs.gdpr</code>	APS 권장(EU)	GDPR이 적용됩니다(1 = 예).
<code>regs.gpp</code>	APS 권장	글로벌 프라이버시 플랫폼 문자열입니다. <code>us_privacy</code> 미국 시청자보다 선호되므로 채우기 속도가 향상됩니다. 모범 사례 을(를) 참조하세요.
<code>regs.gpp_sid</code>	APS 권장	GPP 섹션 IDs(예: 미국 국적[7]의 경우). <code>gpp</code> 가 있는 경우 필요합니다.
<code>regs.ext</code>	선택 사항	확장 객체.

다음 단계

- copy-paste 템플릿은 섹션을 참조하세요 [템플릿 예제](#).
- 채우기 속도 최대화에 대한 권장 사항은 섹션을 참조하세요 [모범 사례](#).

템플릿 예제

이 페이지에서는 일반적인 사용 사례에 대한 완전한 작동 템플릿 구성을 제공합니다. 각 예제에는 템플릿 JSON, 필수 플레이어 파라미터, 사용자 지정할 항목에 대한 메모가 포함되어 있습니다.

Note

예제에서는 APS에서 HTTP 204(no-bidyour_aps_slot_id_here)를 반환하는 자리 표시자 값(예: com.example.myapp,)을 사용합니다. 입찰 응답을 받으려면 모든 값을 실제 앱 번들, 스토어 URL, APS 슬롯 ID, 디바이스 사용자 에이전트 및 IP 주소로 바꿉니다.

예제 1: 미니멀 템플릿(CTV 앱)

가장 간단한 유효한 템플릿입니다. 테스트를 위한 좋은 시작점입니다. 실제 뷰어에게 작동하도록 디바이스 식별에 세션 변수를 사용합니다.

```
{
  "app": {
    "bundle": "com.example.myapp",
    "storeurl": "https://play.google.com/store/apps/details?id=com.example.myapp",
    "name": "My Streaming App",
    "content": {
      "language": "en"
    }
  },
  "device": {
    "ua": "{{session.user_agent}}",
    "ip": "{{session.client_ip}}",
    "devicetype": 3
  },
  "imp": [
    {
      "video": {
        "mimes": ["video/mp4"],
        "protocols": [2, 3, 5, 6],
        "w": 1920,
        "h": 1080,
        "ext": {
          "slotId": "your_aps_slot_id_here"
        }
      }
    }
  ]
}
```

```

    },
    "bidfloor": 5.0
  }
]
}

```

기본 바닥은 \$5 CPM입니다. 테스트 중에 입찰을 받지 못하는 경우이 값(예: 1.0 또는 0.01)을 낮추어 적격 광고주의 풀을 늘리십시오.

플레이어 파라미터 필요: 없음. device.ua 및 device.ip는 최종 사용자의 세션에서 자동으로 채워집니다.

사용 시기: 게시자 ID 및 APS 통합이 작동하는지 초기 테스트 및 검증합니다. 이후 프로덕션에 적합 device.ua하며 세션 변수를 통해 실제 뷰어를 자동으로 device.ip 반영합니다.

예제 2: 플레이어 파라미터가 있는 동적 템플릿

앱 및 디바이스 식별에 플레이어 파라미터를 사용합니다. 프로덕션 다중 디바이스 배포에 적합합니다.

```

{
  "app": {
    "id": "{{player_params.app_id}}",
    "bundle": "com.example.myapp",
    "storeurl": "https://play.google.com/store/apps/details?id=com.example.myapp",
    "name": "{{player_params.app_name}}",
    "domain": "{{player_params.domain}}",
    "content": {
      "genre": "{{asset.genre}}",
      "contentrating": "{{asset.content_rating}}",
      "language": "{{player_params.language}}"
    }
  },
  "device": {
    "ua": "{{session.user_agent}}",
    "ip": "{{session.client_ip}}",
    "devicetype": 3,
    "ifa": "{{player_params.device_ifa}}",
    "os": "{{player_params.os}}",
    "osv": "{{player_params.osv}}",
    "model": "{{player_params.model}}",
    "make": "{{player_params.make}}",
    "language": "{{player_params.language}}",
    "w": 1920,

```



```

    "h": 1080
  },
  "imp": [
    {
      "video": {
        "mimes": ["video/mp4", "video/webm"],
        "protocols": [2, 3, 5, 6],
        "w": 1920,
        "h": 1080,
        "startdelay": -1,
        "plcmt": 1,
        "ext": {
          "slotId": "{{player_params.slot_id}}"
        }
      },
      "bidfloor": 2.0,
      "secure": 1
    }
  ],
  "user": {
    "consent": "{{player_params.consent}}"
  },
  "regs": {
    "coppa": 0,
    "gdpr": "{{player_params.gdpr}}",
    "us_privacy": "{{player_params.us_privacy}}",
    "gpp": "{{player_params.gpp_consent}}",
    "gpp_sid": "{{player_params.gpp_sid}}"
  }
}

```

플레이어 파라미터 권장:

- `playerParams.app_id`, `playerParams.app_name`, `playerParams.domain`
- `playerParams.device_ifa`, `playerParams.os`, `playerParams.osv`,
`playerParams.model`, `playerParams.make`
- `playerParams.language`, `playerParams.slot_id`
- `playerParams.consent`, `playerParams.us_privacy`, `playerParams.gdpr`,
`playerParams.gpp_consent`, `playerParams.gpp_sid`

세션 초기화 예제(POST 본문):

```
POST https://<mediatailor-endpoint>/v1/session/<config-hash>/<origin-id>/master.m3u8

{
  "playerParams": {
    "app_id": "558775_example",
    "app_name": "My Streaming App",
    "domain": "myapp.com",
    "slot_id": "608109df-2378-4bc5-8da5-24bc355f01a4",
    "os": "Tizen",
    "osv": "5.5",
    "make": "Samsung",
    "model": "Tizen TV",
    "device_ifa": "000000000-0000-0000-0000-000000000000",
    "language": "en",
    "consent": "1",
    "us_privacy": "1YNY",
    "gdpr": "0",
    "gpp_consent": "DBABLA~BVQqAAAAAgA",
    "gpp_sid": "[7]"
  }
}
```

예제 3: 모든 기능을 갖춘 템플릿(프로덕션 지원)

모든 APS 권장 필드, 규정 준수, 콘텐츠 대상 지정 및 확장 식별자가 포함된 포괄적인 템플릿입니다.

```
{
  "app": {
    "id": "{{player_params.app_id}}",
    "bundle": "com.example.streamingapp",
    "storeurl": "https://www.amazon.com/dp/B0EXAMPLE",
    "name": "Example Streaming",
    "domain": "streaming.example.com",
    "cat": ["IAB1-1"],
    "content": {
      "genre": "{{asset.genre}}",
      "contentrating": "{{asset.content_rating}}",
      "language": "en",
      "len": 3600
    }
  },
  "device": {
    "ua": "{{session.user_agent}}",
```

```

    "ip": "{{session.client_ip}}",
    "devicetype": 3,
    "dnt": "{{player_params.dnt}}",
    "ifa": "{{player_params.ifa}}",
    "os": "{{player_params.os}}",
    "osv": "{{player_params.osv}}",
    "model": "{{player_params.model}}",
    "make": "{{player_params.make}}",
    "language": "en",
    "w": 1920,
    "h": 1080,
    "connectiontype": 2
  },
  "imp": [
    {
      "video": {
        "mimes": ["video/mp4", "video/webm"],
        "protocols": [1, 2, 3, 4, 5, 6, 7, 8],
        "w": 1920,
        "h": 1080,
        "startdelay": -1,
        "podid": "1",
        "plcmt": 1,
        "linearity": 1,
        "playbackmethod": [1],
        "ext": {
          "slotId": "608109df-a1b2-c3d4-e5f6-789012345678"
        }
      },
      "bidfloor": 3.0,
      "bidfloorcur": "USD",
      "secure": 1,
      "ssai": 1
    }
  ],
  "user": {
    "id": "{{player_params.user_id}}",
    "eids": [
      {
        "source": "liveramp.com",
        "uids": [
          {
            "id": "{{player_params.envelope}}",
            "atype": 3
          }
        ]
      }
    ]
  }
}

```

```

    }
  ]
}
],
"consent": "{{player_params.consent}}"
},
"regs": {
  "coppa": 0,
  "gdpr": "{{player_params.gdpr}}",
  "us_privacy": "{{player_params.us_privacy}}",
  "gpp": "{{player_params.gpp_consent}}",
  "gpp_sid": "{{player_params.gpp_sid}}"
},
"bcat": ["IAB25", "IAB26"],
"badv": ["competitor.com"]
}

```

예제 4: APS 직접 테스트(Postman/curl)

이 예제를 사용하여 MediaTailor 외부에서 직접 APS 엔드포인트를 테스트하여 게시자 ID 및 입찰 파라미터가 유효한 응답을 생성하는지 확인합니다. 이를 Postman에 붙여넣거나 curl를 사용하여 MediaTailor 템플릿을 구성하기 전에 APS가 입찰을 반환하는지 확인합니다.

엔드포인트

```

POST https://emt-ortb-iad.stv.bid-na.ads.aps.amazon-adsystem.com/stv/ortb/ads
Content-Type: application/json

```

Note

리전 AMERICAS 엔드포인트(iad = 미국 동부)입니다. 다른 리전의 경우 호스트 이름이 그에 따라 변경됩니다.

입찰 요청 처리

로 표시된 자리 표시자 값을 실제 값으로 바꿉니다. <YOUR_VALUE>입니다. 다른 모든 필드는 유효한 작업 예제입니다.

```

{
  "id": "test-request-001",
  "imp": [

```

```

{
  "id": "test_imp_1",
  "video": {
    "mimes": ["video/mp4", "video/H264"],
    "minduration": 15,
    "maxduration": 30,
    "protocols": [1, 2, 3, 4, 5, 6, 7, 8],
    "w": 1920,
    "h": 1080,
    "placement": 1,
    "linearity": 1,
    "sequence": 1,
    "minbitrate": 240,
    "maxbitrate": 30000,
    "playbackmethod": [1],
    "delivery": [1, 2],
    "maxseq": 5,
    "poddur": 120,
    "podid": "1",
    "podseq": 0,
    "plcmt": 1
  },
  "tagid": "test_imp_1",
  "bidfloor": 1.5,
  "secure": 1
},
],
"app": {
  "id": "<YOUR_APP_ID>",
  "name": "<YOUR_APP_NAME>",
  "bundle": "<YOUR_APP_BUNDLE_ID>",
  "domain": "<YOUR_APP_DOMAIN>",
  "storeurl": "<YOUR_APP_STORE_URL>",
  "pagecat": ["IAB12"],
  "publisher": {
    "id": "<YOUR_APS_PUBLISHER_ID>"
  },
},
"content": {
  "genre": "Sports",
  "cat": ["IAB12"],
  "prodq": 1,
  "context": 1,
  "contentrating": "TV-G",
  "livestream": 1,

```

```

    "language": "en"
  },
  "device": {
    "ua": "Dalvik/5.36.0 (Linux; U; FIRE; en-US; Amazon Model/AFTHA004 OS/9)",
    "geo": {
      "country": "USA",
      "region": "LA",
      "metro": "622",
      "city": "Metairie",
      "zip": "70001"
    },
    "dnt": 0,
    "lmt": 0,
    "ip": "<YOUR_CLIENT_IP>",
    "devicetype": 3,
    "make": "Amazon",
    "model": "AFFT",
    "os": "Android",
    "osv": "5.0",
    "ifa": "<YOUR_CLIENT_IFA>"
  },
  "at": 1,
  "tmax": 1850,
  "source": {
    "tid": "test-transaction-001"
  },
  "regs": {
    "ext": {
      "gdpr": 0,
      "us_privacy": "1YNY"
    }
  }
}

```

필드 주석

아래 표에서는 각 필드가 나타내는 것과 MediaTailor가 프로덕션 환경에서 설정하는 것과 매핑되는 방법을 설명합니다.

Field	설정자	참고
id	MediaTailor(하드코딩됨)	프로덕션에서 로 자동 생성됩니다{sessionId}_{availId} . 테스트에는 고유한 문자열을 사용합니다.
imp[0].video.minduration	MediaTailor(하드코딩됨)	구성에서 MinimumUnfilledDuration 를 설정합니다. 테스트15에 사용합니다.
imp[0].video.maxduration	MediaTailor(하드코딩됨)	채워지지 않은 휴식 시간을 기준으로 계산됩니다. 테스트30에 사용합니다.
imp[0].video.maxseq	MediaTailor(하드코딩됨)	계산된 값: $\text{floor}(\text{unfilled_duration} / 6)$. 테스트5에 사용합니다.
imp[0].video.poddur	MediaTailor(하드코딩됨)	채워지지 않은 휴식 시간입니다. 테스트120에 사용합니다.
imp[0].video.mimes	템플릿	지원되는 비디오 형식입니다. 면 충분["video/mp4"] 합니다.
imp[0].video.protocols	템플릿	VAST 프로토콜 버전. 호환성[1,2,3,4,5,6,7,8] 을 극대화합니다.
imp[0].video.w / h	템플릿	비디오 해상도. 스트림 출력을 일치시킵니다.
imp[0].bidfloor	템플릿	최소 CPM. 낮음 = 더 많은 입찰이 반환됩니다.

Field	설정자	참고
app.publisher.id	MediaTailor(하드코딩됨)	콘솔 구성의 APS 게시자 ID입니다. 를 사용자의 로 바꿉니다.
app.bundle	템플릿(필수)	앱의 번들/ASIN. 를 사용자의 로 바꿉니다.
app.storeurl	템플릿(필수)	앱 스토어 목록 URL입니다. 를 사용자의 로 바꿉니다.
app.content.*	템플릿	대상 지정에 위한 콘텐츠 메타 데이터입니다. 입찰 품질을 개선합니다.
device.ua	템플릿(필수)	프로덕션 환경에서는의 입니다 <code>{{session.user_agent}}</code> . 테스트에는 실제 디바이스 UA를 사용합니다.
device.ip	템플릿(필수)	프로덕션 환경에서는의 입니다 <code>{{session.client_ip}}</code> . 테스트에는 실제 미국 IP를 사용합니다.
device.geo.*	전달	선택적 지리적 위치. APS는 IP에서 파생될 수도 있습니다.
device.ifa	템플릿	디바이스 광고 ID입니다. 대상 지정을 개선합니다.
regs.ext.us_privacy	템플릿	CCPA 문자열입니다. 더 나은 채우기를 위해 프로덕션에서 GPP(regs.gpp)를 사용합니다.
at	전달	경매 유형. 1 = 첫 번째 가격.

Field	설정자	참고
tmax	전달	최대 응답 대기 시간(ms). APS는 일반적으로 <200ms 내에 응답합니다.

예상 응답

성공 응답(HTTP 200)은 하나 이상의 입찰이 포함된 seatbid 배열을 반환합니다. 각 입찰에는 다음이 포함됩니다.

- price: CPM 입찰 가격(예: 5.00).
- adm: 광고 크리에이티브(Wrapper 또는 InLine)가 포함된 VAST XML입니다.
- adomain: Advertiser 도메인(예: ["landofrost.com"]).
- dur: 초 단위의 광고 기간입니다.
- cat: IAB 콘텐츠 범주.
- ext.dsrc: 수요 소스(예: Amazon DSP"ADSP"의 경우).

HTTP 204를 수신하면 APS에 일치하는 입찰이 없습니다. 이는 일반적으로 다음을 의미합니다.

- publisher.id가 유효하지 않거나 활성화되지 않았습니다.
- app.bundle /가 등록된 앱과 일치하지 app.storeurl 않습니다.
- bidfloor가 사용 가능한 수요에 비해 너무 높습니다.
- device.ip는 APS 적용 범위를 벗어납니다.

curl 예제

```
curl -X POST \
  'https://emt-ortb-iad.stv.bid-na.ads.aps.amazon-adsystem.com/stv/ortb/ads' \
  -H 'Content-Type: application/json' \
  -d '{
    "id": "test-request-001",
    "imp": [{"id": "test_imp_1", "video": {"mimes": ["video/mp4"], "minduration": 15,
    "maxduration": 30, "protocols": [1,2,3,4,5,6,7,8], "w": 1920, "h": 1080, "maxseq": 5,
    "poddur": 120, "podid": "1", "plcmt": 1}, "bidfloor": 1.5, "secure": 1}],
```

```
"app": {"bundle": "<YOUR_BUNDLE>", "storeurl": "<YOUR_STORE_URL>", "publisher":
{"id": "<YOUR_PUBLISHER_ID>", "content": {"genre": "Sports", "livestream": 1,
"language": "en"}},
"device": {"ua": "Dalvik/5.36.0 (Linux; U; FIRE; en-US; Amazon Model/AFTHA004 OS/9)",
"ip": "98.172.189.170", "devicetype": 3, "make": "Amazon", "model": "AFFT", "os":
"Android"},
"at": 1, "tmax": 1850,
"regs": {"ext": {"gdpr": 0, "us_privacy": "1YNY"}}
}'
```

다음 단계

- 채우기 속도 최대화에 대한 권장 사항은 섹션을 참조하세요 [모범 사례](#).
- 모니터링 및 디버깅은 섹션을 참조하세요 [문제 해결 및 모니터링](#).

모범 사례

템플릿 구성

최소 작업 템플릿으로 시작

의 예제 1 [템플릿 예제](#)을 사용하여 APS 통합을 검증합니다. CloudWatch Logs에서 RAW_BID_REQUEST 이벤트를 확인하여 필드를 점진적으로 추가하고 각 추가를 테스트합니다.

모든 APS 권장 필드 제공

에서 "APS Recommended"로 표시된 필드는 채우기 속도에 [ORTB 필드 참조](#) 큰 영향을 미칩니다. 최소한 app.bundle, , app.storeurl, app.name, device.ua, device.ip, , device.make, device.model, device.osvideo.mimes, video.protocols, video.w, 및 video.h를 포함합니다 video.plcmtvideo.ext.slotId.

디바이스 식별에 세션 변수 사용

device.ua 및 device.ip 필드는 적절한 광고 타겟팅을 위해 실제 뷰어를 반영해야 합니다. 최종 사용자의 HTTP 연결에서 자동으로 채워진 {{session.client_ip}}지는 {{session.user_agent}} 및를 사용합니다. 이러한 필드에는 플레이어 파라미터 설정이 필요하지 않습니다.

실제 **app.bundle** 및 제공 **app.storeurl**

APS는 이를 앱 식별 및 광고 타겟팅에 사용합니다. 또는 자리 표시자 값이 잘못되면 입찰 없음(204) 응답이 발생합니다.

적절한 설정 **imp[0].bidfloor**

콘솔 기본값은 \$5 CPM입니다. 채우기 속도가 낮으면 바닥을 낮추는 것이 좋습니다. 바닥이 낮을수록 적격 광고주의 풀이 증가하지만 평균 CPM은 감소할 수 있습니다.

실제 **video.ext.slotId**

슬롯 ID는 APS 인벤토리 구성에 매핑됩니다. (APS 콘솔에서) 실제 슬롯 ID를 제공하면 APS가 슬롯별 대상 지정 및 최저 요금을 적용할 수 있습니다. 기본값("default_slot_id")은 작동하지만 광고 선택이 최적화되지 않을 수 있습니다.

광범위한 **video.protocols** 지원 포함

[2, 3, 5, 6] 최소한 (VAST 2.0, 3.0, 2.0 래퍼, 3.0 래퍼)를 사용합니다. 호환성을 극대화[1, 2, 3, 4, 5, 6, 7, 8]하려면 사용하여 모든 VAST 버전(1.0~4.0, 래퍼 포함)을 지원합니다.

EMT가 관리하는 필드를 설정하지 마십시오.

imp.video.maxduration, , imp.video.poddur, imp.video.mindurationapp.publisher.id, 및 imp.video.maxseq와 같은 필드는 ext.integrationType 항상 콘솔 구성 값을 사용하여 MediaTailor에서 설정합니다. 템플릿에 포함 시키는 것은 무해하지만 오해의 소지가 있습니다.

미국 개인 정보 신호 **regs.us_privacy**에 대해 **regs.gpp** 및 **regs.gpp_sid**를 선호합니다.

미국 프라이버시 신호를 APS에 전달할 때는 이전 us_privacy 문자열 대신 글로벌 프라이버시 플랫폼(GPP) 필드를 사용합니다.

```
"regs": {
  "coppa": 0,
  "gpp": "DBABMA~1---",
  "gpp_sid": [7]
}
```

APS는 us_privacy 문자열(CCPA 옵트아웃 신호)이 표시될 때 더 엄격한 데이터 사용 제한을 적용하여 적격 광고자의 풀을 제한하고 채우기 속도를 줄입니다. GPP는 APS에 더 세분화된 관할권 수준 신호를 제공하여 규정 준수 시행을 더 정확하게 수행하고 적격 입찰 풀을 더 넓게 만듭니다. 플레이어가 두 신호를 모두 전송하는 경우 GPP를 선호하고를 생략합니다us_privacy.

미국 개인 정보 보호를 위한 GPP 섹션 IDs:

섹션 ID	Scope
7	미국 국가(다국어 개인 정보 보호)
8	미국 캘리포니아(CPRA)
9	미국 버지니아(VCDPA)
10	미국 콜로라도(CPA)
11	미국 유타(UCPA)
12	미국 코네티컷(CTDPA)

다운스트림 시스템에 `us_privacy`를 포함해야 하는 경우 둘 다 전달하는 것이 안전하지만 APS는 있을 때 GPP 신호를 사용합니다.

운영상의 고려 사항

리전 구성

APS 리전은 재생 구성별로 설정됩니다. 여러 APS 리전에서 최종 사용자에게 서비스를 제공하는 경우 최적의 지연 시간과 광고 인벤토리 가용성을 위해 리전별로 별도의 재생 구성을 생성합니다.

수율 리프트 모니터링

CloudWatch 지표를 사용하여 수율 리프트를 계산합니다.

- 채우기 속도 개선: 총 가용 시간과 `YieldOptimization.AdsInserted` 비교하여 증분 채우기를 측정합니다. `YieldOptimization.FillRate` (기본 가용 시간당 지표)도 일정 기간 동안 평균화할 수 있습니다.
- 수익 기여도: 고객당 총 수익(삽입된 광고 가격의 합계) `YieldOptimization.Revenue`에 사용합니다. 계정 전체의 평균 CPM `YieldOptimization.BidPriceAdsInserted`에 사용합니다. `YieldOptimization.IndicativeBidRevenue`는 사용 중단 기간 Revenue 동안과 함께 이중 방출되며 제거됩니다. 새 대시보드 `Revenue`를 선호합니다.
- 효율성: `YieldOptimization.BidRequest` `YieldOptimization.AdsInserted`와 비교하여 변환율을 측정합니다.

해결되지 않은 자리 표시자 패턴 방지

ADS URL이 [player_params.X] 사용하고 플레이어가 파라미터 X를 전송하지 않는 경우 리터럴 {X} 자리 표시자는 ORTB 템플릿으로 전달됩니다. APS는 리터럴 문자열 "{APP_NAME}"로 보고 no-bid를 반환합니다. 항상 RAW_BID_REQUEST페이로드에 {UPPER_CASE} 패턴이 포함되어 있지 않은지 확인하세요.

다음 단계

- 모니터링 및 디버깅은 섹션을 참조하세요 [문제 해결 및 모니터링](#).

문제 해결 및 모니터링

이 페이지에서는 수율 최적화 성능을 모니터링하고, 일반적인 문제를 진단하고, 구성이 올바르게 작동하는지 확인할 수 있습니다.

Metrics

MediaTailor는 수율 최적화에 대해 다음과 같은 CloudWatch 지표를 게시합니다. 달리 명시되지 않는 한 모든 지표는 configurationName차원과 함께 내보내집니다.

입찰 요청 지표

지표	설명
YieldOptimization.BidRequest	입찰 요청 기회가 식별되면 1로 내보내집니다.
YieldOptimization.BidResponse	성공적인 입찰 응답이 수신되면 1로 내보내집니다.
YieldOptimization.BidDuration	입찰에서 요청된 중단 기간(초)으로, 가용 기간에서 기본 광고 기간을 뺀 값에서 계산됩니다.
YieldOptimization.Latency	APS까지의 왕복 지연 시간을 밀리초 단위로 표시합니다.
YieldOptimization.NoBid	APS가 HTTP 204(일치하는 수요 없음)를 반환할 때 1로 내보내집니다. 요청 자체가 유효했습니다.

지표	설명
YieldOptimization.BidRequestThrottled	APS가 HTTP 429(제한됨)를 반환할 때 1로 내보내집니다. APS가 역압을 적용하고 있습니다. 다시 시도하기 전에 다시 시작합니다.

광고 삽입 지표

지표	설명
YieldOptimization.AdsInserted	가용 영역에 삽입된 YO 광고 수입니다.
YieldOptimization.AdsReturned	BidResponse에서 반환된 광고 수(필터링 전). 0보다 큰 경우에만 내보내집니다.
YieldOptimization.FilledDuration	YO 광고로 채워진 기간(ms)입니다.
YieldOptimization.UnfilledDuration	입찰 요청에서 APS로 전송된 기간(ms)(기회 차이).
YieldOptimization.FillRate	가용 시간당 채우기 속도: 트랜스코딩된 YO 기간을 미리 채워지지 않은 기간으로 나눈 값입니다. 분자는 트랜스코딩되고 분모는 VAST 기반(범퍼를 빼지 않음)이므로 값이 1.0을 초과할 수 있습니다.

수익 및 요금 지표

지표	설명
YieldOptimization.Revenue	oRTB 입찰 응답의 price 속성에서 얻은 고객의 총 수익입니다. 삽입된 광고의 가격 합계입니다.
YieldOptimization.IndicativeBidRevenue	모든 RETURNED 입찰(노출당 입찰 가격, 삽입당 입찰 가격 아님)의 압축당 표시 값 합계입니다. 사용 중단 기간 Revenue 동안과 함께 이중

지표	설명
	방출되며 대시보드가 전환되면 제거됩니다. 새 대시보드Revenue에를 사용합니다.
YieldOptimization.BidPriceAdsInserted	Yield Optimization에서 삽입한 각 광고의 CPM 가격입니다. 입찰 응답cur(통화) 속성으로 측정됩니다. 고객당 방출되지 않으므로 계정 전체에 적용됩니다.
YieldOptimization.BidPriceAdsReturned	BidResponse에 반환된 각 입찰의 CPM 가격(삽입으로 필터링하기 전). 로 측정됩니다cur. 고객당 방출되지 않으므로 계정 전체에 적용됩니다.

오류 및 지표 건너뛰기

지표	설명
YieldOptimization.Errors	연결 오류, 응답 처리 오류 또는 예상치 못한 예외의 경우 1로 내보내집니다. 또한 성공 시 0으로 내보내집니다(SampleCount가 아닌 Sum 사용).
YieldOptimization.Timeouts	APS 요청 시간이 초과되면 1로 내보내집니다. 또한 성공 시 0으로 내보내집니다(SampleCount가 아닌 Sum 사용).
YieldOptimization.BidRequestConfigurationError	템플릿 또는 입찰 구성이 실패하면(필드 누락, 잘못된 JSON) 1로 내보내집니다.
YieldOptimization.SkipMinUnfilledDurationNotExceeded	채워지지 않은 기간이 임계값 미만이어서 YO를 건너뛰면 1로 내보내집니다.
YieldOptimization.SkipPersonalizationTimeoutExceeded	개인화 시간이 충분하지 않아 YO를 건너뛰면 1로 내보내집니다.

⚠ Important

`YieldOptimization.Errors` 및 `YieldOptimization.Timeouts`는 요청이 성공할 때 마다 0으로 내보내집니다. CloudWatch에서 `Sum(SampleCount 아님)`을 사용하여 실제 오류 또는 제한 시간 수를 가져옵니다. `NoBid` 및 `BidRequestThrottled`는에서 계산되지 않습니다. `Errors`. 이는 합법적인 비수요 및 역압 조건에 대한 별도의 신호입니다.

로그 이벤트

수율 최적화는 `MidasMMSService/ads_interaction_log` 로그 그룹에서 이벤트를 생성합니다.

이벤트 유형	설명
RAW_BID_REQUEST	APS로 전송된 전체 OpenRTB 입찰 요청 JSON입니다. 이를 사용하여 템플릿 보간을 확인합니다.
RAW_BID_RESPONSE	입찰 요금 및 VAST XML을 포함하여 APS의 원시 OpenRTB 입찰 응답입니다.
FILLED_AVAIL	모든 광고가 포함된 전체 가용 시간입니다. YO 광고에는 <code>bidId</code> , <code>bidPrice</code> , 및 <code>fillMode: "YIELD_OPTIMIZATION"</code> 포함되어 <code>currency</code> <code>impressionId</code> 있습니다. <code>seatId</code> .

오류 이벤트는 `YieldOptimizationSupport`, `OrtbBidRequestBuilder`, `HttpOrtbRequestSubmitter` 및와 같은 로거와 `MidasMMSService/application.log` 함께에서 발생합니다. `OrtbResponseProcessor`.

CloudWatch Insights 쿼리 예제

YO 오류 확인

```
fields @timestamp, message.sessionId, message.message, log.logger, log.level
| filter message.awsAccountId = 'YOUR_ACCOUNT_ID'
  and (log.level = 'ERROR' or log.level = 'WARN')
```



```

and (log.logger like /YieldOptimization/
    or log.logger like /OrtbResponseProcessor/
    or log.logger like /HttpOrtbRequestSubmitter/
    or log.logger like /OrtbBidRequestBuilder/)
| sort @timestamp desc
| limit 100

```

로그 그룹: MidasMMSService/application.log

입찰 요청 수와 응답 수 비교

```

fields @timestamp, sessionId, eventType
| filter awsAccountId = 'YOUR_ACCOUNT_ID'
    and (eventType = 'RAW_BID_REQUEST' or eventType = 'RAW_BID_RESPONSE')
| stats count(*) by eventType

```

로그 그룹: MidasMMSService/ads_interaction_log

채워진 가능 구간에서 YO 광고 찾기

각 FILLED_AVAIL 이벤트에는 creativeAds 배열이 포함됩니다. 해당 배열 내의 YO 광고에는 fillMode: "YIELD_OPTIMIZATION", bidId, impressionId, 및 추가 입찰 메타데이터bidPriceseatId가 포함됩니다currency. 이를 사용하여 YO로 채워진 가용 시간, 가용 시간당 삽입된 YO 광고 수, 입찰당 수익 어트리뷰션을 확인할 수 있습니다.

```

fields @timestamp, sessionId, avail.availId, avail.numAds, avail.fillRate,
    avail.filledDuration
| filter awsAccountId = 'YOUR_ACCOUNT_ID'
    and eventType = 'FILLED_AVAIL'
    and @message like /YIELD_OPTIMIZATION/
| sort @timestamp desc
| limit 50

```

특정 세션에 대한 개별 YO 광고 항목(입찰 ID 및 가격)을 가져오려면:

```

fields @timestamp, sessionId, avail.availId, avail.creativeAds
| filter awsAccountId = 'YOUR_ACCOUNT_ID'
    and eventType = 'FILLED_AVAIL'
    and sessionId = 'YOUR_SESSION_ID'
| sort @timestamp desc
| limit 20

```

결과 creativeAds 배열 클라이언트 측에서가 있는 항목을 필터링합니다 fillMode == "YIELD_OPTIMIZATION".

로그 그룹: MidasMMSService/ads_interaction_log

해결되지 않은 템플릿 변수 확인

```
fields @timestamp, sessionId, @message
| filter awsAccountId = 'YOUR_ACCOUNT_ID'
  and eventType = 'RAW_BID_REQUEST'
  and @message like /[A-Z_]+\//
| sort @timestamp desc
| limit 20
```

로그 그룹: MidasMMSService/ads_interaction_log

YO 지연 시간 및 성공률 측정

```
fields @timestamp, @message
| filter customerId = 'YOUR_ACCOUNT_ID'
  and (@message like /YieldOptimization/ or @message like /ortb_/)
| limit 10000
```

로그 그룹: MidasMMSService/service_log

Note

service_log는 계정 ID 필드에 customerId (가 아님awsAccountId)를 사용합니다.

수율 리포트 계산

수율 최적화의 수익 영향을 측정하려면 다음을 사용하여 CloudWatch 대시보드를 생성합니다.

1. 중분 채우기 속도: 삽입된 광고로 이어지는 YO 기회의 비율을 $\text{Sum}(\text{YieldOptimization.AdsInserted}) / \text{Sum}(\text{YieldOptimization.BidRequest})$ 보여줍니다. 또는 가용성별 네이티브 지표 $\text{Average}(\text{YieldOptimization.FillRate})$ 을 사용합니다.
2. 시간당 수익: 시간 $\text{Sum}(\text{YieldOptimization.Revenue})$ 경과에 따른 YO의 총 고객당 수익을 보여줍니다.는 계정 전체의 평균 CPM을 $\text{Sum}(\text{YieldOptimization.BidPriceAdsInserted})$ 제공합니다.

3. 채워진 기간:에 `Sum(YieldOptimization.FilledDuration)` 대해 요청된 기간이 실제로 채워진 정도를 `Sum(YieldOptimization.BidDuration)` 보여줍니다.

일반적인 문제

수율 최적화가 트리거되지 않음

증상	Check]를 선택합니다	해결 방법
YO 지표가 전혀 없음	재생 구성에 YO가 구성되어 있습니까?	콘솔 또는 API에서 YO를 활성화합니다.
<code>SkipMinUnfilledDurationNotExceeded</code> 방출	기본 ADS가 대부분의 휴식 시간을 채우고 있음	기본 채우기면 충분함을 낮추 <code>MinimumUnfilledDuration</code> 거나 수락합니다.
<code>SkipPersonalizationTimeoutExceeded</code> 방출	기본 ADS가 너무 느려 YO에 대한 시간이 없습니다.	기본 ADS 지연 시간을 최적화하거나 개인화 제한 시간 SDC 구성을 확인합니다.

입찰 요청 실패

증상	Check]를 선택합니다	해결 방법
<code>BidRequestConfigurationError</code> 지표	템플릿에 필수 필드가 누락되었거나 비어 있음	특정 오류(예: <code>app.bundle is required</code>)가 <code>application.log</code> 있는지 확인합니다. 템플릿을 수정하거나 플레이어 파라미터가 전송되었는지 확인합니다.
APS의 HTTP 204	<code>YieldOptimization.NoBid</code> 증분	APS가 no-bid를 반환했습니다. 오류가 아닙니다. 자리 표시자 값(<code>{UPPER_CASE}</code> 패턴) <code>RAW_BID_REQUEST</code> 이 있는지 검사하고 번들 및 저장소

증상	Check]를 선택합니다	해결 방법
		가 실제 하위 bidfloor인지 확인합니다.
APS의 HTTP 429	YieldOptimization. BidRequestThrottled 증분	APS가 제한되고 있습니다. 요청 속도를 취소합니다. 영구적일 경우 APS 지원팀에 문의하여 TPS 한도를 높이세요.
APS의 HTTP 400	잘못된 요청 JSON	에서 APS 오류 메시지를 확인합니다RAW_BID_RESPONSE . JSON 구조가 OpenRTB 사양과 일치하는지 확인합니다.
APS의 HTTP 500/503	APS 서버 오류 또는 오버로드	일시적인 문제입니다. YieldOptimization.Errors 모니터링. 영구적일 경우 APS 지원팀에 문의하십시오.

반환되었지만 삽입되지 않은 광고

증상	Check]를 선택합니다	해결 방법
AdsReturned > 0이지만 AdsInserted = 0	삽입 중 필터링된 광고	로그FILLED_AVAIL 에 중복된 크리에이티브, 남은 임계값 초과 또는 광고 변형을 찾을 수 없음(트랜스코딩이 완료되지 않음)이 있는지 확인합니다.

APS 오류 코드 참조

코드	의미	일반적인 원인
200	입찰 성공	올바르게 작동합니다.

코드	의미	일반적인 원인
204	입찰 없음	요청이 유효하지만 사용 가능한 입찰이 없습니다. 이는 템플릿의 오류를 나타내지 않습니다.
400	잘못된 요청	잘못된 JSON, OpenRTB 사양당 필수 필드 누락.
500	서버 오류	APS 내부 문제, 서비스 제한, 제한 시간.
503	서비스 오버로드됨	APS 역압. 적극적으로 재시도하지 마십시오.
429	병목 현상 발생	APS가 요청을 수동으로 제한하기 시작했습니다. 영구적일 경우 APS 지원팀에 문의하십시오.

ADS 요청에 대한 MediaTailor 동적 광고 변수

AWS Elemental MediaTailor 는 동적 광고 변수를 사용하여 보기 세션의 정보를 광고 결정 서버(ADS)로 전달합니다. 이 정보는 ADS가 최종 사용자에게 가장 관련성이 높은 광고를 선택하는 데 도움이 됩니다.

이 섹션에서는 동적 광고 변수에 대한 개요와 특정 구현 가이드에 대한 링크를 제공합니다. step-by-step 구성 지침은 아래의 개별 주제를 참조하세요.

동적 변수 유형

MediaTailor는 네 가지 유형의 동적 변수를 지원합니다.

- 세션 변수 - 세션 ID 및 SCTE-35 데이터와 같이 자동으로 생성된 값입니다. [ADS 요청에 대한 MediaTailor 세션 변수](#)을(를) 참조하세요.
- 플레이어 변수 - 비디오 플레이어가 전송한 사용자 지정 파라미터입니다. [ADS 요청에 대한 MediaTailor 플레이어 변수](#)을(를) 참조하세요.

- 구성 별칭이 있는 도메인 변수 - 다중 오리진 구성을 위한 동적 URL 도메인입니다.
- 서비스 변수 - MediaTailor에서 세션 수준 동작을 제어하는 예약 `aws.*` 파라미터입니다. [서비스 변수](#)(를) 참조하세요.
- 구성 별칭 - 동적 변수 대체를 위해 미리 정의된 매핑입니다. [구성 별칭](#)(을) 참조하세요.

일반 사용 사례

동적 광고 변수를 사용하여 다음을 수행합니다.

- 최종 사용자 인구 통계 및 기본 설정을 ADS에 전달
- 지리적 위치에 따라 요청을 다른 오리진으로 라우팅
- MediaPackage 통합을 사용하여 시간 이동 보기 활성화
- A/B 테스트 및 장애 조치 시나리오 구현

다음 섹션에서는 MediaTailor에서 동적 광고 변수를 사용하는 방법에 대한 추가 세부 정보를 제공합니다.

주제

- [ADS 요청에 대한 MediaTailor 세션 변수](#)
- [ADS 요청에 대한 MediaTailor 플레이어 변수](#)
- [여러 콘텐츠 소스에 대한 MediaTailor 도메인 변수](#)
- [세션 제어를 위한 MediaTailor 서비스 변수](#)
- [MediaTailor 구성 별칭 개요](#)
- [파라미터를 ADS에 전달하는 MediaTailor](#)
- [ADS 및 오리진에 대한 MediaTailor 파라미터 라우팅](#)
- [MediaTailor 및 MediaPackage 시간 이동 보기 통합](#)
- [MediaTailor 파라미터 세션 동작 및 지속성](#)
- [MediaTailor 파라미터 참조 및 제한 사항](#)
- [MediaTailor 파라미터 문제 해결 가이드](#)
- [MediaTailor 구성 별칭 문제 해결 가이드](#)

파라미터 형식 지정 요구 사항 및 문제 해결은 [MediaTailor 파라미터 참조 및 제한 사항](#) 및 섹션을 참조하세요. [MediaTailor 파라미터 문제 해결 가이드](#).

ADS 요청에 대한 MediaTailor 세션 변수

AWS Elemental MediaTailor 는 템플릿 ADS URL의이 섹션에 나열된 변수 중 하나 이상을 지정 AWS Elemental MediaTailor 하도록을 구성할 때 세션 데이터를 Ad Decision Server(ADS)로 전송합니다. 개별 변수를 사용할 수 있으며 여러 변수를 연결하여 단일 값을 생성할 수 있습니다. MediaTailor는 일부 값을 생성하고 매니페스트 및 플레이어의 세션 초기화 요청과 같은 소스에서 나머지 값을 가져옵니다.

다음 표에서는 템플릿 ADS 요청 URL 구성에 사용할 수 있는 세션 데이터 변수를 설명합니다. 표에 나열된 섹션 번호는 케이블 통신 엔지니어 협회(SCTE)-35 사양의 2019a 버전, [디지털 프로그램 삽입 대기열 메시지](#), 광고 미리 가져오기에 대한 자세한 내용은 섹션을 참조하세요 [광고 미리 가져오기](#).

이름	광고 미리 가져오기에 사용 가능	SCTE-35 사양 섹션	설명
[avail.index]	예		인덱스에서 광고 가능성의 위치를 나타내는 숫자입니다. 재생 세션이 시작될 때 MediaTailor는 매니페스트에 모든 광고 시간의 인덱스를 생성하고 세션의 나머지 기간 동안 인덱스를 저장합니다. MediaTailor가 ADS에 가용을 채우도록 요청하면 여기에는 광고 가능 인덱스 번호가 포함됩니다. 이 파라미터를 사용하면 ADS가 경쟁 배제 및 빈도 제한과 같은 기능을 사용하여 광고 선택을 개선할 수 있습니다.
[avail.random]	예		MediaTailor가 ADS에 대한 각 요청에 대해 생성하는 0에서 10,000,000,000 사이의 난수입니다. 일부 광고 서버는 이 파라미터를 사용하여 광고를 경쟁사와 분리하는 것과 같은 기능을 활성화합니다.
[scte.archive_allowed_flag]	예	10.3.3.1	선택적 부울 값입니다. 이 값이 0이면 세그먼트에 레코딩 제한이 적용됩니다. 이 값이 1이면 세그먼트에 레코딩 제한이 적용되지 않습니다.
[scte.avail_num]	예	9.7.2.1	MediaTailor가 SCTE-35 필드에서 긴 숫자avail_num로 구문 분석한 값입니다. MediaTailor는이 값을 사용하여 선형 광고 가능 시간을 지정할 수 있습니다.

이름	광고 미리 가져오기에 사용 가능	SCTE-35 사양 섹션	설명
			값은 정수여야 합니다.
[scte.ava ils_expec ted]	예	9.7.2.1	현재 이벤트 내의 예상 가용 수를 제공하는 선택적 긴 값입니다.
[scte.del ivery_not _restrict ed_flag]	예	10.3.3.1	선택적 부울 값입니다. 이 값이 0이면 다음 5비트가 예약됩니다. 이 값이 1인 경우 다음 5비트는 SCTE-35 사양에 설명된 의미를 따릅니다.
[scte.dev ice_restr ictions]	예	10.3.3.1	사전 정의된 독립 및 비계층적 디바이스 그룹 3개에 신호를 보내는 선택적 정수 값입니다. 이 변수에 대한 자세한 내용은 SCTE-35 사양의 segments_expected 설명을 참조하세요.
[scte.eve nt_id]	예	9.1 및 9.7.2.1	MediaTailor가 SCTE-35 필드에서 긴 숫자splice_event_id 로 구문 분석한 값입니다. MediaTailor는 이 값을 사용하여 선택적 광고 가능 시간을 지정하거나 광고 포드 위치와 같은 광고 서버 쿼리 문자열을 채웁니다. 값은 정수여야 합니다.
[scte.no_ regional_ blackout_ flag]	예	10.3.3.1	선택적 부울 값입니다. 이 값이 0이면 리전 블랙아웃 제한이 세그먼트에 적용됩니다. 이 값이 1이면 리전 블랙아웃 제한이 세그먼트에 적용되지 않습니다.
[scte.seg ment_num]	예	10.3.3.1	세그먼트 모음 내의 세그먼트에 번호를 지정하는 선택적 정수 값입니다. 이 변수에 대한 자세한 내용은 SCTE-35 사양의 segment_num 설명을 참조하세요.

이름	광고 미리 가져오기에 사용 가능	SCTE-35 사양 섹션	설명
[scte.segmentation_event_id]	예	10.3.3.1	MediaTailor는 이 변수를 로 노출합니다 scte.event_id .
[scte.segmentation_type_id]	예	10.3.3.1	분할 유형을 지정하는 선택적 8비트 정수 값입니다. 이 변수에 대한 자세한 내용은 SCTE-35 사양의 segmentation_type_id 설명을 참조하세요.

이름	광고 미리 가져오기에 사용 가능	SCTE-35 사양 섹션	설명
[scte.segmentation_upid]	segmentation_upid_type : 예 private_data : 예	segmentation_upid: 10.3.3.1 관리형 프라이빗 UPID: 10.3.3.3	SCTE-35 segmentation_upid 요소에 해당합니다. segmentation_upid 요소에는 segmentation_upid_type 및 segmentation_upid_length가 포함되어 있습니다. MediaTailor는 다음 segmentation_upid 유형을 지원합니다. • ADS 정보(0x0E) - 광고 정보. 자세한 내용은 SCTE-35 사양의 segmentation_upid 설명을 참조하세요. • Managed Private UPID(0x0C) - SCTE-35 사양에 정의된 Managed Private UPID(MPU) 구조입니다. MediaTailor는 바이너리 또는 DASH XML SCTE 표현을 지원합니다. 포드버스터 워크플로에서이 구조를 사용할 수 있습니다. 이렇게 하려면 32비트(4바이트)를 지정format_identifier 하고 private_data 속성에 다음 파라미터를 포함합니다. <pre>ABCD{"assetId": " my_program ", "cueData": {"cueType": " theAdType ", "key": " pb", "value": " 123456"}}</pre> MediaTailor는 이전 JSON의 값을 구문 분석하여, scte.segmentation_upid.assetId, scte.segmentation_upid.cueData.key 및 scte.segmentation_upid.cueData.value 동적 변수로 전달합니다. • 사용자 정의(0x01) - 사용자 정의 구조입니다. 자세한 내용은 SCTE-35 사양의 segmentation_upid 설명을 참조하세요.

이름	광고 미리 가져오기에 사용 가능	SCTE-35 사양 섹션	설명
[scte.segmentation_upid.assetId]	예		포드버스터 워크플로를 segmentation_upid_type 위해 Managed Private UPID(0xC)와 함께 사용됩니다. MediaTailor는 MPU의 private_data JSON 구조의 assetId 파라미터에서 이 값을 추출합니다. 자세한 내용은 Managed Private UPID JSON structure for a podbustar workflow 단원을 참조하십시오.
[scte.segmentation_upid.cueData.key]	예		포드버스터 워크플로를 segmentation_upid_type 위해 Managed Private UPID(0xC)와 함께 사용됩니다. MediaTailor는 MPU의 private_data JSON 구조의 cueData.key 파라미터에서 이 값을 추출합니다. 자세한 내용은 Managed Private UPID JSON structure for a podbustar workflow 단원을 참조하십시오.
[scte.segmentation_upid.cueData.value]	예		포드버스터 워크플로를 segmentation_upid_type 위해 Managed Private UPID(0xC)와 함께 사용됩니다. MediaTailor는 MPU의 private_data JSON 구조의 cueData.key 파라미터에서 이 값을 추출합니다. 자세한 내용은 Managed Private UPID JSON structure for a podbustar workflow 단원을 참조하십시오. 값은 문자열일 수 있습니다.

이름	광고 미리 가져오기에 사용 가능	SCTE-35 사양 섹션	설명
[scte.segmentation_upid.private_data.{index}]	예		대상 광고 워크플로를 segmentation_upid_type 위해 Managed Private UPID(0xC)와 함께 사용됩니다. MediaTailor는 콜론으로 구분된 분할 UPID 토큰을 분할하고 인덱싱된 세션 변수를 생성합니다. 인덱스는 콜론으로 구분된 목록의 위치에 해당하며 초기 콜론의 선행 공백은 무시합니다. 예를 들어 인 경우 다음을 segmentation_upid = ":3213214:2313321/5:3943" 수행합니다. [scte.segmentation_upid.private_data.0] = 3213214 [scte.segmentation_upid.private_data.1] = 2313321/5 [scte.segmentation_upid.private_data.2] = 3943 값은 문자열일 수 있습니다.
[scte.segments_expected]	예	10.3.3.1	세그먼트 모음 내의 개별 세그먼트의 예상 수를 제공하는 선택적 정수 값입니다. 이 변수에 대한 자세한 내용은 SCTE-35 사양의 segments_expected 설명을 참조하세요.
[scte.sub_segment_num]	예	10.3.3.1	하위 세그먼트 모음 내의 특정 하위 세그먼트를 식별하는 선택적 정수 값입니다. 이 변수에 대한 자세한 내용은 SCTE-35 사양의 sub_segment_num 설명을 참조하세요.
[scte.sub_segments_expected]	예	10.3.3.1	하위 세그먼트 모음 내의 개별 하위 세그먼트의 예상 개수를 제공하는 선택적 정수 값입니다. 이 변수에 대한 자세한 내용은 SCTE-35 사양의 sub_segments_expected 설명을 참조하세요.

이름	광고 미리 가져오기에 사용 가능	SCTE-35 사양 섹션	설명
[scte.unique_program_id]	예	9.7.2.1	SCTE-35 splice_insert 필드에서 MediaTailor가 구문 분석한 정수 값입니다unique_program_id . ADS는 고유 프로그램 ID(UPID)를 사용하여 라이브 선형 스트림에 대한 프로그램 수준의 광고 타겟팅을 제공합니다. SCTE-35 명령이 스플라이스 삽입이 아닌 경우 MediaTailor는 이를 빈 값으로 설정합니다. 값은 정수여야 합니다.
[session.avail_duration_ms]	예		광고 가용성 슬롯의 밀리초 단위 기간입니다. 기본값은 300,000ms입니다. 다음과 같이 입력 매니페스트에서 기간 값을 AWS Elemental MediaTailor 가져옵니다. - HLS의 경우: MediaTailor는 #EXT-X-CUE-OUT: DURATION 또는 #EXT-X-DATERANGE 태그의 값에서 기간을 가져옵니다. 입력 매니페스트에 해당 태그의 광고 시간에 대해 null, 유효하지 않음 또는 0 지속 시간이 있는 경우 MediaTailor는 기본값을 사용합니다. - DASH의 경우: MediaTailor는 이벤트 기간이 지정된 경우 이벤트 기간에서 기간 값을 가져옵니다. 그렇지 않으면 기본값이 사용됩니다. - VOD의 경우: VOD 스트림이 롤링 전 광고 호출을 트리거할 때 매니페스트에 지속 시간 값이 있는 SCTE 메시징이 포함되지 않은 경우 MediaTailor는 기본 지속 시간 값을 포함하여 [session.avail_duration_ms]의 지속 시간을 입력하지 않습니다.
[session.avail_duration_secs]	예		광고 가용성 슬롯 또는 광고 가능 시간의 초 단위 지속 시간으로, 가장 가까운 초로 반올림됩니다. MediaTailor 는를 결정하는 것과 동일한 방식으로이 값을 결정합니다[session.avail_duration_ms] .

이름	광고 미리 가져오기에 사용 가능	SCTE-35 사양 섹션	설명
[session.client_ip]	아니요		MediaTailor 요청이 시작된 원격 IP 주소입니다. X-forwarded-for 헤더가 설정된 경우 해당 값은 MediaTailor가 사용하는 값입니다client_ip .
[session.id]	아니요		현재 재생 세션의 고유한 숫자 식별자입니다. 플레이어가 세션에 대해 실행하는 모든 요청의 경우 id가 동일하므로 이 id는 단일 시청에 대한 여러 요청을 관련시키기 위한 ADS 필드에 사용될 수 있습니다.
[session.referrer]	아니요		일반적으로 비디오 플레이어에서 호스팅하는 페이지의 URL입니다. MediaTailor는 이 변수를 플레이어가 MediaTailor에 요청할 때 사용한 Referer 헤더 값으로 설정합니다. 플레이어가 이 헤더를 제공하지 않으면 MediaTailor는 [session.referrer] 비워 둡니다. 매니페스트 엔드포인트 앞에 콘텐츠 전송 네트워크(CDN) 또는 프록시를 사용하고 이 변수를 표시하려면 여기에서 플레이어의 올바른 헤더를 프록시합니다.
[session.user_agent]	아니요		MediaTailor가 플레이어의 세션 초기화 요청에서 받은 User-Agent 헤더입니다. 매니페스트 엔드포인트 앞에서 CDN 또는 프록시를 사용 중인 경우 여기 플레이어에서 올바른 헤더에 프록시를 설정해야 합니다.
[session.uuid]	아니요		의 대안입니다 [session.id] . 이는 다음과 같은 현재 재생 세션의 고유한 식별자입니다. e039fd39-09f0-46b2-aca9-9871cc116cde

이름	광고 미리 가져오기에 사용 가능	SCTE-35 사양 섹션	설명
[avail.source_content_time_epoch_ms]	아니요		HLS의 경우 값은 가용 구간을 시작한 오리진 세그먼트의 PDT입니다. DASH의 경우 값은 <Period> 포함된 <SupplementalProperty> urn:scte:dash:utc-time 의 입니다<EventStream> . - 프리롤 HLS 또는 DASH 광고의 경우 첫 번째 HLS 세그먼트의 PDT와 첫 번째 <SupplementalProperty> urn:scte:dash:utc-time 의 PDT[avail.source_content_time_epoch_ms] 입니다<Period>. 매니페스트 기간이 짧은 라이브 스트림의 경우 값은 스트림 시청을 시작하는 시점에 따라 뷰어마다 다릅니다. 전체 이벤트 DVR 창이 있는 라이브스트림의 경우 값은 각 뷰어에 대해 동일합니다. - 가용 억제로 인한 부분 가용의 경우 [avail.source_content_time_epoch_ms] 는 가용을 시작한 소스 콘텐츠 세그먼트의 PDT입니다. 예를 들어 MediaTailor가 가용 시간의 처음 20초를 제거해도 [avail.source_content_time_epoch_ms] 는 여전히 20초만큼 이동되지 않고 원래 PDT로 설정됩니다. - 미리 가져오기 ADS 요청의 경우 [avail.source_content_time_epoch_ms] 는 비어 있지만 고객은 미리 가져오기 일정 검색 동적 변수를 사용하여 설정할 수 있습니다.

Example

ADS가 deviceSession이라는 쿼리 파라미터에 고유한 세션 식별자를 사용하여 전달될 것을 요구하는 경우 AWS Elemental MediaTailor 의 템플릿 ADS URL은 다음과 같이 보일 수 있습니다.

```
https://my.ads.server.com/path?deviceSession=[session.id]
```

AWS Elemental MediaTailor 는 각 스트림에 대한 고유 식별자를 자동으로 생성하고 대신 식별자를 입력합니다. `session.id` 식별자가 인 경우 MediaTailor가 ADS에 보내는 1234567 최종 요청은 다음과 같습니다.

```
https://my.ads.server.com/path?deviceSession=1234567
```

ADS에 여러 쿼리 파라미터를 전달해야 하는 경우의 템플릿 ADS URL은 다음과 같은 AWS Elemental MediaTailor 수 있습니다.

```
https://my.ads.server.com/sample?
e=[scte.avails_expected]&f=[scte.segment_num]&g=[scte.segments_expected]&h=[scte.sub_segment_num]
```

다음 DASH 마커 예제 XML 조각은 사용 방법을 보여줍니다 `scte35:SpliceInsert`.

```
<Period start="PT444806.040S" id="123456" duration="PT15.000S">
  <EventStream timescale="90000" schemeIdUri="urn:scte:scte35:2013:xml">
    <Event duration="1350000">
      <scte35:SpliceInfoSection protocolVersion="0" ptsAdjustment="180832" tier="4095">
        <scte35:SpliceInsert spliceEventId="1234567890"
spliceEventCancelIndicator="false" outOfNetworkIndicator="true"
spliceImmediateFlag="false" uniqueProgramId="1" availNum="1" availsExpected="1">
          <scte35:Program><scte35:SpliceTime ptsTime="5672624400"/></scte35:Program>
          <scte35:BreakDuration autoReturn="true" duration="1350000"/>
        </scte35:SpliceInsert>
      </scte35:SpliceInfoSection>
    </Event>
  </EventStream>
</Period>
```

다음 DASH 마커 예제 XML 조각은 사용 방법을 보여줍니다 `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>
  </EventStream>
</Period>
```



```

    </scte35:SegmentationDescriptor>
  </scte35:SpliceInfoSection>
</Event>

```

다음 DASH 마커 예제 XML 조각은 사용 방법을 보여줍니다 `scte35:Binary`.

```

<Period start="PT444806.040S" id="123456" duration="PT15.000S">
  <EventStream schemeIdUri="urn:scte:scte35:2014:xml+bin" timescale="1">
    <Event presentationTime="1541436240" duration="24" id="29">
      <scte35:Signal xmlns="http://www.scte.org/schemas/35/2016">
        <scte35:Binary>/DAhAAAAAAAAAAP/wEAUAAAHaf+9/fgAg9YDAAAAAAAAA25aoh</Binary>
      </scte35:Signal>
    </Event>
    <Event presentationTime="1541436360" duration="24" id="30">
      <scte35:Signal xmlns="http://www.scte.org/schemas/35/2016">
        <scte35:Binary>QW5vdGhlciB0ZXN0IHN0cmVuZyBmb3IgZW5jb2RpbmcgdG8gQmFzZTY0IGVuY29kZWQgYmluYXJ5Lg==
      </scte35:Signal>
    </Event>
  </EventStream>
</Period>

```

다음 HLS 태그 예제에서는를 사용하는 방법을 보여줍니다 `EXT-X-DATERANGE`.

```

#EXT-X-DATERANGE:ID="splice-6FFFFFF0",START-DATE="2014-03-05T11:
15:00Z",PLANNED-DURATION=59.993,SCTE35-OUT=0xFC002F0000000000FF0
00014056FFFFFFF000E011622DCAFF0000526362000000000000A0008029896F50
0000087000000000

```

다음 HLS 태그 예제에서는를 사용하는 방법을 보여줍니다 `EXT-X-CUE-OUT`.

```

#EXT-OATCLS-SCTE35:/DA0AAAAAAAAAAAAABQb+ADAQ6QAeAhxDVUVJQAAA03/PAAEUrEoICAAAAAg
+2UBNAAANvrtoQ==
#EXT-X-ASSET:CAID=0x00000000020FB6501
#EXT-X-CUE-OUT:201.467

```

다음 HLS 태그 예제에서는를 사용하는 방법을 보여줍니다 `EXT-X-SPLICEPOINT-SCTE35`.

```

#EXT-X-SPLICEPOINT-SCTE35:/DA9AAAAAAAAAAP/wBQb+uYbZqwAnAiVDVUVJAAAKqX//
AAEjW4AMEU1EU05CMDaxMTMyMjE5M190NAAAmXz5JA==

```

다음 예제에서는 `scte35:Binary` 디코딩을 사용하는 방법을 보여줍니다.

```

{
  "table_id": 252,
  "section_syntax_indicator": false,
  "private_indicator": false,
  "section_length": 33,
  "protocol_version": 0,
  "encrypted_packet": false,
  "encryption_algorithm": 0,
  "pts_adjustment": 0,
  "cw_index": 0,
  "tier": "0xFFF",
  "splice_command_length": 16,
  "splice_command_type": 5,
  "splice_command": {
    "splice_event_id": 448,
    "splice_event_cancel_indicator": false,
    "out_of_network_indicator": true,
    "program_splice_flag": true,
    "duration_flag": true,
    "splice_immediate_flag": false,
    "utc_splice_time": {
      "time_specified_flag": false,
      "pts_time": null
    },
  },
  "component_count": 0,
  "components": null,
  "break_duration": {
    "auto_return": false,
    "duration": {
      "pts_time": 2160000,
      "wall_clock_seconds": 24.0,
      "wall_clock_time": "00:00:24:000000"
    }
  },
  },
  "unique_program_id": 49152,
  "avail_num": 0,
  "avails_expected": 0,
  "segment_num": 0,
  "segments_expected": 0,
  "sub_segment_num": 0,
  "sub_segments_expected": 0,
  },
  "splice_descriptor_loop_length": 0,

```

```

"splice_descriptors": null,
"Scte35Exception": {
  "parse_status": "SCTE-35 cue parsing completed with 0 errors.",
  "error_messages": [],
  "table_id": 252,
  "splice_command_type": 5
}
}

```

ADS 요청에 대한 MediaTailor 플레이어 변수

AWS Elemental MediaTailor 는 템플릿 ADS URL에서 `player_params.<query_parameter_name>` 변수를 지정하도록 AWS Elemental MediaTailor 를 구성할 때 플레이어에서 수신한 데이터를 ADS로 전송합니다. 예를 들어 플레이어가 요청에 `user_id` 이름이 인 쿼리 파라미터를 MediaTailor로 보내는 경우 ADS 요청에 해당 데이터를 전달하기 위해를 ADS URL 구성[`player_params.user_id`]에 포함합니다.

이렇게 하면 ADS 요청에 포함되어 있는 쿼리 파라미터를 제어할 수 있습니다. 일반적으로, ADS가 인식하는 특수 쿼리 파라미터를 ADS 요청 URL에 추가하고 키-값 페어를 파라미터의 값으로 제공합니다.

다음 절차에 사용된 예제는 다음 키-값 페어를 사용합니다.

- 값이 1인 param1:
- 값이 2인 param2:

쿼리 파라미터를 키-값 페어로 추가하려면

1. 에서 파라미터를 참조하도록 ADS 요청 템플릿 URL을 AWS Elemental MediaTailor구성합니다. 다음 URL은 예제 파라미터가 포함되어 있음을 보여줍니다.

```
https://my.ads.com/path?param1=[player_params.param1]&param2=[player_params.param2]
```

2. (선택 사항) 서버 측 광고 추적 보고의 경우 플레이어의 키 값 페어를 URL 인코딩합니다. MediaTailor는 세션 초기화 요청을 수신하면 값을 ADS 요청 URL로 대체하기 전에 값을 한 번 URL 디코딩합니다.

Note

ADS가 URL 인코딩 값을 요구하는 경우 플레이어에서 값을 두 번 URL 인코딩합니다. 이렇게 하면 MediaTailor에서 수행한 디코딩으로 인해 ADS에 대해 한 번 인코딩된 값이 생성됩니다.

예를 들어 ADS에 전송된 값의 디코딩된 표현이 param1=value1:¶m2=value2:인 경우 URL 인코딩된 표현은 param1=value1%3A¶m2=value2%3A입니다.

3. 플레이어의 세션 초기화 호출에서 키-값 페어를 단일 쿼리 파라미터의 값으로 MediaTailor에 전달합니다. 다음 예제 호출은 서버 측 및 클라이언트 측 광고 추적 보고에 대해 예제 키-값 페어를 제공합니다.

- 서버 측 광고 추적 보고에 대한 예제 요청 - URL 인코딩 페어 사용

HLS:

```
<master>.m3u8?ads.param1=value1%3A&ads.param2=value2%3A
```

DASH:

```
<manifest>.mpd?ads.param1=value1%3A&ads.param2=value2%3A
```

- 클라이언트 측 광고 추적 보고에 대한 예제 요청 - URL 인코딩 사용하지 않음

HLS:

```
POST <master>.m3u8
{
  "adsParams": {
    "param1": "value1:",
    "param2": "value2:"
  }
}
```

DASH:

```
POST <manifest>.mpd
{
```

```

    "adsParams": {
      "param1": "value1:",
      "param2": "value2:"
    }
  }
}

```

서버 측 보고의 경우 MediaTailor는 플레이어 요청이 수신될 때 파라미터를 디코딩합니다. 클라이언트 측 보고의 경우 JSON 페이로드에서 수신된 파라미터는 변경되지 않습니다. MediaTailor는 ADS에 다음 요청을 보냅니다.

```
https://my.ads.com/<path>?param1=value1:&param2=value2:
```

이러한 방식으로 param1 및 param2 키-값 페어는 최고 수준의 쿼리 파라미터로 ADS 요청에 포함됩니다.

여러 콘텐츠 소스에 대한 MediaTailor 도메인 변수

AWS Elemental MediaTailor 동적 도메인 변수를 사용하면 구성의 플레이어 파라미터와 함께 URL my-ads-server.com http://my-ads-server.com 같은 여러 도메인을 사용할 수 있습니다. 이렇게 하면 단일 구성에서 둘 이상의 콘텐츠 소스 또는 광고 결정 서버(ADS)를 사용할 수 있습니다.

URI가 포함된 모든 파라미터와 함께 도메인 변수를 사용할 수 있습니다.

- AdDecisionServerUrl
- AdSegmentUrlPrefix
- ContentSegmentUrlPrefix
- LivePreroll.AdDecisionServerUrl
- VideoContentSourceUrl

도메인 변수는 구성 별칭과 함께 동적 변수 교체를 수행하는 데 사용됩니다. 구성 별칭은 동적 도메인 구성에 사용되는 플레이어 파라미터에 별칭 및 값 집합을 매핑합니다. 설정 절차는 단원을 참조하십시오 [오MediaTailor를 사용하여 구성 별칭 생성 및 사용](#). 자세한 참조 정보는 섹션을 참조하세요 [MediaTailor 구성 별칭 개요](#).

세션 제어를 위한 MediaTailor 서비스 변수

AWS Elemental MediaTailor 는 세션 수준 동작을 제어하는 서비스 변수에 대한 aws. 쿼리 파라미터 네임스페이스를 예약합니다. ads. 파라미터(ADS로 전달됨) 및 manifest. 파라미터(개인화된 매니

페스트 URLs에 추가됨)와 달리 `aws.` 파라미터는 MediaTailor에서 직접 사용되며 오리진 서버 또는 ADS로 전달되지 않습니다.

지원되는 파라미터

다음 표에는 세션 수준 동작을 제어하는 데 사용할 수 있는 `aws.*` 파라미터가 나열되어 있습니다.

파라미터	유형	값	기본값	설명
<code>aws.startTime</code>	ISO 8601 타임 스탬프	예: 2026-06-17T10:00:00Z	설정되지 않음(라이브 엣지 조인)	DVR 창 의 특정 지점에서 세션을 시작합니다. MediaTailor는 타임스탬프를 가장 가까운 세그먼트 경계로 확인하고 HLS 매니페스트 <code>EXT-X-START:TIME-OFFSET</code> 에서를 내보냅니다.
<code>aws.preroll</code>	문자열 열거형	enabled, disabled (대/소문자 비구분)	enabled	세션에 대해 프리롤 광고 삽입이 발생하는지 여부를 제어합니다. disabled인 경우 재생 구성에 있더라도 프리롤이 억제됩니다. <code>LivePreRollConfiguration</code> .
<code>aws.overlayAvails</code>	문자열 열거형	on, off (대/소문자 비구분)	on	오버레이(비선형) 광고 시간이 세션에 대해 처리되는지 여부를 제어합니다. off인 경우 소스 매니페스트의 오버레이 광고 마커는 무시되고 오버레이 광고는 삽입되지 않습니다.
<code>aws.logMode</code>	문자열 열거형	DEBUG, DISABLED	DISABLED	세션에 대한 상세 디버그 로깅을 활성화합니다. 로 설정하면 DEBUG MediaTailor는 문제 해결을 위해 자세한 세션 로그를 CloudWatch Logs로 내보냅니다.
<code>aws.availSuppressionMode</code>	문자열 열거형	OFF, BEHIND_LIVE_EDGE ,	OFF	라이브 엣지에 상대적인 위치를 기반으로 광고 가능 구간이 억제되는지 여부를 제어합니다.

파라미터	유형	값	기본값	설명
		AFTER_LIVE_EDGE (대/소문자 비구분)		
aws.availSuppressionValue	지속 시간	HH:MM:SS 형식 (예: 00:00:10)	설정되지 않음	가용성 억제 기간입니다. availSuppressionMode 이 BEHIND_LIVE_EDGE 또는 일 때 필요합니다AFTER_LIVE_EDGE .
aws.availSuppressionFillPolicy	문자열 열 열거형	FULL_AVAI L_ONLY , PARTIAL_AVAI L (대/소문자 비구분)	FULL_AVAI L_ONLY	모드가 이면 부분적으로 억제된 가용 구간이 채워지는지 여부를 AFTER_LIVE_EDGE 제어합니다.

aws.startTime

aws.startTime이 설정되면 MediaTailor는 지정된 프로그램 날짜-시간에 가장 가까운 세그먼트 경계에서 세션을 시작합니다.

용도

매니페스트 요청에서 쿼리 파라미터aws.startTime로 전달:

```
GET /v1/master/{hashed-account-id}/{origin-id}/{asset}.m3u8?
aws.startTime=2026-06-17T10:00:00Z
```

또는 명시적 세션 초기화에서 aws. 접두사 없이 최상위 필드로 전달합니다.

```
POST /v1/session/{hashed-account-id}/{origin-id}/{asset}.m3u8

{
  "startTime": "2026-06-17T10:00:00Z"
}
```

요구 사항

에는 aws.startTime다음 요구 사항이 적용됩니다.

- 소스 매니페스트는 세그먼트에 EXT-X-PROGRAM-DATE-TIME (PDT)를 포함해야 합니다. PDT가 없으면가 확인할 `aws.startTime` 수 없으며 무시됩니다.
- HLS 라이브 세션(SSAI 및 SGAI 모두)에만 적용됩니다.

동작

다음 표에서는가 다양한 시나리오에서 어떻게 `aws.startTime` 작동하는지 설명합니다.

시나리오	결과
DVR 기간 내 타임스탬프	가장 가까운 세그먼트 경계에서 시작하여 내보냅니다. EXT-X-START
DVR 기간보다 오래된 타임스탬프	창 헤드에서 $3 \times \text{targetDuration}$ 으로 고정
라이브 엣지의 $3 \times \text{targetDuration}$ 이내 타임스탬프	를 라이브 엣지 조인으로 취급(없음EXT-X-START)
라이브 엣지 또는 그 이후의 타임스탬프	일반 라이브 엣지 조인
ISO 8601 형식이 잘못되었거나 아님	무시됨 - 라이브 엣지 조인으로 돌아가고 오류가 기록됨
매니페스트에 PDT가 없음	무시됨 - 라이브 엣지 조인으로 돌아가고 오류가 기록됨
파라미터 생략됨	기본값: 일반 라이브 엣지 조인

클램핑 동작

시작점이 세션 중 DVR 기간을 벗어나는 경우 EXT-X-START는 창 헤드에서 $3 \times \text{targetDuration}$ 으로 고정됩니다. $3 \times \text{targetDuration}$ 버퍼는 RFC 8216 §6.3.3 플레이어 버퍼링 권장 사항에 부합합니다.

Example `aws.startTime`을 사용한 매니페스트 출력

플레이어가 로 세션을 초기화`aws.startTime=2026-06-17T10:00:00Z`하고 확인된 오프셋이 라이브 엣지에서 120초가 되면 MediaTailor는 다음을 내보냅니다.

```
#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
...
```

는 플레이어에게 라이브 엣지 120초 후 요청된 시작 시간에 가장 가까운 세그먼트 경계에서 재생을 시작하도록 TIME-OFFSET=-120.120 지시합니다.

제한 사항

다음과 같은 제한이 적용됩니다.

- 세션 초기화 후에는이 파라미터를 변경할 수 없습니다.
- 소스 매니페스트EXT-X-PROGRAM-DATE-TIME에 필요합니다.
- EXT-X-START는 플레이어 힌트입니다. MediaTailor는 모든 플레이어가 이를 준수한다고 보장할 수 없습니다.

aws.preroll

aws.preroll=disabled인 경우 MediaTailor는 재생 구성에 있더라도 세션에 대한 프리롤 광고 삽입을 억제합니다LivePreRollConfiguration.

용도

매니페스트 요청에서 쿼리 파라미터aws.preroll로 전달:

```
GET /v1/master/{hashed-account-id}/{origin-id}/{asset}.m3u8?aws.preroll=disabled
```

또는 명시적 세션 초기화에서 aws. 접두사 없이 최상위 필드로 전달합니다.

```
POST /v1/session/{hashed-account-id}/{origin-id}/{asset}.m3u8

{
  "preroll": "disabled"
}
```

요구 사항

에는 `aws.preroll` 다음 요구 사항이 적용됩니다.

- 이 파라미터를 적용하려면 재생 구성 `LivePreRollConfiguration`에 있어야 합니다. 프리롤이 구성되지 않은 경우 이 파라미터를 설정해도 아무런 효과가 없습니다.
- HLS 라이브 세션(SSAI 및 SGAI 모두)에 적용됩니다.

동작

다음 표에서는 `aws.preroll` 작동하는 방식을 설명합니다.

값	결과
<code>enabled</code> (또는 생략됨)	프리롤이 평소처럼 삽입됨
<code>disabled</code>	이 세션에 대해 프리롤이 금지됨
잘못된 값	오류로 로깅되고 로 처리됨 <code>enabled</code>

제한 사항

세션 초기화 후에는 이 파라미터를 변경할 수 없습니다.

`aws.overlayAvails`

`aws.overlayAvails=off`인 경우 MediaTailor는 소스 매니페스트에서 오버레이(비선형) 광고 마커를 무시하고 세션에 오버레이 광고를 삽입하지 않습니다.

용도

매니페스트 요청에서 쿼리 파라미터 `aws.overlayAvails`로 전달합니다.

```
GET /v1/master/{hashed-account-id}/{origin-id}/{asset}.m3u8?aws.overlayAvails=off
```

요구 사항

에는 `aws.overlayAvails` 다음 요구 사항이 적용됩니다.

- 이 파라미터가 효과를 발휘하려면 소스 매니페스트에 오버레이 광고 마커(예: 오버레이 분할 유형의 SCTE-35 이벤트)가 포함되어야 합니다.

- HLS 및 DASH, 라이브 및 VOD 세션(SSAI 및 SGAI 모두)에 적용됩니다.

동작

다음 표에서는 `aws.overlayAvails` 작동하는 방식을 설명합니다.

값	결과
on (또는 생략됨)	오버레이 가용성이 처리되고 광고가 평소와 같이 삽입됩니다.
off	오버레이 광고 마커가 무시되고 오버레이 광고가 삽입되지 않음
잘못된 값	오류로 로깅됨, 지정되지 않음으로 처리됨(기본값: on)

제한 사항

세션 초기화 후에는 이 파라미터를 변경할 수 없습니다.

aws.logMode

`aws.logMode=DEBUG`인 경우 MediaTailor는 세션에 대한 상세 디버그 로깅을 활성화합니다. 디버그 로그는 CloudWatch Logs로 보내지고 매니페스트 개인화, 광고 결정 서버 요청 및 세션 상태에 대한 자세한 정보를 제공합니다. 이는 광고 삽입 문제를 해결하는 데 유용합니다.

용도

매니페스트 요청에서 쿼리 파라미터 `aws.logMode`로 전달합니다.

```
GET /v1/master/{hashed-account-id}/{origin-id}/{asset}.m3u8?aws.logMode=DEBUG
```

요구 사항

에는 `aws.logMode` 다음 요구 사항이 적용됩니다.

- 디버그 로그를 보내려면 재생 구성에 로깅이 활성화(`PercentEnabled > 0` 또는 `EnabledLoggingStrategies` 구성)되어 있어야 합니다.

- 디버그 로깅은 과도한 로그 볼륨을 방지하기 위해 고객별로 속도가 제한됩니다.

동작

다음 표에서는가 `aws.logMode` 작동하는 방식을 설명합니다.

값	결과
DEBUG	세션에 대해 생성된 세부 정보 디버그 로그
DISABLED (또는 생략됨)	일반 로깅 동작(재생 구성 설정 기준)
잘못된 값	오류 발생, 세션 초기화 실패

제한 사항

다음과 같은 제한이 적용됩니다.

- 세션 초기화 후에는이 파라미터를 변경할 수 없습니다.
- 값은 대/소문자를 구분합니다(DEBUG가 아님debug).

aws.availSuppressionMode

라이브 엣지에 상대적인 위치를 기반으로 광고 가능 구간이 억제되는지 여부를 제어합니다. 예를 들어 이벤트 중간에 라이브 스트림에 조인할 때 최종 사용자가 이미 통과한 광고 시간을 채우지 않도록 하려면 이를 사용하여 라이브 엣지 뒤 또는 뒤의 기간 내에 속하는 광고 시간을 건너뛸니다.

용도

매니페스트 요청에서 가용 억제 파라미터를 쿼리 파라미터로 전달합니다.

```
GET /v1/master/{hashed-account-id}/{origin-id}/{asset}.m3u8?
aws.availSuppressionMode=BEHIND_LIVE_EDGE&aws.availSuppressionValue=00:00:10
```

이 파라미터는 다음 두 가지 컴패니언 파라미터와 함께 작동합니다.

- `aws.availSuppressionValue` - 기간(모드가가 아닌 경우 필수OFF)
- `aws.availSuppressionFillPolicy` - 부분 채우기 동작을 제어합니다(AFTER_LIVE_EDGE모드에만 적용됨).

요구 사항

가용성 억제 파라미터에는 다음 요구 사항이 적용됩니다.

- HLS 및 DASH 라이브 세션에 적용됩니다.
- `aws.availSuppressionValue` 모드가 `BEHIND_LIVE_EDGE` 또는 `인` 경우 `HH:MM:SS` 형식으로 제공해야 합니다 `AFTER_LIVE_EDGE`.
- `aws.availSuppressionFillPolicy`는 모드가 `인` 경우에만 유효합니다 `AFTER_LIVE_EDGE`.

모드 동작

다음 표에서는 각 억제 모드의 효과를 설명합니다.

Mode	Effect
OFF (또는 생략됨)	가용성 억제 없음 - 모든 광고 중단이 정상적으로 채워집니다.
BEHIND_LIVE_EDGE	라이브 엣지 뒤의 지정된 기간 내에 시작되는 광고 시간 억제
AFTER_LIVE_EDGE	라이브 엣지에서 지정된 기간 이후에 시작되는 광고 시간 억제

채우기 정책(AFTER_LIVE_EDGE만 해당)

다음 표에서는 모드가 일 때의 채우기 정책 동작을 설명합니다. `AFTER_LIVE_EDGE`

채우기 정책	Effect
FULL_AVAIL_ONLY (기본값)	억제 기간 외부에 있는 가용 영역만 채웁니다.
PARTIAL_AVAIL	억제 기간을 넘어 확장되는 가용 영역 부분을 채웁니다.

Example가용성 억제 예제

다음 요청은 라이브 엣지 뒤 10초 이내에 광고 시간을 억제합니다.

```
GET /v1/master/{hashed-account-id}/{origin-id}/{asset}.m3u8?
aws.availSuppressionMode=BEHIND_LIVE_EDGE&aws.availSuppressionValue=00:00:10
```

제한 사항

다음과 같은 제한이 적용됩니다.

- 세션 초기화 후에는이 파라미터를 변경할 수 없습니다.
- aws.availSuppressionValue 모드가 인 경우를 제공해서는 안 됩니다OFF.
- 의 시간 형식이 잘못aws.availSuppressionValue되면 모드가 로 돌아갑니다OFF.

MediaTailor 구성 별칭 개요

AWS Elemental MediaTailor 구성 별칭을 사용하면 URL 도메인 및 기타 지원되는 필드에서 동적 변수 교체가 가능합니다. 이 기능을 사용하여 여러 도메인을 사용하고 세션 초기화 중에 URLs 동적으로 구성합니다.

사용 사례

구성 별칭을 사용하면 다음 시나리오에서 정교한 다중 구성 아키텍처를 사용할 수 있습니다.

- 지리적 라우팅: 리전별 별칭을 사용하여 최종 사용자 위치에 따라 요청을 다른 오리진 또는 광고 서버로 라우팅합니다. 구현 지침은 [CloudFront 오리진 장애 조치를 참조하세요](#).
- 콘텐츠 기반 라우팅: 다양한 콘텐츠 유형을 특수 오리진 또는 처리 파이프라인으로 전달합니다. 라우팅 동작 구성은 섹션을 참조하세요[the section called “CDN 라우팅 동작 설정”](#).
- 장애 조치 시나리오: 별칭 전환을 사용하여 자동 장애 조치를 통해 백업 오리진 및 광고 서버를 구현합니다. 자세한 구현은 [MQAR을 사용하여 MediaTailor에 대한 다중 리전 복원력 구현](#) 및 섹션을 참조하세요[the section called “CDN 통합 계획”](#).
- A/B 테스트: 플레이어 파라미터를 기반으로 트래픽을 라우팅하여 다양한 광고 서버, 오리진 또는 구성을 테스트합니다. 로드 테스트 지침은 [실제 사용자 모니터링을 사용하여 Amazon CloudFront에 대한 성능 테스트 준비 및 실행](#)을 참조하세요.
- 디바이스별 최적화: 다양한 디바이스 유형 또는 기능에 대한 콘텐츠 전송 및 광고 제공을 최적화합니다. 포괄적인 지침은 섹션을 참조하세요[MediaTailor, MediaPackage 및 CDN을 사용하여 매니페스트 필터링 설정](#).
- 로드 밸런싱: 동적 라우팅을 사용하여 여러 오리진 또는 광고 서버에 로드를 분산합니다. 구현 지침은 [MQAR을 사용하여 MediaTailor에 대한 다중 리전 복원력 구현](#) 및 섹션을 참조하세요[the section called “CDN 통합 계획”](#).

지원되는 필드

다음 구성 필드에서 동적 변수를 사용할 수 있습니다.

- VideoContentSourceUrl
- AdDecisionServerUrl
- LivePreroll.AdDecisionServerUrl
- AdSegmentUrlPrefix
- ContentSegmentUrlPrefix
- TranscodeProfileName
- SlateAdUrl
- StartUrl
- EndUrl

다음 섹션에서는 구성 별칭을 사용하는 방법을 설명합니다.

주제

- [MediaTailor를 사용하여 구성 별칭 생성 및 사용](#)
- [MediaTailor를 사용한 구성 별칭 사용 예제](#)

MediaTailor를 사용하여 구성 별칭 생성 및 사용

도메인 변수를 사용하기 전에 구성에 대한 구성 별칭을 생성합니다. 세션 초기화 시 구성 별칭을 도메인 대체 변수로 사용합니다.

제한 사항

구성 별칭을 사용할 때 다음 제한 사항에 유의하세요.

- 도메인에 사용되는 모든 동적 변수는 ConfigurationAliases 동적 변수로 정의되어야 합니다.
- 플레이어 파라미터 변수에는 접두사가 붙어야 합니다 player_params.. 예를 들어 player_params.origin_domain입니다.
- 별칭이 지정된 값 목록은 중요한 URLs(VideoContentSourceUrl, AdSegmentUrlPrefix, ContentSegmentUrlPrefix)의 도메인 변수에 대해 포괄적이어야 합니다.

- 동적 변수를 지정하지 않거나 잘못된 별칭을 사용하는 중요한 URLs의 도메인 변수에 대한 요청이 이루어진 경우 HTTP 400 상태 코드와 함께 요청이 실패합니다. 중요하지 않은 필드(SlateAdUrl, TranscodeProfileName, 범퍼 URLs)는 경고를 로깅하지만 요청에 실패하지는 않습니다.

누락된 별칭에 대한 대체 동작

구성 별칭을 찾을 수 없거나 유효하지 않은 경우 MediaTailor는 다음 대체 동작을 구현합니다.

- 도메인 변수: 도메인 변수 별칭이 누락되거나 유효하지 않은 경우 HTTP 400 상태 코드와 함께 요청이 실패합니다. 모든 도메인 변수에는 유효한 별칭이 정의되어 있어야 합니다.
- 비도메인 변수: URLs의 비도메인 부분에 사용되는 변수(예: 경로 요소 또는 쿼리 파라미터)의 경우 별칭이 누락되면 빈 문자열이 대체됩니다.
- 구성 검증: MediaTailor는 구성 생성 및 업데이트 작업 중에 필요한 모든 별칭이 있는지 확인합니다.

1단계: 구성 별칭 생성

MediaTailor 콘솔을 사용하여 도메인 교체에 사용할 구성 별칭을 생성하려면 다음 절차를 수행합니다.

Console

콘솔을 사용하여 구성 별칭을 생성하려면

1. <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.
2. 구성 페이지의 구성 별칭 섹션에서 플레이어 파라미터 추가를 선택합니다.
3. 플레이어 파라미터에 동적 변수로 사용할 플레이어 파라미터의 이름을 입력합니다. 예를 들어 `player_params.origin_domain`입니다.
4. 별칭에 플레이어 파라미터에 사용할 별칭과 해당 값을 입력합니다.
5. 확인을 선택합니다.

AWS Elemental MediaTailor 는 구성 별칭 섹션의 테이블에 새 파라미터를 표시합니다.

6. 이전 단계를 반복하여 플레이어 파라미터를 더 추가합니다.
7. 저장을 선택합니다.

API

API를 사용하여 구성 별칭을 생성하려면

MediaTailor 구성을 생성하거나 업데이트할 때 다음 JSON 구조와 함께 ConfigurationAliases 파라미터를 사용합니다.

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

2단계: 세션 초기화에서 구성 별칭 사용

구성 별칭을 설정한 후 세션 초기화 요청에서 도메인의 대체 변수로 사용할 수 있습니다. 이렇게 하면 세션의 도메인을 동적으로 구성할 수 있습니다.

Example기본 구성 별칭 예제

다음은 구성 별칭과 동적 도메인 변수를 포함하는 구성의 기본 예제입니다.

```
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"
    }
}
}

```

Example별칭을 사용한 세션 초기화

앞의 구성을 사용하면 플레이어 변수와 별칭을 사용한 세션 초기화 요청은 다음과 비슷합니다.

```

POST index.m3u8
{
  "playerParams": {
    "origin_domain": "pdx",
    "region": "pdx",
    "endpoint_id": "pdx",
    "ad_type": "customized"
  }
}

```

MediaTailor는 별칭 문자열을 구성 별칭 구성의 매핑된 값으로 바꿉니다.

ADS에 대한 요청은 다음과 같습니다.

```
https://abc.execute-api.us-west-2.amazonaws.com/ads?sid=[session.id]&ad_type=abc12345
```

매니페스트의 오리진에 대한 요청은 다음과 같습니다.

```
https://abc.mediapackage.us-west-2.amazonaws.com/out/v1/abcd
```

MediaTailor를 사용한 구성 별칭 사용 예제

다음 예제에서는 구성 별칭이 있는 전체 MediaTailor 구성, 별칭이 있는 세션 초기화 요청 및 별칭에 대한 처리 흐름을 보여줍니다.

Example별칭을 사용하여 구성 완료

다음 예제는 구성 별칭과 동적 도메인 변수를 포함하는 전체 구성을 보여줍니다.

```

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"
    }
}
}

```

Example별칭을 사용한 세션 초기화

다음 예제에서는 플레이어 변수와 별칭을 지정하는 세션 초기화 요청을 보여줍니다.

```

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"
  }
}

```

Example파라미터 처리 흐름

다음 예제에서 MediaTailor는 별칭 문자열을 구성 별칭의 매핑된 값으로 바꿉니다. 처리 결과 다음과 같은 요청이 발생합니다.

- ADS 요청:

```
https://abc.execute-api.us-west-2.amazonaws.com/ads?sid=[session.id]&ad_type=abc12345
```

- VideoContentSource 요청:

```
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]
```

파라미터를 ADS에 전달하는 MediaTailor

AWS Elemental MediaTailor 는 다음 단계를 사용하여 ADS에 대한 MediaTailor 요청에서 동적 변수 설정을 지원합니다.

- 쿼리 파라미터에 지원되는 형식 지정에 대한 자세한 내용은 섹션을 참조하세요 [MediaTailor 파라미터 참조 및 제한 사항](#).
- 구성 별칭 및 도메인 변수는 섹션을 참조하세요 [MediaTailor 구성 별칭 개요](#).

- ADS 요청에 대한 추가 사용자 지정은 섹션을 참조하세요 [고급 사용](#).

세션 초기화 방법

MediaTailor는 세션 초기화 및 파라미터 전달을 위한 여러 방법을 지원합니다.

1. 요청 본문이 있는 POST:

```
POST <master>.m3u8
{
  "adsParams": {"param1": "value1", "param2": "value2"},
  "playerParams": {"param3": "value3"}
}
```

2. URL의 쿼리 파라미터:

```
GET <master>.m3u8?ads.param1=value1&ads.param2=value2&playerParams.param3=value3
```

Important

초기화 시 파라미터를 한 번만 지정할 수 있습니다. 구성 별칭은 전달하기 전에 실제 값으로 확인됩니다.

세션 및 플레이어 정보를 ADS에 전달하려면

1. ADS와 협력하여 광고 쿼리에 응답해야 하는 정보를 결정합니다 AWS Elemental MediaTailor.
2. ADS 요구 사항을 충족하는 템플릿 ADS 요청 URL을 사용하는 구성을 MediaTailor에서 생성합니다. URL에는 정적 파라미터를 포함시키고 동적 파라미터의 자리 표시자를 포함시킵니다. 구성의 Ad decision server(광고 의사결정 서버) 필드에 템플릿 URL을 입력합니다.

다음 예제 템플릿 URL에서 correlation은 세션 데이터를 제공하고 deviceType은 플레이어 데이터를 제공합니다.

```
https://my.ads.server.com/path?
correlation=[session.id]&deviceType=[player_params.deviceType]
```

3. 플레이어에서는 AWS Elemental MediaTailor 이 플레이어 데이터의 파라미터를 제공하도록 세션 시작 요청을 구성합니다. 세션 시작 요청에 파라미터를 포함시키고 세션에 대한 후속적인 요청에서 이를 제외합니다.

세션을 초기화하기 위해 플레이어가 수행하는 호출 유형에 따라 플레이어(클라이언트) 또는 MediaTailor(서버)가 세션에 대한 광고 추적 보고를 제공하는지 여부가 결정됩니다. 이러한 두 가지 옵션에 대한 자세한 내용은 [광고 추적 데이터 보고](#) 단원을 참조하십시오.

서버 측 또는 클라이언트 측 광고 추적 보고 중 어떤 것을 원하는지 여부에 따라 다음 호출 유형 중 하나를 실행합니다. 두 가지 예제 호출의 경우 모두 userID는 ADS를 위한 것이며 auth_token은 오리진을 위한 것입니다.

- (선택 사항) 서버 측 광고 추적 보고를 위한 호출 - MediaTailor가 사용하여 ADS로 전송할 파라미터의 접두사를 지정합니다ads. MediaTailor가 오리진 서버로 전송할 파라미터의 접두사를 끈 상태로 둡니다.

다음 예제에서는에 대한 HLS 및 DASH에 대한 수신 요청을 보여줍니다 AWS Elemental MediaTailor. MediaTailor는 ADS에 대한 요청deviceType에서를 사용하고 오리진 서버에 대한 요청auth_token에서를 사용합니다.

HLS 예:

```
GET master.m3u8?ads.deviceType=ipad&auth_token=kjhdsaf7gh
```

DASH 예:

```
GET manifest.mpd?ads.deviceType=ipad&auth_token=kjhdsaf7gh
```

- (선택 사항) 클라이언트 측 광고 추적 보고를 위한 호출 - adsParams 객체 내의 ADS에 대한 파라미터를 제공합니다.

HLS 예:

```
POST master.m3u8
{
  "adsParams": {
    "deviceType": "ipad"
  }
}
```

DASH 예:

```
POST manifest.mpd
{
  "adsParams": {
    "deviceType": "ipad"
  }
}
```

플레이어가 세션을 시작하면 템플릿 ADS 요청 URL의 변수를 세션 데이터 및 플레이어의 ads 파라미터로 바 AWS Elemental MediaTailor 끕니다. 나머지 파라미터는 플레이어에서 오리진 서버로 전달됩니다.

Example광고 변수를 사용한 MediaTailor 요청

다음 예제에서는 앞에 나온 플레이어의 세션 초기화 호출 예제와 일치하는 AWS Elemental MediaTailor 에서의 ADS 및 오리진 서버로의 호출을 보여줍니다.

- MediaTailor는 세션 데이터와 플레이어의 디바이스 유형을 사용하여 ADS를 호출합니다.

```
https://my.ads.server.com/path?correlation=896976764&deviceType=ipad
```

- MediaTailor는 플레이어의 권한 부여 토큰을 사용하여 오리진 서버를 호출합니다.
- HLS 예:

```
https://my.origin.server.com/master.m3u8?auth_token=kjhdsaf7gh
```

- DASH 예:

```
https://my.origin.server.com/manifest.mpd?auth_token=kjhdsaf7gh
```

고급 사용

플레이어 및 세션 데이터로 다양한 방법으로 ADS 요청을 사용자 지정할 수 있습니다. ADS 호스트 이름만 포함하면 됩니다.

다음 예시에서는 요청을 사용자 지정할 수 있는 방법의 일부를 보여줍니다.

- 플레이어 파라미터와 세션 파라미터를 연결하여 새 파라미터를 생성합니다. 예제:


```
https://my.ads.com?key1=[player_params.value1][session.id]
```

- 플레이어 파라미터를 경로 요소의 일부로 사용합니다. 예제:

```
https://my.ads.com/[player_params.path]?key=value
```

- 플레이어 파라미터를 사용하여 단지 값을 전달하기 보다는 경로 요소와 키 자체를 둘 다 전달합니다. 예제:

```
https://my.ads.com/[player_params.path]?[player_params.key1]=[player_params.value1]
```

ADS 및 오리진에 대한 MediaTailor 파라미터 라우팅

AWS Elemental MediaTailor 는 접두사와 용도에 따라 쿼리 파라미터를 다른 대상으로 라우팅합니다. 파라미터 라우팅을 이해하는 것은 MediaPackage를 사용한 시간 이동 보기와 같은 오리진별 기능을 구현하는 데 필수적입니다.

파라미터 라우팅 규칙

MediaTailor는 쿼리 파라미터에 다음과 같은 라우팅 규칙을 사용합니다.

- 오리진 파라미터(접두사 없음): 특정 접두사가 없는 파라미터는 오리진별 기능을 위해 오리진 서버로 전달됩니다.
- ADS 파라미터(**ads.** 접두사): 접두사가 인 파라미터ads.가 Ad Decision Server로 전송됩니다.
- 매니페스트 파라미터(**manifest.** 접두사): 접두사가 인 파라미터manifest.는 CDN 라우팅 및 권한 부여에 사용됩니다.

Example파라미터 라우팅 예제

다음 세션 초기화는 파라미터 라우팅을 보여줍니다.

```
POST /v1/session/123456789/originId/index.m3u8
{
  "adsParams": {
    "param1": "value1",
    "param2": "value2"
  },
  "manifestParams": {
```

```

    "auth_token": "abc123"
  }
}

```

이 예시는 다음과 같이 설정되어 있습니다.

- param1 및 param2는 ADS로 전송됩니다.
- auth_token는 CDN 라우팅 및 권한 부여에 사용됩니다.
- 접두사가 없는 파라미터는 오리진 서버로 전달됩니다.

오리진 서버 파라미터 동작

오리진 서버에 전달되는 파라미터는 시간 이동 보기, 콘텐츠 필터링 및 인증과 같은 오리진별 기능을 활성화합니다.

일반적인 오리진 파라미터 사용 사례

오리진 파라미터는 일반적으로 다음에 사용됩니다.

- 시간 이동 보기: MediaPackage 시간 이동 콘텐츠에 대한 start 및 end 파라미터
- 콘텐츠 인증: 오리진 서버에 필요한 인증 토큰
- 콘텐츠 필터링: 반환되는 콘텐츠 변형을 제어하는 파라미터
- 오리진별 기능: 오리진 서버가 콘텐츠 처리에 사용하는 모든 파라미터

Important

파라미터는 세션 초기화 시 처리되고 세션 전체에서 유지됩니다. 시간 전환 기간과 같은 파라미터를 수정하려면 업데이트된 값으로 새 세션을 생성해야 합니다.

MediaTailor 및 MediaPackage 시간 이동 보기 통합

AWS Elemental MediaTailor 는 시간 이동 보기 파라미터를 MediaPackage 오리진에 전달하여 스타트 오버 및 캐치업 보기 기능을 활성화할 수 있습니다. 이 통합을 통해 뷰어는 이전 시점에서 라이브 콘텐츠 시청을 시작할 수 있습니다.

MediaPackage 시간 이동 보기 파라미터

MediaPackage는 MediaTailor를 통해 전달할 수 있는 다음과 같은 시간 이동 보기 파라미터를 지원합니다.

- **start**: 시간 이동 매니페스트의 시작을 정의하는 Epoch 또는 ISO 8601 타임스탬프
- **end**: 시간 이동 매니페스트의 끝을 정의하는 Epoch 또는 ISO 8601 타임스탬프
- **time_delay**: 콘텐츠 가용성을 지정된 초만큼 지연
- **manifest_window_seconds**: 구성된 기간보다 짧은 매니페스트 요청

Example MediaPackage 시간 이동 파라미터를 사용한 MediaTailor 세션 초기화 MediaPackage

다음 예제에서는 시간 이동 보기 파라미터로 세션을 초기화하는 방법을 보여줍니다.

```
GET /v1/master/123456789/originId/index.m3u8?
start=2024-08-26T10:00:00Z&end=2024-08-26T11:00:00Z
```

또는 명시적 세션 초기화 사용:

```
POST /v1/session/123456789/originId/index.m3u8
{
  "adsParams": {
    "param1": "value1"
  }
}
```

추가 쿼리 파라미터 사용:

```
?start=2024-08-26T10:00:00Z&end=2024-08-26T11:00:00Z
```

세션 중 파라미터 동작

시간 이동 보기 파라미터에는 특정 동작 특성이 있습니다.

- **세션 초기화**: 세션이 생성될 때 파라미터가 처리됩니다.
- **파라미터 지속성**: 재생 내내 파라미터가 세션과 연결된 상태로 유지됩니다.
- **초기화 후 변경 불가**: 활성 세션 중에는 파라미터를 변경할 수 없습니다.
- **새 세션 필요**: 시간 전환 기간을 수정하려면 업데이트된 파라미터 값으로 새 세션을 생성합니다.

MediaPackage 스타트오버 기간 요구 사항

MediaPackage에서 작동하는 시간 이동 보기의 경우 다음을 확인하세요.

1. MediaPackage 엔드포인트에서 스타트오버 기간 구성(최대 24시간)
2. CDN이 필요한 쿼리 파라미터를 MediaPackage에 전달하는지 확인합니다.
3. 플레이어 세션에서 일관된 재생 기간을 사용하여 CDN 캐싱 개선
4. 시작 및 종료 시간이 구성된 시작 기간 내에 속하는지 확인

Important

시간 이동 보기를 사용하는 경우 각 뷰어에 대해 고유한 시작 또는 종료 시간을 생성하는 대신 플레이어 세션에서 일관된 재생 기간을 사용합니다. 이렇게 하면 CDN에서 캐싱이 향상되고 잠재적인 제한이 방지됩니다.

MediaPackage 시간 이동 보기 구성 및 파라미터에 대한 자세한 내용은 AWS Elemental MediaPackage 사용 설명서의 [를 사용한 시간 이동 보기를 AWS Elemental MediaPackage](#) 참조하세요.

MediaTailor 파라미터 세션 동작 및 지속성

AWS Elemental MediaTailor 는 세션 초기화 시 파라미터를 처리하고 세션 수명 주기 동안 파라미터를 유지합니다. 동적 파라미터 시나리오를 구현하려면 세션 동작을 이해하는 것이 중요합니다.

세션 초기화 방법

MediaTailor는 파라미터를 사용한 세션 초기화를 위한 여러 방법을 지원합니다.

1. 암시적 세션 초기화: 초기 매니페스트 요청에 포함된 파라미터

```
GET /v1/master/123456789/originId/index.m3u8?
manifest.auth_token=abc123&start=2024-08-26T10:00:00Z
```

2. 명시적 세션 초기화(POST): 요청 본문에 제공된 파라미터

```
POST /v1/session/123456789/originId/index.m3u8
{
  "adsParams": {"param1": "value1"},
  "manifestParams": {"auth_token": "abc123"}
```

}

3. 명시적 세션 초기화(GET): 쿼리 파라미터로 제공되는 파라미터

```
GET /v1/session/123456789/originId/index.m3u8?
ads.param1=value1&manifestParams.auth_token=abc123
```

파라미터 지속성 및 불변성

MediaTailor 파라미터 동작은 다음 규칙을 따릅니다.

- 일회성 사양: 파라미터는 세션 초기화 시 한 번만 지정할 수 있습니다.
- 세션 전체 지속성: 파라미터는 전체 세션에서 유지 관리됩니다.
- 초기화 후 변경할 수 없음: 세션이 생성된 후에는 파라미터를 수정할 수 없습니다.
- 구성 별칭 확인: 별칭은 대상으로 전달하기 전에 실제 값으로 확인됩니다.

파라미터 수정 시나리오

재생 중에 파라미터를 수정하려면:

- 새 세션 생성: 업데이트된 파라미터 값으로 새 세션 초기화
- 플레이어 전환: 플레이어를 새 세션으로 원활하게 전환
- 파라미터 상속: 변경되지 않은 파라미터를 전달하여 일관성을 유지합니다.

Example시간 전환 파라미터 수정

1시간 기간에서 2시간 기간으로 변경하려면:

1. 현재 세션: start=2024-08-26T10:00:00Z&end=2024-08-26T11:00:00Z
2. 새 세션 생성: start=2024-08-26T10:00:00Z&end=2024-08-26T12:00:00Z
3. 플레이어를 새 세션 URL로 전환

Important

단일 세션에 대한 다중 다변량 재생 목록 요청은 첫 번째 요청 후 파라미터를 업데이트하지 않습니다. 파라미터는 세션 기간 동안 변경할 수 없습니다.

MediaTailor 파라미터 참조 및 제한 사항

동적 광고 변수를 구성하기 전에 모든 MediaTailor 구성에 적용되는 파라미터 형식 지정 요구 사항 및 제한 사항을 이해합니다.

AWS Elemental MediaTailor 는 매니페스트 쿼리 파라미터와 ADS 파라미터 모두에 대해 파라미터 문자 제한, 길이 제한 및 지원되는 형식에 대한 포괄적인 정보를 제공합니다.

매니페스트 쿼리 파라미터 문자 제한

매니페스트 쿼리 파라미터는 특정 문자를 지원하며 길이 제한이 정의되어 있습니다.

지원되는 문자(URL 인코딩 제외)

매니페스트 쿼리 파라미터에 다음 문자를 직접 사용할 수 있습니다.

- 영숫자 문자(A~Z, a~z, 0~9)
- 마침표(.)
- 하이픈(-)
- 밑줄(_)
- 백슬래시(\)

URL 인코딩이 지원되는 문자

URL 인코딩 시 다음과 같은 특수 문자가 지원됩니다.

- 마침표(.) = %2E
- 대시(-) = %2D
- 밑줄(_) = %5F
- 백분율(%) = %25
- 물결표(~) = %7E
- 슬래시(/) = %2F
- 별표(*) = %2A
- 같음(=) = %3D
- 질문(?) = %3F

URL 인코딩 지원

MediaTailor는 URL 인코딩에 사용할 때 백분율(%) 기호를 지원합니다(예: hello%20world = hello world). HTTP 사양에 따라 유효한 URL 인코딩인 경우 URL 인코딩된 문자를 사용할 수 있습니다.

지원되지 않는 문자

URL 인코딩 없이는 매니페스트 쿼리 파라미터에 :, , ?, &, =%, / (슬래시) 문자를 사용할 수 없습니다.

Important

MediaTailor는 %%% 또는 ==와 같은 이중 문자를 지원하지 않습니다. 문자 제한으로 인해 전체 URLs 매니페스트 쿼리 파라미터 값으로 사용할 수 없습니다.

길이 제한

모든 매니페스트 쿼리 파라미터(키와 값을 합친 값)의 총 길이는 2,000자를 초과할 수 없습니다.

ADS 파라미터 길이 제한

ADS에 대한 요청에 사용되는 파라미터에는 다음과 같은 길이 제한이 적용됩니다.

- ADS 파라미터 이름: 최대 10,000자
- ADS 파라미터 값: 최대 25,000자
- ADS URL: 최대 25,000자

MediaTailor 파라미터 문제 해결 가이드

AWS Elemental MediaTailor 는 문자 제한, URL 인코딩 문제 및 구성 별칭 오류를 포함하여 MediaTailor의 일반적인 파라미터 관련 문제를 해결하기 위한 지침을 제공합니다.

문자 제한 오류

지원되지 않는 문자가 포함된 파라미터 값은 오류 또는 예기치 않은 동작을 일으킬 수 있습니다.

일반적인 증상

다음 증상은 문자 제한 문제를 나타낼 수 있습니다.

- 매니페스트 URLs에 나타나지 않는 파라미터
- 세션 초기화 중 HTTP 400 오류

- 잘리거나 손상된 파라미터 값
- 잘못된 URLs

해결 단계

문자 제한 오류를 해결하려면:

1. 지원되지 않는 문자에 대한 파라미터 값 검토: :, ?, &, =, % /
2. 특수 문자에 적절한 URL 인코딩 적용(참조 [MediaTailor 파라미터 참조 및 제한 사항](#))
3. %% 또는와 같은 이중 문자를 사용하지 마세요. ==
4. 전체 URLs 사용할 수 없는 경우 대체 파라미터 형식을 고려하세요.

Example URL 인코딩 예제

를 사용하는 대신:

```
manifest.redirect_url=https://example.com/path?param=value
```

URL 인코딩 형식 사용:

```
manifest.redirect_url=https%3A%2F%2Fexample.com%2Fpath%3Fparam%3Dvalue
```

길이 제한 오류

길이 제한을 초과하는 파라미터는 잘리거나 오류가 발생할 수 있습니다.

길이 제한

다음 길이 제한이 적용됩니다([MediaTailor 파라미터 참조 및 제한 사항](#) 자세한 내용은 참조).

- 매니페스트 쿼리 파라미터(총): 2,000자
- ADS 파라미터 이름: 10,000자
- ADS 파라미터 값: 25,000자
- ADS URLs: 25,000자

해결 전략

길이 제한을 처리하려면:

1. 가능한 경우 더 짧은 파라미터 이름과 값을 사용합니다.
2. 큰 파라미터 값을 여러 개의 작은 파라미터로 분할
3. 구성 별칭을 사용하여 짧은 별칭을 더 긴 값에 매핑(참조 [MediaTailor 구성 별칭 개요](#))
4. 파라미터 참조와 함께 대용량 데이터에 외부 스토리지 사용 고려

구성 별칭 오류

구성 별칭 문제로 인해 HTTP 400 오류 또는 예기치 않은 파라미터 값이 발생할 수 있습니다.

일반적인 구성 별칭 오류

구성 별칭에서 일반적으로 발생하는 오류는 다음과 같습니다.

- HTTP 400 오류: 누락되거나 잘못된 별칭 값
- 도메인 변수가 올바르게 해결되지 않음
- 플레이어 파라미터가 별칭 값으로 대체되지 않음

해결 체크리스트

구성 별칭 오류를 해결하려면:

1. 모든 도메인 변수가 로 정의되었는지 확인 ConfigurationAliases
2. 플레이어 파라미터 변수가 player_params. 접두사를 사용하는지 확인
3. 중요한 URLs(VideoContentSourceUrl, , ContentSegmentUrlPrefix)의 도메인 변수에 대해 AdSegmentUrlPrefix별칭이 지정된 값 목록이 완전한지 확인합니다.
4. 세션 초기화 요청이 유효한 별칭 값을 지정하는지 확인
5. ConfigurationAliases 파라미터의 JSON 구조 검증

자세한 문제 해결 지침은 섹션을 참조하세요 [MediaTailor 구성 별칭 문제 해결 가이드](#).

Example구성 별칭 검증

구성에 필요한 모든 별칭이 포함되어 있는지 확인합니다.

```
"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
    }
  }
}

```

파라미터 처리 흐름 문제

파라미터 처리 흐름을 이해하면 파라미터 전달 및 변환 문제를 해결하는 데 도움이 됩니다.

파라미터 처리 순서

MediaTailor는 다음 순서로 파라미터를 처리합니다.

1. 세션 초기화 파라미터 검증
2. 구성 별칭 확인(해당하는 경우)
3. 파라미터 필터링(ADS vs 오리진 vs 매니페스트)
4. URL 인코딩 및 형식 지정
5. URLs에 대한 파라미터 애플리케이션

파라미터 흐름 디버깅

파라미터 처리 문제를 디버깅하려면:

1. 세션 초기화 시 파라미터가 올바르게 지정되었는지 확인
2. 구성 별칭이 예상 값으로 확인되는지 확인
3. 파라미터가 올바른 URLs에 나타나는지 확인
4. URL 인코딩이 올바르게 적용되었는지 확인

Example파라미터 흐름 예제

세션 초기화:

```

POST master.m3u8
{
  "playerParams": {"origin_domain": "pdx"},
  "manifestParams": {"test": "123"}
}

```

}

별칭 확인 및 처리 후:

- 오리진 요청: `https://abc.mediapackage.us-west-2.amazonaws.com/out/v1/abcd`
- 매니페스트 URL: `/v1/master/.../index.m3u8?aws.sessionId=session&test=123`

보안 고려 사항 및 모범 사례

MediaTailor는 파라미터 처리를 위한 보안 조치를 구현하여 일반적인 보안 문제를 방지합니다.

보안 조치

MediaTailor는 다음과 같은 보안 조치를 구현합니다.

1. 데이터베이스 팅창을 방지하기 위한 입력 크기 제한
2. 사용자 입력의 적절한 인코딩 및 삭제
3. 응답 손상을 방지하기 위한 입력의 URL 인코딩

모범 사례

안전한 파라미터 처리를 위해 다음 모범 사례를 따르세요.

- 전송하기 전에 클라이언트 측의 파라미터 값 검증
- 구성 별칭을 사용하여 가능한 파라미터 값 제한
- 파라미터에 민감한 정보를 포함하지 마세요.
- 파라미터 사용량에서 비정상적인 패턴 모니터링
- 파라미터 값을 권장 길이 제한 이내로 유지

MediaTailor 구성 별칭 문제 해결 가이드

AWS Elemental MediaTailor 는 일반적인 구성 별칭 문제 및 오류 시나리오에 대한 체계적인 문제 해결 지침을 제공합니다.

구성 별칭 검증 오류

구성 별칭이 누락되거나 유효하지 않은 경우 MediaTailor는 문제를 식별하는 데 도움이 되도록 특정 오류 응답을 반환합니다.

일반적인 오류 시나리오

다음 표에서는 일반적인 구성 별칭 오류와 해결 단계를 설명합니다.

오류	원인	해결 방법
HTTP 400: 잘못된 플레이어 파라미터 별칭	ConfigurationAliases에서 플레이어 파라미터 값을 찾을 수 없음	플레이어 파라미터 값이 해당 ConfigurationAliases 매핑에 키로 존재하는지 확인
HTTP 400: 필수 구성 별칭 누락	해당 ConfigurationAliases 항목 없이 사용되는 도메인 변수	모든 필수 별칭 매핑을 사용하여 누락된 플레이어 파라미터를 ConfigurationAliases에 추가
HTTP 400: 구성 검증 실패	ConfigurationAliases 구조의 형식이 잘못되었거나 불완전함	JSON 구조를 검증하고 모든 도메인 변수에 해당 별칭이 있는지 확인합니다.
URLs의 빈 문자열 대체	도메인이 아닌 변수 별칭을 찾을 수 없음	누락된 별칭 매핑 추가 또는 ConfigurationAliases에 기본값 제공

검증 체크리스트

다음 체크리스트를 사용하여 구성 별칭 설정을 검증합니다.

1. 도메인 변수 적용 범위: URLs 있는지 확인합니다. ConfigurationAliases
2. 별칭 완전성: 가능한 모든 플레이어 파라미터 값이 별칭 매핑에 키로 포함되어 있는지 확인합니다.
3. JSON 구조: ConfigurationAliases JSON의 형식이 올바르게 지정되고 중첩되었는지 확인합니다.
4. 파라미터 이름 지정: 모든 플레이어 파라미터가 player_params. 접두사를 사용하는지 확인
5. 값 일관성: 별칭 값이 용도에 맞는지 확인(URLs, 프로필 이름 등)

구성 별칭 확인 디버깅

이 체계적인 접근 방식에 따라 구성 별칭 해결 문제를 디버깅합니다.

Step-by-step 디버깅 방법론

다음 단계에 따라 구성 별칭 문제를 식별하고 해결합니다.

구성 별칭 디버깅 절차

1. 구성 구조 확인: 재생 구성에 올바른 형식의 ConfigurationAliases가 포함되어 있는지 확인합니다.

```
{
  "ConfigurationAliases": {
    "player_params.example_param": {
      "alias1": "value1",
      "alias2": "value2"
    }
  }
}
```

2. 플레이어 파라미터 형식 확인: 세션 초기화에 올바른 형식의 플레이어 파라미터가 포함되어 있는지 확인합니다.

```
{
  "playerParams": {
    "example_param": "alias1"
  }
}
```

3. 별칭 매핑 검증: 플레이어 파라미터 값("alias1")이 ConfigurationAliases 매핑에 키로 존재하는지 확인합니다.
4. 간단한 구성으로 테스트: 최소한의 구성으로 시작하여 문제를 격리합니다.
5. 오류 응답 모니터링: 특정 검증 메시지에 대한 MediaTailor 오류 응답 확인
6. 해결된 URLs 확인: 최종 해결된 URLs 유효하고 액세스 가능한지 확인

구성 별칭 모범 사례

신뢰할 수 있는 구성 별칭 구현을 보장하려면 다음 모범 사례를 따르세요.

보안 고려 사항

구성 별칭을 사용할 때 다음 보안 조치를 구현합니다.

- 입력 검증: 별칭 확인에 사용하기 전에 모든 플레이어 파라미터 값을 검증합니다.
- 별칭 값 삭제: 별칭 값에 예상 문자 및 형식만 포함되는지 확인합니다.
- 도메인 제한: 도메인 별칭을 신뢰할 수 있고 제어된 도메인으로 제한

- 액세스 제어: 권한 있는 직원으로만 구성 수정 제한

성능 최적화

다음 권장 사항을 사용하여 구성 별칭 성능을 최적화합니다.

- 별칭 수 최소화: 처리 오버헤드를 줄이기 위해 필요한 별칭만 사용
- 효율적인 이름 지정: 별칭 및 파라미터에 명확하고 일관된 이름 지정 규칙 사용
- 기본값: 일반적인 사용 사례에 적합한 기본 별칭 제공
- 구성 캐싱: MediaTailor의 구성 캐싱을 활용하여 성능 향상

유지 관리 및 모니터링

다음 방법을 사용하여 안정적인 구성 별칭 작업을 유지합니다.

- 정기적인 검증: 모든 별칭 매핑이 최신이고 작동하는지 정기적으로 검증합니다.
- 오류 모니터링: 누락되거나 잘못된 별칭과 관련된 HTTP 400 오류 모니터링
- 설명서: 모든 별칭 매핑 및 용도에 대한 명확한 설명서 유지
- 테스트 절차: 모든 별칭 조합에 대한 포괄적인 테스트 구현

MediaTailor 매니페스트 쿼리 파라미터

AWS Elemental MediaTailor 는 CDN 라우팅 및 권한 부여를 위한 매니페스트 쿼리 파라미터와 오리진 별 기능에 사용할 수 있는 기타 쿼리 파라미터 등 다양한 목적으로 쿼리 파라미터를 처리합니다.

AWS Elemental MediaTailor 는 세션 초기화로부터 쿼리 파라미터를 보존하고 이를 개인화된 매니페스트 URLs 및 기타 자산에 추가합니다. MediaTailor와 클라이언트 플레이어 사이에 콘텐츠 전송 네트워크(CDN)가 있는 경우 이 기능을 사용합니다.

CDN에 다음에 대한 쿼리 파라미터가 필요한 경우 매니페스트 쿼리 파라미터를 사용합니다.

- 다른 MediaTailor 엔드포인트로 동적 라우팅
- 토큰 권한 부여

클라이언트 측 동작과 CDN 동작 비교

MediaTailor는 클라이언트 측 보고 엔드포인트에 대한 쿼리 파라미터를 추가하지만 CDN 세그먼트에는 추가하지 않습니다. 업데이트된 기능은 다양한 MediaTailor 자산에서 쿼리 파라미터에 대한 보다 포괄적인 지원을 제공하여 CDN 라우팅 및 권한 부여 사용 사례의 유연성을 향상시킵니다.

MediaTailor는 클라이언트 측 보고 엔드포인트에 대한 쿼리 파라미터를 추가하지만 CloudFront(또는 기타 CDN) 세그먼트에 대한 쿼리 파라미터는 추가하지 않습니다.

파라미터 보존을 사용하려면 [AWS Support](#)에 문의하여 해당 매니페스트 쿼리 파라미터 전달 활성화를 요청합니다.

동작은 HLS와 DASH, 명시적 세션 초기화와 암시적 세션 초기화에 따라 다릅니다. 다음 주제에서는 MediaTailor가 파라미터를 매니페스트에 전달하도록 세션 초기화 요청을 구성하는 방법을 설명합니다.

오리진에 대한 MediaTailor 쿼리 파라미터 처리

AWS Elemental MediaTailor 는 용도에 따라 쿼리 파라미터를 다르게 처리합니다. 매니페스트 쿼리 파라미터(접두사 manifest.)는 CDN 라우팅 및 권한 부여에 사용되는 반면, 다른 쿼리 파라미터는 오리진별 기능에 사용할 수 있습니다.

MediaPackage를 사용한 시간 이동 보기

MediaPackage를 사용한 시간 이동 보기 기능의 경우 요청에 start 및 end 파라미터를 포함할 수 있습니다. 이러한 파라미터는 스타트오버 및 캐치업 보기를 위한 특정 콘텐츠 기간을 정의합니다.

Example시간 이동 보기 요청

시간 이동 보기에 대한 매니페스트 요청에 시작 및 종료 파라미터를 포함합니다.

```
GET /v1/master/111122223333/originId/index.m3u8?  
start=2024-08-26T10:00:00Z&end=2024-08-26T11:00:00Z
```

세션 중 파라미터 동작

쿼리 파라미터는 세션 초기화 시 처리됩니다. 시간 이동 보기 또는 기타 오리진별 기능의 경우:

- 세션 초기화: 초기 매니페스트 요청에 필수 파라미터 포함
- 파라미터 지속성: 파라미터는 세션과 연결되며 재생 내내 유지됩니다.
- 파라미터 변경: 시간 전환 기간 또는 기타 파라미터를 수정하려면 업데이트된 값으로 새 세션을 생성합니다.

⚠ Important

쿼리 파라미터의 특정 처리는 오리진 구성과 콘텐츠 오리진이 지원하는 파라미터에 따라 달라 집니다. MediaPackage 통합의 경우 CDN이 설명된 대로 필요한 쿼리 파라미터를 전달하도록 구성되어 있는지 확인합니다 [필수 쿼리 파라미터 구성](#).

MediaTailor 파라미터 문자 제한 및 URL 인코딩

AWS Elemental MediaTailor 는 매니페스트 쿼리 파라미터에서 특정 문자를 지원합니다. 특수 문자에 URL 인코딩을 사용할 수 있습니다.

URL 인코딩이 지원되는 문자

URL 인코딩에서는 다음과 같은 특수 문자가 지원됩니다.

- 마침표(.) = %2E
- 대시(-) = %2D
- 밑줄(_) = %5F
- 백분율(%) = %25
- 물결표(~) = %7E
- 슬래시(/) = %2F
- 별표(*) = %2A
- 같음(=) = %3D
- 질문(?) = %3F

URL 인코딩 지원

MediaTailor는 URL 인코딩에 사용할 때 백분율(%) 기호를 지원합니다(예: hello%20world = hello world). HTTP 사양에 따라 유효한 URL 인코딩인 경우 URL 인코딩된 문자를 사용할 수 있습니다.

⚠ Important

MediaTailor는 %%% 또는 ==와 같은 이중 문자를 지원하지 않습니다.

보안 고려 사항

MediaTailor는 파라미터 처리를 위해 다음과 같은 보안 조치를 구현합니다.

1. 데이터베이스 팅창을 방지하기 위한 입력 크기 제한
2. 사용자 입력의 적절한 인코딩 및 삭제
3. 응답 손상을 방지하기 위한 입력의 URL 인코딩

주제

- [MediaTailor HLS 암시적 세션 초기화](#)
- [MediaTailor DASH 암시적 세션 초기화](#)
- [MediaTailor HLS 및 DASH 명시적 세션 초기화](#)
- [MediaTailor 프로토콜별 파라미터 동작](#)
- [MediaTailor CDN 통합 및 파라미터 라우팅](#)

MediaTailor HLS 암시적 세션 초기화

AWS Elemental MediaTailor 는 요청에 키가 인 쿼리 파라미터가 포함된 경우 MediaTailor 리소스 링크에 쿼리 파라미터를 포함합니다manifest.*. 다음 예제에서는이 요청 형식을 보여줍니다.

```
GET /v1/master/111122223333/originId/index.m3u8?manifest.test=123&other=456
```

링크에는 manifest. 접두사가 포함되지 않습니다.

HLS의 파라미터 애플리케이션

HLS 암시적 세션의 경우 MediaTailor는 매니페스트 계층 구조의 다음 위치에 파라미터를 적용합니다.

- 다변량 재생 목록 URLs
- 미디어 재생 목록 URLs
- 콘텐츠 세그먼트 URLs
- 광고 세그먼트 URLs
- HLS 초기화 URLs

Example다변량 재생 목록

다음 예제에서는 MediaTailor가 다변량 재생 목록의 URL에 쿼리 파라미터를 포함하는 방법을 보여줍니다.

```
#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
```

Example미디어 재생 목록

다음 예제에서는 MediaTailor가 콘텐츠 세그먼트의 URLs에 쿼리 파라미터를 포함하는 방법을 보여줍니다.

```
#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 암시적 세션 초기화

AWS Elemental MediaTailor 는 클라이언트에 대한 세션을 생성하고 클라이언트가 세션 없이 매니페스트 요청을 할 때 쿼리 파라미터로 리디렉션합니다. 다음 예제에서는이 요청 형식을 보여줍니다.

```
GET /v1/dash/111122223333/originId/index.mpd?manifest.test=123&other=456
```

MediaTailor는 클라이언트에 대한 세션을 생성하고 쿼리 파라미터로 리디렉션합니다.

```
/v1/dash/111122223333/originId/index.mpd?sessionId=session&test=123
```

DASH의 파라미터 애플리케이션

DASH 매니페스트 응답에는 콘텐츠 세그먼트, 광고 세그먼트 및 초기화 URLs 포함됩니다. MediaTailor는 다음에 파라미터를 적용합니다.

- DASH 매니페스트 위치 요소
- SegmentTemplate 초기화 속성
- SegmentTemplate 미디어 속성
- 콘텐츠 세그먼트 URLs
- 광고 세그먼트 URLs

클라이언트가 요청을 하면 MediaTailor는 다음 예와 유사한 DASH 매니페스트를 사용하여 응답합니다. 첫 번째 기간은 콘텐츠 기간이므로 MediaTailor는 매니페스트 쿼리 파라미터를 여기에 삽입하지 않습니다. 광고 기간인 두 번째 기간에 MediaTailor는 매니페스트 쿼리 파라미터를 SegmentTemplate 요소의 initialization 속성 및 media 속성에 삽입합니다. Location 요소에는 매니페스트 쿼리 파라미터도 있습니다.

```
<?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 및 DASH 명시적 세션 초기화

AWS Elemental MediaTailor 는 클라이언트가 명시적 세션 초기화 요청을 할 때 다변량 재생 목록에 쿼리 파라미터 `manifestParams`로 포함하고 응답에 URLs을 추적합니다.

세션 초기화 방법

명시적 세션 초기화의 경우 요청 본문에 POST를 사용하거나 쿼리 파라미터에 GET을 사용할 수 있습니다.

1. 요청 본문이 있는 POST:

```

POST /v1/session/111122223333/originId/index.m3u8
{
    "adsParams": {"param1": "value1", "param2": "value2", "param3": "value3"},
    "manifestParams": {"test": "123"}
}

```

2. 쿼리 파라미터를 사용한 GET:

```
GET /v1/session/111122223333/originId/index.m3u8?
ads.param1=value1&ads.param2=value2&manifestParams.test=123
```

Example세션 초기화 요청

```
POST /v1/session/111122223333/originId/index.m3u8
{
  "adsParams": {
    "param1": "value1",
    "param2": "value2",
    "param3": "value3"
  },
  "manifestParams": {
    "test": "123"
  },
  "reportingMode": "client"
}
```

Example매니페스트 및 추적 응답

```
{
  "manifestUrl": "/v1/master/111122223333/originId/index.m3u8?
aws.sessionId=session&test=123",
  "trackingUrl": "/v1/tracking/111122223333/originId/session?test=123"
}
```

세션에 대한 매니페스트 응답에는 이전에 설명한 암시적 세션 초기화 워크플로와 유사한 manifestParams MediaTailor URLs의가 있습니다. 주요 차이점은 명시적 세션 초기화를 위한 매니페스트 파라미터가 로 시작되지 않는다는 것입니다manifest..

매니페스트 쿼리 파라미터는 변경할 수 없으며 세션 초기화 시에만 설정할 수 있습니다. 클라이언트가 단일 세션에 대해 다중 다변량 재생 목록을 요청하는 경우 MediaTailor는 첫 번째 요청 후 매니페스트 쿼리 파라미터를 업데이트하지 않습니다.

파라미터 처리 흐름

초기화 시 파라미터를 한 번만 지정할 수 있습니다. 구성 별칭은 전달하기 전에 실제 값으로 확인됩니다. 예:는 ConfigurationAliases 구성을 `ad_type=abc12345` 기반으로 `player_params.ad_type=customized` 확인됩니다.

MediaTailor 프로토콜별 파라미터 동작

AWS Elemental MediaTailor 는 HLS 및 DASH 프로토콜에 대해 매니페스트 쿼리 파라미터를 다르게 처리합니다. 각 프로토콜 유형에는 특정 애플리케이션 위치와 처리 방법이 있습니다.

HLS와 DASH 파라미터 처리 비교

다음 표에서는 MediaTailor가 HLS 및 DASH 프로토콜에서 매니페스트 쿼리 파라미터를 처리하는 방법을 비교합니다.

속성	HLS 동작	DASH 동작
파라미터 애플리케이션	매니페스트 URLs 및 세그먼트 URLs에 직접 적용	위치 요소, SegmentTemplate 속성 및 세그먼트 URLs
매니페스트 계층 구조	다변량 재생 목록 → 미디어 재생 목록 → 세그먼트	MPD → 기간 → AdaptationSets → 표시
초기화 URLs	존재하는 경우 HLS init URLs에 적용됨	SegmentTemplate 초기화 속성에 적용됨
세션 처리	재생 목록 업데이트 전반에 보존된 파라미터	세션 연속성을 위해 MPD 위치 요소에 포함된 파라미터
광고 세그먼트 처리	미디어 재생 목록의 광고 세그먼트 URLs에 적용됨	광고 기간 SegmentTemplate 미디어 속성에 적용됨

파라미터 애플리케이션 위치

MediaTailor는 매니페스트 쿼리 파라미터를 다음 위치에 적용합니다.

HLS 파라미터 애플리케이션

HLS 스트림의 경우 MediaTailor는 매니페스트 쿼리 파라미터를 다음에 적용합니다.

- 다변량 재생 목록 URLs: 미디어 재생 목록 참조에 파라미터가 추가됩니다.

- 미디어 재생 목록 URLs: 파라미터는 미디어 재생 목록 내의 세그먼트 URLs에 포함됩니다.
- 콘텐츠 세그먼트 URLs: 모든 콘텐츠 세그먼트에는 매니페스트 쿼리 파라미터가 포함됩니다.
- 광고 세그먼트 URLs
- HLS 초기화 URLs: 스트림에 있는 경우 Init 세그먼트에 파라미터 포함
- 슬레이트 세그먼트 URLs: 슬레이트 콘텐츠에는 일관된 CDN 동작을 위한 파라미터가 포함됩니다.

Example HLS 파라미터 애플리케이션 예제

세션 초기화를 고려할 때:

```
GET /v1/master/123456789/originId/index.m3u8?
manifest.auth_token=abc123&manifest.region=us-west
```

다변량 재생 목록에는 미디어 재생 목록 참조의 파라미터가 포함됩니다.

```
#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
```

미디어 재생 목록에는 세그먼트 URLs의 파라미터가 포함됩니다.

```
#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 파라미터 애플리케이션

DASH 스트림의 경우 MediaTailor는 매니페스트 쿼리 파라미터를 다음에 적용합니다.

- MPD Location 요소: Location 요소에는 매니페스트 새로 고침 요청에 대한 파라미터가 포함됩니다.
- SegmentTemplate 초기화 속성: Init 세그먼트 URLs 파라미터 포함
- SegmentTemplate 미디어 속성: 미디어 세그먼트 URL 템플릿에는 파라미터가 포함됩니다.
- 콘텐츠 세그먼트 URLs: 템플릿에서 생성된 모든 콘텐츠 세그먼트에는 파라미터가 포함됩니다.

- 광고 세그먼트 URLs: 광고 기간 세그먼트에는 CDN 통합을 위한 파라미터가 포함됩니다.
- 서버 측 보고 리디렉션: 302개의 광고 세그먼트 리디렉션으로 파라미터 보존

Example DASH 파라미터 애플리케이션 예제

세션 초기화를 고려할 때:

```
GET /v1/dash/123456789/originId/index.mpd?
manifest.auth_token=abc123&manifest.region=us-west
```

DASH 매니페스트에는 여러 위치의 파라미터가 포함됩니다.

```
<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 통합 및 파라미터 라우팅

AWS Elemental MediaTailor 매니페스트 쿼리 파라미터를 사용하면 정교한 CDN 통합 시나리오가 가능합니다. 동적 라우팅, 권한 부여 및 로드 밸런싱에 사용할 수 있습니다.

CDN 라우팅 사용 사례

매니페스트 쿼리 파라미터의 이점을 활용하는 일반적인 CDN 통합 시나리오는 다음과 같습니다.

- 지리적 라우팅: 최종 사용자 위치에 따라 리전별 MediaTailor 엔드포인트로 요청 라우팅
- 토큰 기반 권한 부여: 보안 콘텐츠 액세스를 위해 CDN을 통해 MediaTailor에 권한 부여 토큰 전달
- 로드 밸런싱: CDN 라우팅 로직을 사용하여 여러 MediaTailor 엔드포인트에 트래픽 분산
- A/B 테스트: 테스트를 위해 서로 다른 사용자 세그먼트를 서로 다른 MediaTailor 구성으로 라우팅
- 디바이스별 최적화: 디바이스 유형 또는 기능을 기반으로 요청 라우팅

CDN 계층 간 파라미터 보존

MediaTailor는 매니페스트 쿼리 파라미터가 여러 CDN 계층 및 요청 유형에서 보존되도록 합니다.

1. 초기 요청: 파라미터가 세션 초기화 요청에서 추출됩니다.
2. 매니페스트 생성: 매니페스트의 모든 관련 URLs에 파라미터가 적용됩니다.
3. 세그먼트 요청: 일관된 CDN 동작을 위해 모든 세그먼트 URLs에 파라미터가 포함됩니다.
4. 광고 삽입: 광고 삽입 및 세그먼트 교체 중에 파라미터가 보존됩니다.

Example CDN 권한 부여 흐름

다음 예제에서는 매니페스트 쿼리 파라미터를 사용한 전체 CDN 권한 부여 흐름을 보여줍니다.

1. 권한 부여 토큰이 포함된 클라이언트 요청 매니페스트:

```
GET https://cdn.example.com/mediatailor/v1/master/123456789/originId/index.m3u8?
manifest.auth_token=jwt_token_here&manifest.user_id=12345
```

2. CDN은 파라미터를 사용하여 MediaTailor에 요청을 전달합니다.

```
GET https://mediatailor.amazonaws.com/v1/master/123456789/originId/index.m3u8?
manifest.auth_token=jwt_token_here&manifest.user_id=12345
```

3. MediaTailor는 모든 URLs에 적용된 파라미터를 사용하여 매니페스트를 생성합니다.

```
#EXTM3U
#EXT-X-STREAM-INF:BANDWIDTH=2665212
../.../manifest/123456789/originId/session/0.m3u8?
auth_token=jwt_token_here&user_id=12345
```

4. 후속 세그먼트 요청에는 CDN 권한 부여를 위한 파라미터가 포함됩니다.

```
GET https://cdn.example.com/mediatailor/segment/123456789/originId/session/0/1?
auth_token=jwt_token_here&user_id=12345
```

광고 추적 데이터 보고

MediaTailor는 시청자가 시청한 광고의 양을 추적하고 보고하는 두 가지 옵션을 제공합니다. 서버 측 광고 보고 접근 방식에서 MediaTailor는 광고를 추적하고 비컨(추적 신호)을 광고 서버로 직접 전송합

니다. 또는 클라이언트 측 추적 접근 방식에서 클라이언트 플레이어(사용자의 디바이스)는 광고를 추적하고 비컨을 광고 서버로 전송합니다. 재생 세션에 사용되는 광고 보고 유형은 플레이어가 MediaTailor에서 세션을 시작하기 위해 수행하는 특정 요청에 따라 달라집니다.

동적 변수를 사용하여 세션 및 플레이어 데이터를 광고 서버에 전달하는 방법에 대한 자세한 내용은 섹션을 참조하세요 [ADS 요청에 대한 MediaTailor 동적 광고 변수](#). 세션 초기화 파라미터에 대한 자세한 내용은 섹션을 참조하세요 [MediaTailor 매니페스트 쿼리 파라미터](#).

주제

- [MediaTailor 서버 측 광고 추적 및 보고](#)
- [클라이언트 측 광고 추적](#)

MediaTailor 서버 측 광고 추적 및 보고

AWS Elemental MediaTailor 는 포괄적인 광고 추적 및 측정을 위해 기본적으로 서버 측 보고를 사용합니다. 서버 측 보고인 경우, 플레이어가 매니페스트에 광고 URL을 요청할 때 서비스는 즉시 광고 추적 URL에 광고 조회량을 보고합니다. 플레이어가 MediaTailor로 재생 세션을 초기화한 후에는 서버 측 보고를 수행하기 위해 사용자 또는 플레이어의 추가 입력이 필요하지 않습니다. 각 광고가 재생되면 MediaTailor는 광고 서버에 비컨을 전송하여 광고가 얼마나 많이 보였는지 보고합니다. MediaTailor는 첫 번째 사분위수, 중간점, 세 번째 사분위수, 광고 완료 등 광고 시작 및 사분위수의 광고 진행에 대한 비컨을 전송합니다.

서버 측 광고 보고를 수행하려면

- 플레이어에서 프로토콜에 따라 다음 형식 중 하나의 요청을 사용하여 새 MediaTailor 재생 세션을 초기화합니다.

- 예: HLS 형식

```
GET <mediatailorURL>/v1/master/<hashed-account-id>/<origin-id>/<asset-id>?ads.<key-value-pairs-for-ads>&<key-value-pairs-for-origin-server>
```

- 예: DASH 형식

```
GET <mediatailorURL>/v1/dash/<hashed-account-id>/<origin-id>/<asset-id>?ads.<key-value-pairs-for-ads>&<key-value-pairs-for-origin-server>
```

키-값 페어는 광고 추적을 위한 동적 타겟팅 파라미터입니다. 요청에 파라미터를 추가하는 방법에 대한 자세한 내용은 [the section called “동적 광고 변수”](#) 단원을 참조하십시오.

AWS Elemental MediaTailor 는 매니페스트 URL을 사용하여 요청에 응답합니다. 매니페스트에는 미디어 매니페스트용 URL이 포함됩니다. 미디어 매니페스트에는 광고 세그먼트 요청 링크가 포함되어 있습니다.

Note

MediaTailor가 추적 URL에서 이중 슬래시(//)를 발견하면 슬래시를 1(/)로 축소합니다.

플레이어가 광고 세그먼트 URL(/v1/segment 경로)에 재생을 요청하면 AWS Elemental MediaTailor 은 광고 추적 URL을 통해 광고 서버에 적절한 비콘을 전송합니다. 동시에 서비스는 실제 *.ts 광고 세그먼트로 리디렉션을 발행합니다. 광고 세그먼트는 MediaTailor가 트랜스코딩된 광고를 저장하는 Amazon CloudFront 배포 또는 광고를 캐싱한 콘텐츠 전송 네트워크(CDN)에 있습니다.

다음 섹션에서는 MediaTailor의 서버 측 광고 추적 작업에 대한 자세한 정보를 제공합니다.

주제

- [서버 가이드 광고 삽입\(SGAI\)을 사용한 서버 측 추적](#)
- [서버 측 추적 비콘 용어집](#)
- [서버 측 추적 타이밍 및 캐싱 동작](#)
- [MediaTailor 서버 측 추적 기능](#)

서버 가이드 광고 삽입(SGAI)을 사용한 서버 측 추적

서버 가이드 광고 삽입(SGAI)을 사용하는 경우 서버 측 추적은 위에서 설명한 연결 모드 접근 방식과 다른 세션 없는 비콘 메커니즘을 사용합니다. MediaTailor는 광고 세그먼트를 콘텐츠 매니페스트(/v1/segment요청 추적)에 스티칭하는 대신 SGAI는 광고 URIs.

세션 없는 서버 측 비콘 작동 방식

다음 단계에서는 SGAI 세션에서 서버 측 비콘이 작동하는 방식을 설명합니다.

1. 세션 초기화: 플레이어가를 사용하여 HLS 다변량 재생 목록을 요청합니다
다aws.insertionMode=GUIDED. 서버 측 보고가 기본값입니다(aws.reportingMode파라미터 필요 없음). 연결 모드와 달리 세션 초기화 응답에는가 포함되지 않습니다trackingUrl.
2. 캐시 가능 매니페스트: MediaTailor는 MediaTailor 중간 자산 목록 엔드포인트를 가리키는 CLASS="com.apple.hls.interstitial" 및 X-ASSET-LIST 속성이 있는 EXT-X-DATERANGE 태그가 포함된 캐시 가능 매니페스트를 반환합니다.
3. 비컨 메타데이터가 포함된 자산 목록: 플레이어가 광고 시간을 발견하면 자산 목록을 가져옵니다. MediaTailor는 각 광고 URI에 암호화된 비컨 메타데이터가 포함된 JSON 응답을 반환합니다.

```
{
  "ASSETS": [
    {
      "DURATION": 30.0,
      "URI": "https://cdn.example.com/ad/master.m3u8?
awsBeaconData=<encrypted>&awsBeaconDomain=<MediaTailor-
endpoint>&awsConfigurationName=<config-name>"
    }
  ]
}
```

서버 측 보고가 활성 상태이면 응답에 TRACKING 섹션이 포함되지 않습니다. 광고 URIs는 모든 비컨 데이터를 전달합니다.

4. HLS 변수 대체: 플레이어가 광고 다변량 재생 목록을 가져옵니다. 광고 매니페스트는 #EXT-X-DEFINE:QUERYPARAM 지시문을 사용하여 HLS 변수 대체를 통해 URI 쿼리 문자열의 비컨 파라미터를 세그먼트 URLs로 전달합니다.

```
#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
```

플레이어는 광고 매니페스트 URI 쿼리 문자열의 값을 사용하여

{awsBeaconData}{awsBeaconDomain}, 및 {awsConfigurationName} 변수를 확인한 다음 MediaTailor를 통해 각 광고 세그먼트를 요청합니다.

5. 세그먼트 요청 시 비컨 실행: 플레이어가 각 광고 세그먼트를 요청하면 요청은 MediaTailor를 통해 라우팅됩니다. 서비스는 비컨 데이터를 복호화하고, 광고 내 세그먼트의 위치(억제, 1사분위수, 중간점, 3사분위수 또는 완료)를 결정하고, 광고 서버에 적절한 VAST 추적 비컨을 실행합니다. 그런 다음 MediaTailor는 플레이어를 실제 광고 콘텐츠 세그먼트로 리디렉션합니다.

SGAI 서버 측 비컨에 대한 플레이어 요구 사항

SGAI에서 서버 측 비컨을 사용하려면 플레이어가 다음 요구 사항을 충족해야 합니다.

- HLS 버전 11 이상
- HLS Interstitials EXT-X-DATERANGE CLASS 속성 지원
- #EXT-X-DEFINE:QUERYPARAM 변수 대체 지원(RFC 8216bis). 플레이어는 쿼리 파라미터 값을 세그먼트 URLs로 대체하기 전에 백분율 디코딩해야 합니다.

Note

SGAI 서버 측 비컨은 현재 HLS에서만 지원됩니다. DASH는 아직 SGA 서버 측 비컨에 대해 지원되지 않습니다.

스티칭 모드 서버 측 추적과의 비교

다음 표에는 서버 측 추적이 스티칭 광고 삽입과 서버 가이드 광고 삽입 간에 어떻게 다른지 요약되어 있습니다.

속성	Stitched(SSAI)	서버 가이드(SGA)
매니페스트 캐시 가능성	세션당, 캐시할 수 없음	캐시 가능, 최종 사용자 간에 공유
광고 세그먼트 라우팅	세션 ID를 /v1/segment/ 사용하여	암호화된 비컨 데이터 BLOB을 /v1/segment/ 사용하여
비컨의 세션 상태	MediaTailor에 세션당 저장됨	세션 없음 - 모든 상태가 암호화된 awsBeaconData 파라미터로 전달됩니다.

속성	Stitched(SSAI)	서버 가이드(SGAI)
세션 시작 시 URL 추적	세션 초기화 응답에 반환됨	제공되지 않음 - 비컨 데이터가 각 자산 목록 응답의 광고 URIs에 포함됩니다.
DASH 지원	지원됨	아직 지원되지 않음

Note

라이브 SGAI 세션의 경우를 사용하여 매니페스트 기반 광고 미리 가져오기를 활성화할 수 있습니다. `aws.guidedPrefetchMode=MANIFEST`. 이는 연결된 (SSAI) 세션과 함께 사용되는 일정 기반 미리 가져오기 API와는 별개입니다. 자세한 내용은 [매니페스트 하트비트를 사용한 안내 미리 가져오기](#)를 참조하세요.

서버 측 추적 비컨 용어집

MediaTailor 서버 측 추적은 표준화된 비컨 세트를 사용하여 광고 서버 및 확인 서비스에 광고 시청 진행 상황을 보고합니다. 이러한 비컨은 비디오 광고 측정에 대한 대화형 광고국(IAB) 표준에 부합하며 광고 노출 및 완료율을 정확하게 보고합니다.

서버 측 추적 비컨 유형

비컨 유형	실행 시	용도	타이밍 세부 정보
노출	플레이어가 첫 번째 광고 세그먼트를 요청하는 경우	광고 콘텐츠가 로드를 시작했으며 최종 사용자에게 표시되려고 함을 나타냅니다.	광고에 대한 첫 번째 /v1/segment 요청 시 실행됩니다. 노출을 계산하기 전에 광고 콘텐츠 로딩을 시작하도록 요구하는 IAB 지침에 부합합니다. 전체 시퀀스는 서버 측 추적 워크플로 단원을 참조하십시오.
시작	플레이어가 광고 콘텐츠를 렌더링하기 시작하는 경우	광고 재생이 실제로 시작되었는지 확인합니다.	일반적으로 첫 번째 세그먼트 요청의 노출 비컨과 동시에 실행되지만 광고 렌더링의 실제

비컨 유형	실행 시	용도	타이밍 세부 정보
			시작을 나타냅니다. 이러한 구분은 노출 이벤트와 시작 이벤트를 별도로 추적하는 확인 서비스에 중요합니다.
첫 번째 사분위수	플레이어가 광고 기간의 25%에 도달한 경우	광고의 1분기 동안 지속적인 광고 시청을 측정합니다.	플레이어가 광고 기간의 25% 지점이 포함된 세그먼트를 요청할 때 실행됩니다. 예를 들어 2초 세그먼트가 있는 20초 광고에서는 일반적으로 3번째 세그먼트에 대한 요청 시(광고 후 약 4~6초) 실행됩니다.
중간점	플레이어가 광고 기간의 50%에 도달하는 경우	광고의 절반을 통해 지속적인 광고 시청을 측정합니다.	플레이어가 광고 기간의 50% 지점이 포함된 세그먼트를 요청할 때 실행됩니다. 예를 들어 2초 세그먼트가 있는 20초 광고에서는 일반적으로 5번째 세그먼트에 대한 요청 시(광고 후 약 8~10초) 실행됩니다.
세 번째 사분위수	플레이어가 광고 기간의 75%에 도달하는 경우	광고의 4분의 3을 통해 지속적인 광고 시청을 측정합니다.	플레이어가 광고 기간의 75% 지점이 포함된 세그먼트를 요청할 때 실행됩니다. 예를 들어 2초 세그먼트가 있는 20초 광고에서는 일반적으로 8번째 세그먼트에 대한 요청 시(광고 후 약 14~16초) 실행됩니다.

비컨 유형	실행 시	용도	타이밍 세부 정보
완료	플레이어가 광고 끝에 도달하면	전체 광고가 최종 사용자에게 전달되었는지 확인합니다.	플레이어가 광고의 최종 세그먼트를 요청할 때 실행됩니다. 이는 최종 사용자가 전체 광고 콘텐츠를 볼 수 있음을 나타냅니다. 예를 들어 2초 세그먼트가 있는 20초 광고에서는 일반적으로 10번째 세그먼트에 대한 요청 시(광고 후 약 18~20초) 실행됩니다.

Note

비컨 실행의 정확한 타이밍은 세그먼트 지속 시간 및 광고 길이에 따라 달라집니다. MediaTailor는 특정 광고 기간 및 세그먼트 구조를 기반으로 각 사분위수 위치에 해당하는 적절한 세그먼트 요청을 계산합니다.

서버 측 추적 타이밍 및 캐싱 동작

서버 측 보고에서 MediaTailor는 매니페스트 구문 분석 또는 사전 로드 활동이 아닌 플레이어의 실제 세그먼트 요청을 기반으로 추적 이벤트를 실행합니다. 이 접근 방식은 비디오 광고 측정에 대한 업계 표준에 맞는 정확한 노출 계수를 보장합니다.

주요 타이밍 원칙

MediaTailor 서버 측 추적은 다음과 같은 기본 타이밍 원칙을 따릅니다.

- 실제 세그먼트 요청에서 이벤트 추적 실행 - 플레이어가 매니페스트 구문 분석 또는 캐싱이 아닌 /v1/segment URLs에 HTTP 요청을 하는 경우에만 비컨이 전송됩니다.
- 매니페스트의 플레이어 캐싱 및 사전 로드는 이벤트를 트리거하지 않습니다. 플레이어는 추적 이벤트를 생성하지 않고도 매니페스트 정보를 구문 분석, 캐시 또는 사전 로드할 수 있습니다.
- 세그먼트 미리 가져오기는 이벤트를 트리거합니다 - 플레이어가 재생 전에 실제 광고 세그먼트를 미리 가져오는 경우 세그먼트 요청이 유효한 노출을 구성하는 업계 표준 동작을 따릅니다.
- 각 /v1/세그먼트 요청은 적절한 비컨을 트리거합니다. 특정 추적 이벤트(억제, 사분위수, 완료)는 요청되는 광고 위치 및 세그먼트에 따라 결정됩니다.

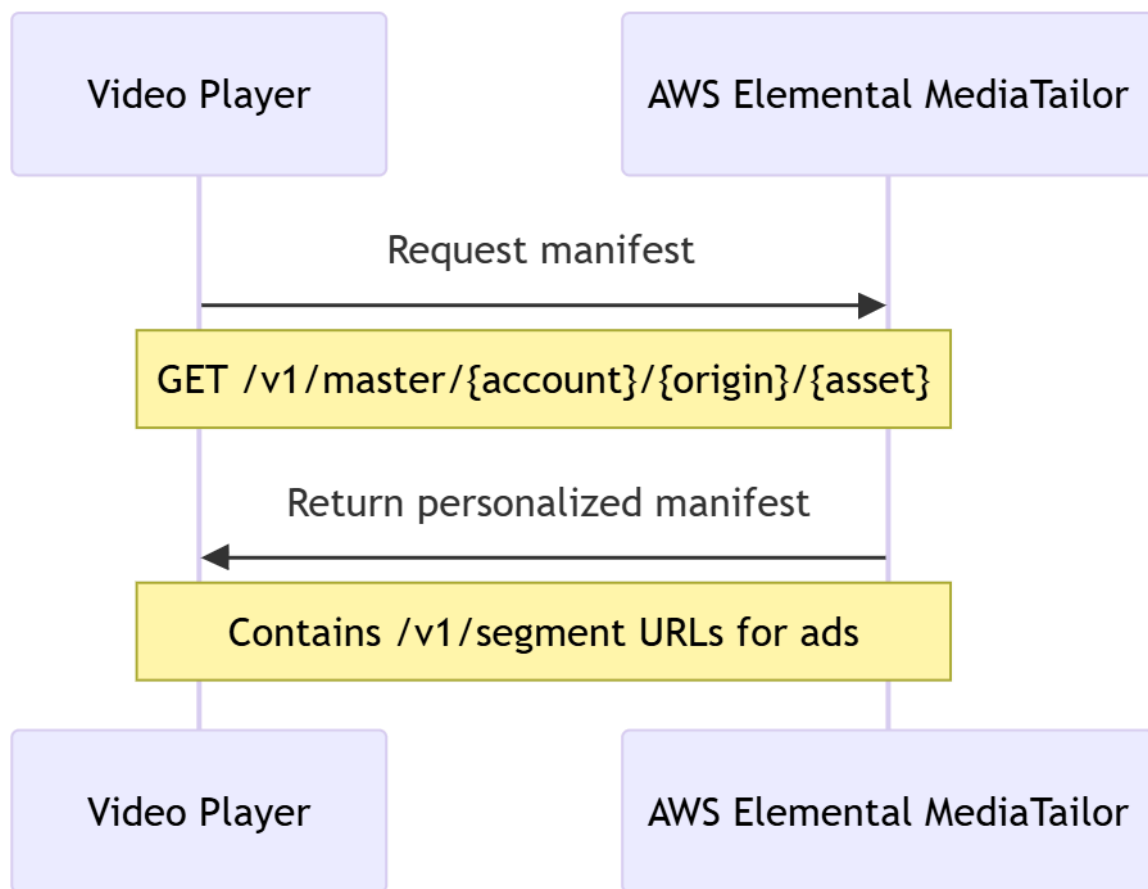
- 타이밍은 IAB 표준에 부합 -이 접근 방식은 비디오 광고 측정 및 노출 계산에 대한 대화형 광고국 지침을 따릅니다.

서버 측 추적 워크플로

다음 다이어그램은 플레이어 요청과 관련하여 이벤트 추적이 실행되는 시기를 보여주는 전체 서버 측 추적 워크플로를 보여줍니다.

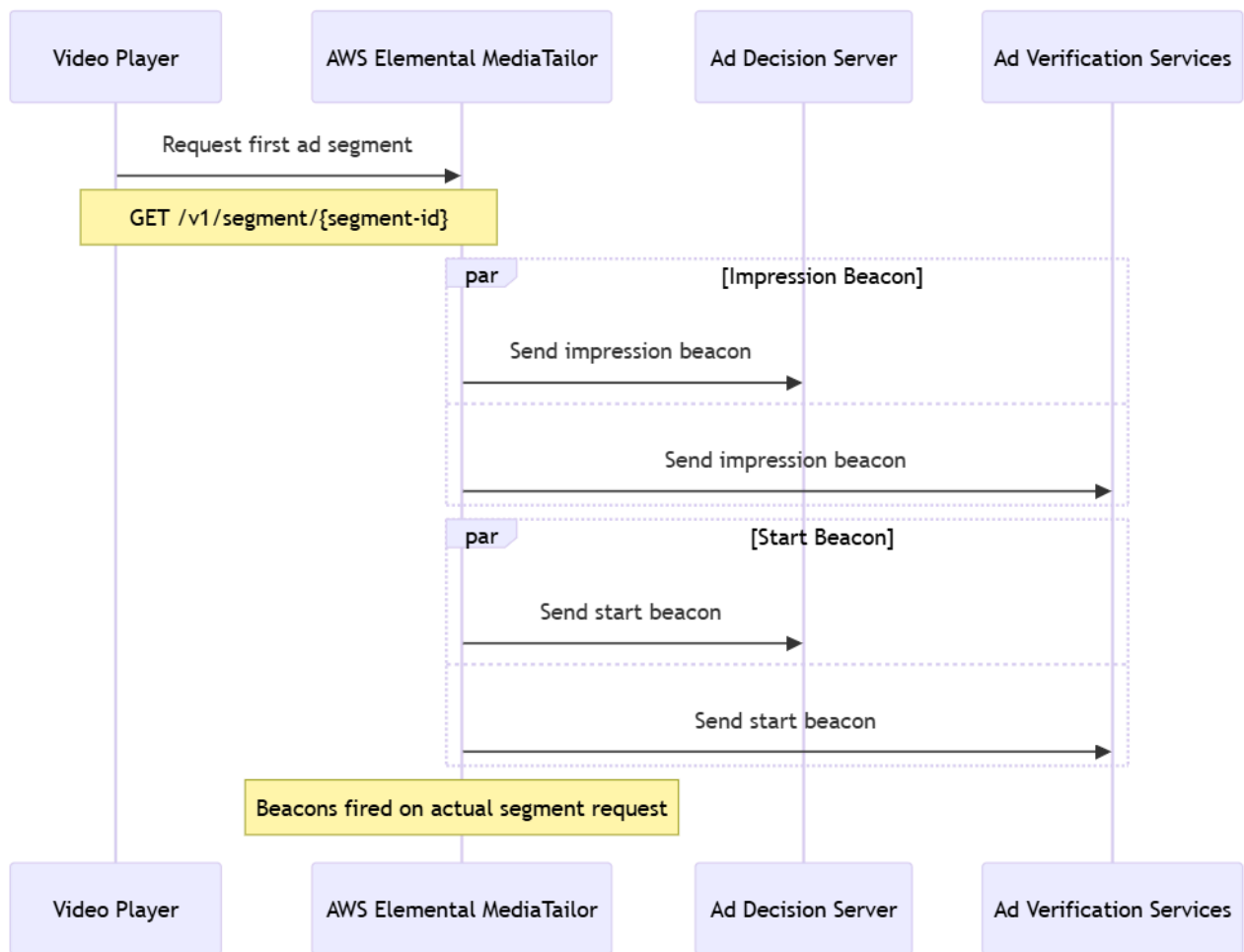
1단계: 세션 초기화

플레이어가 MediaTailor에서 매니페스트를 요청하면 광고 세그먼트 URLs.



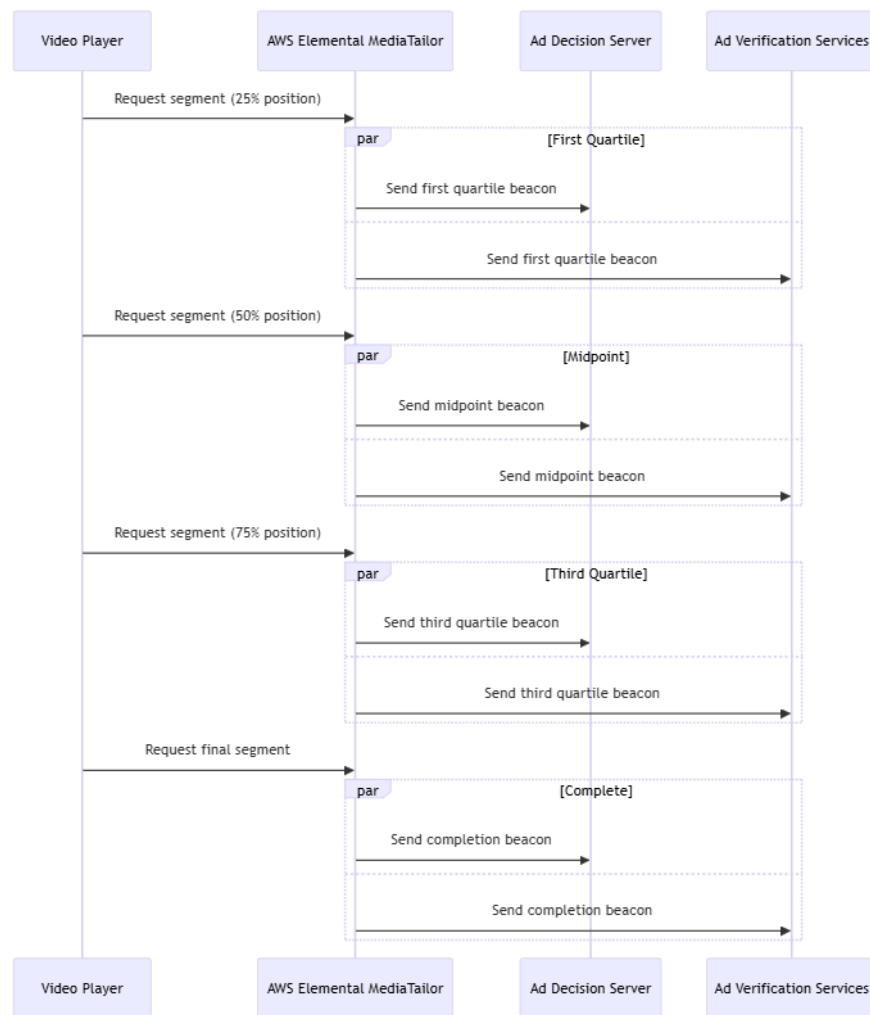
2단계: 광고 요청 및 노출 추적

플레이어가 첫 번째 광고 세그먼트를 요청하면 MediaTailor는 노출을 실행하고 광고 결정 서버와 광고 확인 서비스 모두에 비컨을 시작합니다.



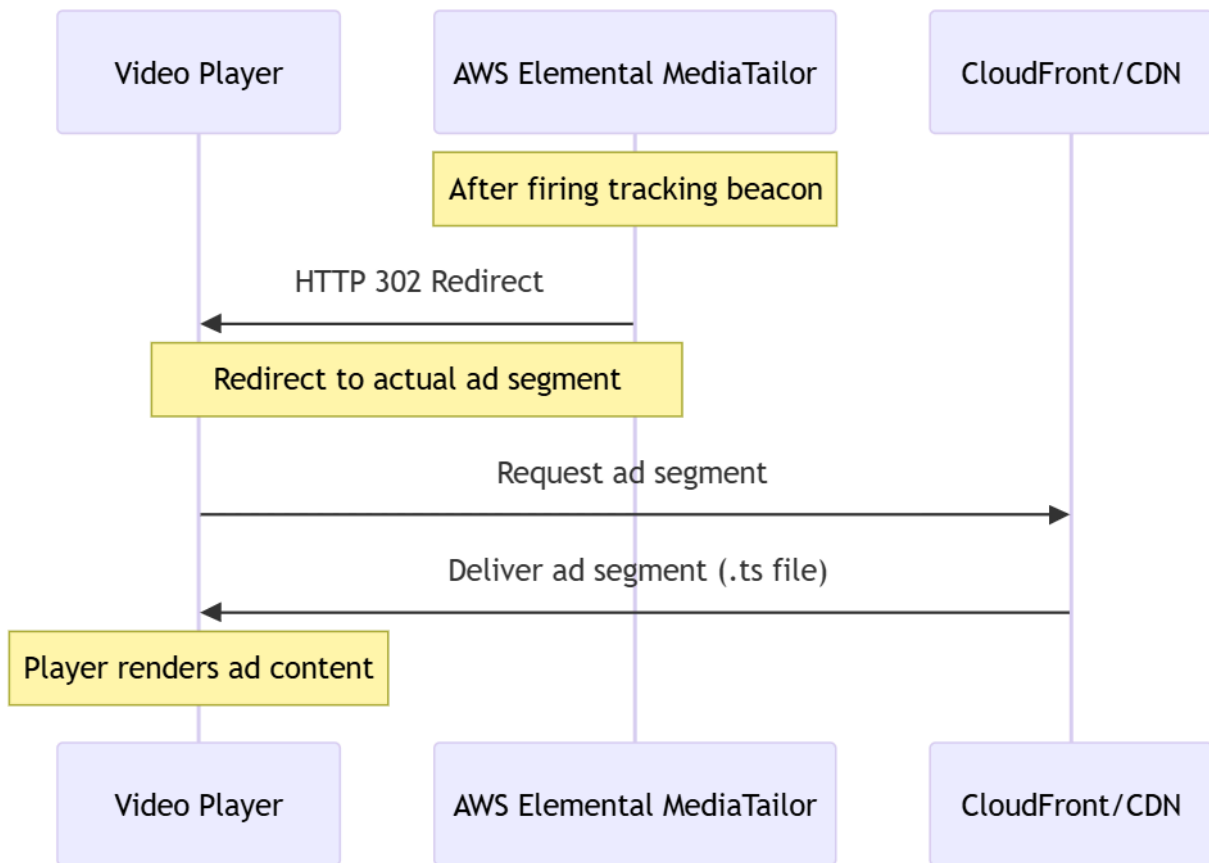
3단계: 사분위 추적

MediaTailor는 후속 세그먼트 요청에 따라 사분위 비컨(1사분위수, 중간점, 3사분위수, 완료)을 실행합니다.



4단계: 세그먼트 전송

추적 비컨을 실행한 후 MediaTailor는 Amazon CloudFront 또는 CDN에서 실제 광고 세그먼트로 리디렉션합니다.



서버 측 추적 워크플로에는 다음과 같은 주요 타이밍 동작이 포함됩니다.

1. 세션 초기화 - 플레이어가 MediaTailor에서 매니페스트를 요청합니다. MediaTailor는 /v1/segment 경로가 있는 광고 세그먼트 URLs이 포함된 개인화된 매니페스트를 반환합니다.
2. 매니페스트 구문 분석 및 캐싱 - 플레이어는 매니페스트를 구문 분석하고 세그먼트 정보를 미리 로드하거나 캐싱할 수 있습니다. 플레이어 캐싱 동작에 관계없이 이 단계에서는 추적 이벤트가 실행되지 않습니다.
3. 광고 세그먼트 요청 및 노출 추적 - 플레이어가 실제로 첫 번째 광고 세그먼트(일반적으로 재생용)를 요청하면 MediaTailor는 노출 비컨을 실행하고 광고 결정 서버와 광고 확인 서비스 모두에 대한 이벤트 추적을 시작합니다. 이는 매니페스트를 구문 분석할 때가 아니라 /v1/segment URL에 대한 실제 HTTP 요청에서 발생합니다.

4. 세그먼트 요청을 기반으로 한 사분위 추적 - MediaTailor는 광고 기간 내에 계산된 사분위수 위치에 해당하는 후속 세그먼트 요청을 기반으로 광고 결정 서버와 광고 확인 서비스 모두에 사분위수 비컨 (1사분위수, 중간점, 3사분위수, 완료)을 실행합니다.
5. 세그먼트 전송 - 적절한 추적 비컨을 실행한 후 MediaTailor는 실제 광고 세그먼트(Amazon CloudFront 또는 CDN)로 HTTP 리디렉션을 실행합니다.

플레이어 캐싱 및 사전 로드 고려 사항

MediaTailor 서버 측 추적은 정확한 노출 측정을 유지하면서 다양한 플레이어 캐싱 및 사전 로드 전략과 호환되도록 설계되었습니다.

- 매니페스트 사전 로드 - 매니페스트 정보를 사전 로드하거나 캐시하는 플레이어는 추적 이벤트를 트리거하지 않습니다. 추적 이벤트는 실제 세그먼트 요청이 이루어진 경우에만 실행됩니다.
- 세그먼트 미리 가져오기 - 플레이어가 재생 전에 광고 세그먼트를 미리 가져오는 경우 추적 이벤트는 해당 세그먼트가 요청될 때 실제 재생 시간보다 빠를 수 있습니다. 이 동작은 세그먼트 요청을 유효한 노출로 간주하는 업계 표준에 부합합니다.
- 플레이어 버퍼링 - 표준 플레이어 버퍼링 동작(재생보다 약간 앞선 세그먼트 요청)은 세그먼트 요청 패턴에 따라 적절한 시간에 추적 이벤트를 트리거합니다.

불일치 추적 문제 해결

MediaTailor 서버 측 추적과 타사 지표 간에 불일치가 발견되면 다음 요소를 고려하세요.

- 플레이어 동작 차이 - 플레이어마다 세그먼트 요청 시기에 영향을 미치는 다양한 사전 가져오기 및 버퍼링 전략이 있을 수 있습니다.
- 네트워크 조건 - 네트워크 조건이 좋지 않으면 플레이어가 세그먼트를 예상과 여러 번 또는 다른 간격으로 요청할 수 있습니다.
- CDN 구성 - /v1/segment 요청의 잘못된 CDN 캐싱으로 인해 추적 이벤트가 누락되거나 중복될 수 있습니다.
- 세션 관리 - 각 재생 세션이 이벤트 충돌 추적을 방지하기 위해 고유한 세션 식별자를 사용하는지 확인합니다.

자세한 문제 해결 지침은 [섹션을 참조하세요](#) [일반적인 문제 해결](#).

MediaTailor 서버 측 추적 기능

AWS Elemental MediaTailor 는 이러한 통합 서버 측 추적 기능을 자동으로 적용하여 광고 측정 정확도와 신뢰성을 최적화합니다. 시스템은 중복 비컨을 방지하고, 대량 기간 동안 트래픽을 관리하고, 적절한 이벤트 시퀀싱을 유지하고, 사용자의 구성 없이 포괄적인 성능 모니터링을 제공합니다. 광고 결정 서버(ADS)가 VAST 응답에서 추적 비컨을 제공하도록 하기만 하면 됩니다.

Note

이러한 기능은 2025년 9월 30일부터 신규 고객에게 제공됩니다. 기존 고객은 지속적인 서비스 개선의 일환으로 2025년 내내 액세스할 수 있습니다. 이러한 기능에 즉시 액세스하려면 [AWS Support](#)에 문의하십시오.

Note

이러한 기능은 스티칭(SSAI) 광고 삽입 방법과 서버 가이드(SGAI) 광고 삽입 방법 모두에 적용됩니다. 비컨 유형과 타이밍은 두 모드 모두에서 동일합니다. MediaTailor가 비컨을 트리거하는 방식이 다릅니다. SGAI 서버 측 비컨에 [서버 가이드 광고 삽입\(SGAI\)을 사용한 서버 측 추적](#) 대한 자세한 내용은 섹션을 참조하세요.

비컨 중복 제거

MediaTailor는 동일한 광고 이벤트에 대한 중복 비컨 실행을 방지합니다. 서버 측 추적 시스템은 각 노출, 사분위수 및 완료 비컨을 광고 보기 세션당 한 번만 전송합니다. 비디오 플레이어는 네트워크 조건, 비트 전송률 변경 또는 버퍼링 전략으로 인해 동일한 광고 세그먼트를 여러 번 요청하면 MediaTailor는 실행된 비컨을 추적하고 중복 전송을 차단합니다.

중복 제거는 부풀려진 비컨 수를 유발하는 일반적인 시나리오를 자동으로 해결합니다.

- 적응형 비트레이트 스트리밍 - 플레이어가 동일한 광고 세그먼트의 다양한 품질 변형을 다운로드하는 경우
- 네트워크 재시도 시나리오 - 플레이어가 네트워크 문제 또는 제한 시간으로 인해 세그먼트를 재요청하는 경우
- 플레이어 버퍼링 전략 - 플레이어가 버퍼링 목적으로 세그먼트를 미리 가져오거나 다시 가져오는 경우

시스템은 플레이어가 서로 다른 품질 수준 간에 전환하더라도 노출 비컨을 한 번만 발사하도록 설계되었습니다.

적응형 제한 및 비컨 재시도

MediaTailor는 서버 응답 지표를 기반으로 비컨 트래픽 속도를 자동으로 관리합니다. 시스템은 HTTP 응답 패턴, 연결 제한 시간 및 오류 코드를 모니터링하여 정체를 감지한 다음 그에 따라 트래픽 속도를 조정합니다. 시스템이 서버 스트레스 지표를 식별하면 영향을 받는 도메인의 트래픽 속도가 줄어들고 서버에서 용량 개선이 입증되면 자동으로 속도가 증가합니다.

시스템은 다음 표시기를 사용하여 서버 상태를 모니터링합니다.

- HTTP 연결 제한 시간 - 측정 플랫폼이 예상 기간 내에 응답하지 않는 경우
- 오류 응답 코드 - 서버 과부하를 나타내는 503, 504 및 507 응답입니다. 또한 광고 서버는 완전한 호환성을 위해 이러한 오류 코드를 지원해야 합니다.
- 응답 패턴 - 용량 문제를 나타내는 측정 플랫폼 성능 변경

재시도 동작은 시도 사이에 최소 30초의 지연 시간으로 최대 1시간 동안 전송을 자동으로 시도합니다. 이 재시도 동작은 구성할 수 없습니다.

초당 비컨 트래픽 관리

TPS 제한을 설정하여 비컨 전송 속도를 제어할 수 있습니다. 서버 측 추적 기능에 대해 구성 가능한 유일한 설정입니다. 계정 수준 제한은 모든 측정 파트너에게 전송된 총 광고 추적 요청 수를 제한합니다. MediaTailor는 최소 TPS 한도 10,000을 적용하여 엔터프라이즈 규모의 작업에 충분한 용량을 제공합니다.

AWS 지원 티켓을 제출하여 다음 정보와 함께 TPS 한도를 설정합니다.

- AWS 계정 ID - 특정 계정 식별자
- 대상 리전 - TPS 제한을 적용하려는 AWS 리전
- 원하는 TPS 임계값 - 필요한 초당 트랜잭션 한도(최소 10,000)

기본적으로 TPS 제한은 없습니다. 광고 결정 서버(ADS)에 필요한 경우 TPS 제한을 요청할 수 있지만 제한은 10,000TPS보다 커야 합니다. MediaTailor는 지정된 제한을 초과하지 않지만 해당 한도까지 일관된 처리량을 보장하지 않습니다. 광고 결정 서버는 지원할 수 있는 TPS 제한을 알려줍니다.

순서 내 비컨

MediaTailor는 광고 추적 이벤트의 순차적 전송을 자동으로 유지합니다. 시스템은 네트워크 문제, 재시도 또는 트래픽 관리가 발생하는 경우에도 비컨 순서를 유지합니다. 이렇게 하면 측정 파트너가 정확한 분석을 위해 올바른 순서로 이벤트를 수신할 수 있습니다.

시스템은 표준 산업 비컨 시퀀스를 따릅니다.

1. 이벤트 시작 - 광고 재생이 시작될 때 실행
2. 첫 번째 사분위수 이벤트 - 25% 광고 완료 시 발사
3. 중간 이벤트 - 50% 광고 완료 시 발사
4. 세 번째 사분위수 이벤트 - 75% 광고 완료 시 발사
5. 완료 이벤트 - 광고가 완료되면 실행

이러한 기능은 자동으로 함께 작동합니다.

- 적절한 순서를 유지하기 위해 제한 중에 비컨이 보관됩니다.
- 각 측정 파트너 도메인에는 속도 조정 중 중단을 방지하기 위한 별도의 이벤트 대기열이 있습니다.
- 중복 제거는 시간 순서를 유지하면서 이벤트 유형 및 타임라인 위치를 추적합니다.

클라이언트 측 광고 추적

AWS Elemental MediaTailor 클라이언트 측 추적 API를 사용하면 스트리밍 워크플로에서 광고 시간 동안 플레이어 제어를 통합할 수 있습니다. 클라이언트 측 추적에서 플레이어 또는 클라이언트는 노출 및 사분위 광고 비컨과 같은 추적 이벤트를 광고 결정 서버(ADS) 및 기타 광고 확인 엔터티로 내보냅니다. 이러한 이벤트는 각 브레이크 내의 전체 광고 브레이크 상태와 개별 광고 시간을 모두 추적합니다. 노출 및 사분위수(ADS) 및 기타 광고 확인 엔터티에 대한 자세한 내용을 참조하세요. 노출 및 사분위 광고 비컨에 대한 자세한 내용은 섹션을 참조하세요 [클라이언트 측 비컨](#). ADS 및 기타 광고 확인 엔터티에 대한 자세한 내용은 섹션을 참조하세요 [클라이언트 측 광고 추적 통합](#).

클라이언트 측 추적을 위해 플레이어 파라미터 및 세션 데이터를 ADS에 전달하는 방법에 대한 자세한 내용은 [ADS 요청에 대한 MediaTailor 플레이어 변수](#) 및 섹션을 참조하세요 [ADS 요청에 대한 MediaTailor 세션 변수](#).

클라이언트 측 추적을 사용하면 다음과 같은 기능을 사용할 수 있습니다.

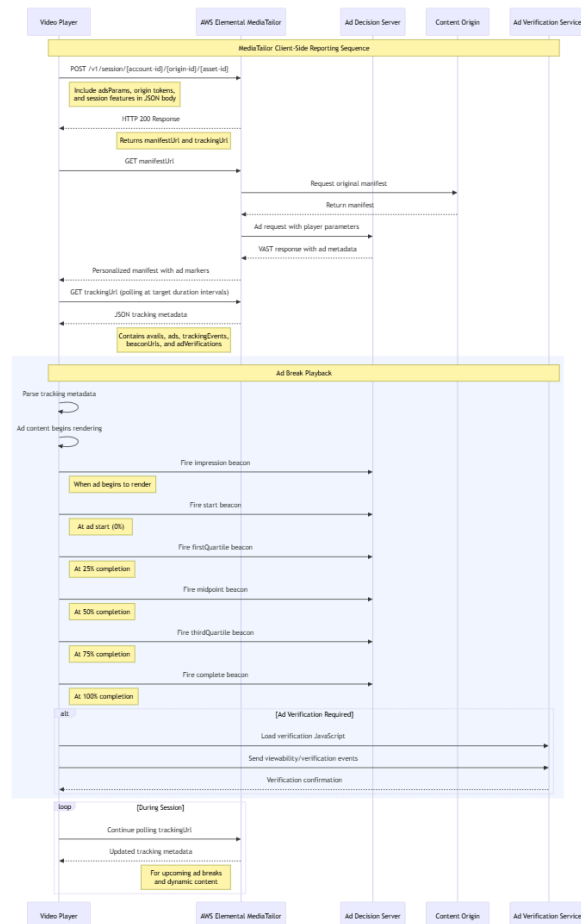
- 애드 브레이크 카운트다운 타이머 - 자세한 내용은 섹션을 참조하세요 [광고 카운트다운 타이머](#).

- 광고 클릭 - 자세한 내용은 섹션을 참조하세요 [광고 클릭](#).
- 컴패니언 광고 표시 - 자세한 내용은 섹션을 참조하세요 [컴패니언 광고](#).
- 건너뛰기 가능한 광고 - 자세한 내용은 섹션을 참조하세요 [건너뛰기 가능한 광고](#).
- 프라이버시 규정 준수를 위한 VAST 아이콘 표시 - 자세한 내용은 섹션을 참조하세요 [Google WTA\(Why This Ad\)용 아이콘](#).
- 광고 중 플레이어 스크러빙 제어 - 자세한 내용은 섹션을 참조하세요 [스크러빙](#).

MediaTailor 클라이언트 측 추적 API를 사용하면 클라이언트 측 추적 외에도 기능을 활성화하는 메타 데이터를 재생 디바이스로 전송할 수 있습니다.

클라이언트 측 보고 워크플로

다음 다이어그램은 세션 초기화부터 광고 재생 및 비컨에 이르기까지 전체 클라이언트 측 보고 워크플로를 보여줍니다.



클라이언트 측 보고 워크플로에는 다음 단계가 포함됩니다.

1. 세션 초기화 - 비디오 플레이어는 adsParams, 오리진 토큰 및 세션 기능을 포함한 JSON 메타데이터를 사용하여 MediaTailor 세션 엔드포인트에 POST 요청을 보냅니다. MediaTailor는 세션에 trackingUrl 대해 manifestUrl 및 로 응답합니다.
2. 매니페스트 요청 및 광고 결정 - 플레이어가 MediaTailor에서 개인화된 매니페스트를 요청합니다. MediaTailor는 오리진에서 원본 콘텐츠 매니페스트를 요청하고, 플레이어 파라미터를 사용하여 광고 요청을 Ad Decision Server(ADS)에 보내고, 광고 메타데이터가 포함된 VAST 응답을 수신하고, 광고 마커가 포함된 맞춤형 매니페스트를 플레이어에게 전달합니다.
3. 추적 데이터 검색 - 플레이어는 추적 URL을 정기적으로 폴링합니다(HLS의 대상 기간 또는 DASH의 최소 업데이트 기간과 일치). MediaTailor는 가용성, 광고, 추적 이벤트, 비컨 URLs 및 광고 확인 데이터가 포함된 JSON 추적 메타데이터를 반환합니다.
4. 광고 재생 및 비컨 - 광고가 중단되는 동안 플레이어는 추적 메타데이터를 구문 분석하고, 광고 렌더링이 시작될 때 노출 비컨을 실행하고, 적절한 시간에 사분위 비컨(시작, firstQuartile, 중간점, thirdQuartile, 완료)을 실행하고, 필요한 경우 광고 확인 JavaScript를 로드 및 실행하고, 가시성/검증 이벤트를 타사 확인 서비스에 전송합니다.
5. 연속 폴링 - 플레이어는 세션 전체에서 추적 URL을 계속 폴링하여 예정된 광고 시간 및 동적 콘텐츠에 대한 업데이트된 메타데이터를 수신합니다.

이 워크플로를 사용하면 개인 정보 보호를 위해 광고 카운트다운 타이머, 클릭 기능, 컴패니언 광고, 건너뛰기 가능한 광고, VAST 아이콘 표시와 같은 고급 기능을 사용할 수 있습니다.

주제

- [클라이언트 측 추적 활성화](#)
- [광고 서버 파라미터](#)
- [오리진 상호 작용 쿼리 파라미터](#)
- [세션 구성 기능](#)
- [클라이언트 측 추적 모범 사례](#)
- [클라이언트 측 광고 추적 스키마 및 속성](#)
- [광고 추적 활동 타이밍](#)
- [클라이언트 측 광고 추적을 위한 플레이어 제어 및 기능](#)
- [클라이언트 측 비컨](#)
- [서버 측 광고 비컨이 있는 하이브리드 모드](#)
- [클라이언트 측 광고 추적 통합](#)

- [GetTracking을 사용하여 광고 비컨 페이징](#)

클라이언트 측 추적 활성화

각 세션에 대해 클라이언트 측 추적을 활성화합니다. 플레이어는 MediaTailor 구성의 세션 초기화 접두사 엔드포인트에 HTTPPOST를 생성합니다. 선택적으로 플레이어는 광고 호출, 매니페스트의 오리진 호출, 세션 수준에서 MediaTailor 기능 호출 또는 비활성화 시 MediaTailor가 사용할 추가 메타데이터를 전송할 수 있습니다.

다음 예제에서는 JSON 메타데이터의 구조를 보여줍니다.

```
{
  "adsParams": {
    "param1": "value1",
    "param2": "value2",
    # 'adsParams' is case sensitive
    # key is not case sensitive
    # Values can contain spaces. For example, 'value 2'
    # is an allowed value.
  },
  "origin_access_token": "abc123", # this is an example of a query parameter designated
  # for the origin
  "overlayAvails": "on"
  # 'overlayAvails' is case sensitive. This is an
  # example of a feature that is enabled at the session level.
}
```

MediaTailor 콘솔 또는 API를 사용하여 이러한 파라미터를 참조하도록 ADS 요청 템플릿 URL을 구성합니다. 다음 예제에서 `player_params.param1`는의 플레이어 파라미터이고 `param1player_params.param2`는의 플레이어 파라미터입니다`param2`.

```
https://my.ads.com/path?param1=[player_params.param1]&param2=[player_params.param2]
```

광고 서버 파라미터

JSON 구조의 최상위 수준은 `adsParams` JSON 객체입니다. 이 객체 내에는 MediaTailor가 모든 세션 요청에서 읽고 광고 서버로 전송할 수 있는 키/값 페어가 있습니다. MediaTailor는 다음 광고 서버를 지원합니다.

- Google Ad Manager
- SpringServe
- FreeWheel

- 퍼블릭

오리진 상호 작용 쿼리 파라미터

, adsParams availSuppression 및 overlayAvails는 쿼리 파라미터의 형태로 오리진 요청 URL에 추가되지 않습니다. MediaTailor가 오리진에 수행하는 모든 세션 매니페스트 요청에는 이러한 쿼리 파라미터가 포함됩니다. 오리진은 불필요한 쿼리 파라미터를 무시합니다. 예를 들어 MediaTailor는 키/값 페어를 사용하여 오리진에 액세스 토큰을 보낼 수 있습니다.

세션 구성 기능

세션 초기화 JSON 구조를 사용하여, 및 overlayAvails availSuppression와 같은 MediaTailor 기능을 활성화, 비활성화 또는 재정의합니다 adSignaling. 세션 초기화 중에 전달된 모든 기능 구성은 MediaTailor 구성 수준에서 설정을 재정의합니다.

Note

세션 초기화 시 MediaTailor에 제출된 메타데이터는 변경할 수 없으며 세션 기간 동안 추가 메타데이터를 추가할 수 없습니다. SCTE-35 마커를 사용하여 세션 중에 변경되는 데이터를 전달합니다. 자세한 내용은 [ADS 요청에 대한 MediaTailor 세션 변수](#) 단원을 참조하십시오.

Example: HLS에 대한 클라이언트 측 광고 추적 수행

```
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: DASH에 대한 클라이언트 측 광고 추적 수행

```
POST mediatailorURL/v1/session/hashed-account-id/origin-id/asset-id.mpd

{
```

```

      "adsParams": {
        "deviceType": "androidmobile",
        "uid": "xjhddli-9189901-uic"
      }
    }
  }

```

보고 모드 파라미터

요청 본문에 `reportingMode` 파라미터를 포함하여 세션을 초기화할 때 보고 모드를 지정할 수 있습니다. 이 파라미터는 MediaTailor가 세션에 대한 클라이언트 측 또는 서버 측 광고 추적을 수행할지 여부를 제어합니다.

- `client` - 플레이어가 광고 추적을 수행하고 비컨을 광고 서버로 보냅니다. 지정되지 않은 경우 기본 모드 `reportingMode`입니다.
- `server` - MediaTailor는 서버 측 광고 추적을 수행하고 비컨을 광고 서버로 직접 전송합니다.

Example서버 측 보고 모드를 사용한 세션 초기화

POST *mediatailorURL*/v1/session/*hashed-account-id/origin-id/asset-id*.m3u8

```

{
  "adsParams": {
    "deviceType": "ipad",
    "uid": "abdgfdyei-2283004-ueu"
  },
  "reportingMode": "server"
}

```

Example클라이언트 측 보고 모드를 사용한 세션 초기화(명시적)

POST *mediatailorURL*/v1/session/*hashed-account-id/origin-id/asset-id*.mpd

```

{
  "adsParams": {
    "deviceType": "androidmobile",
    "uid": "xjhddli-9189901-uic"
  },
  "reportingMode": "client"
}

```

Note

reportingMode 파라미터는 세션 초기화 시 설정되며 세션 중에는 변경할 수 없습니다. 지정하지 않으면 MediaTailorreportingMode는 이전 버전과의 호환성을 유지하기 위해 기본적으로 클라이언트 측 보고를 사용합니다.

성공적인 응답은 응답 본문이 200 있는 HTTP입니다. 본문에는 manifestUrl 및 trackingUrl 키가 있는 JSON 객체가 포함되어 있습니다. 값은 플레이어가 재생 및 광고 이벤트 추적 모두에 사용할 수 있는 상대 URLs입니다.

```
{
  "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"
}
```

클라이언트 측 추적 스키마에 대한 자세한 내용은 섹션을 참조하세요 [클라이언트 측 광고 추적 스키마 및 속성](#).

클라이언트 측 추적 모범 사례

이 섹션에서는 라이브 및 VOD 워크플로 모두에 대한 MediaTailor의 클라이언트 측 추적 모범 사례를 간략하게 설명합니다.

라이브 워크플로

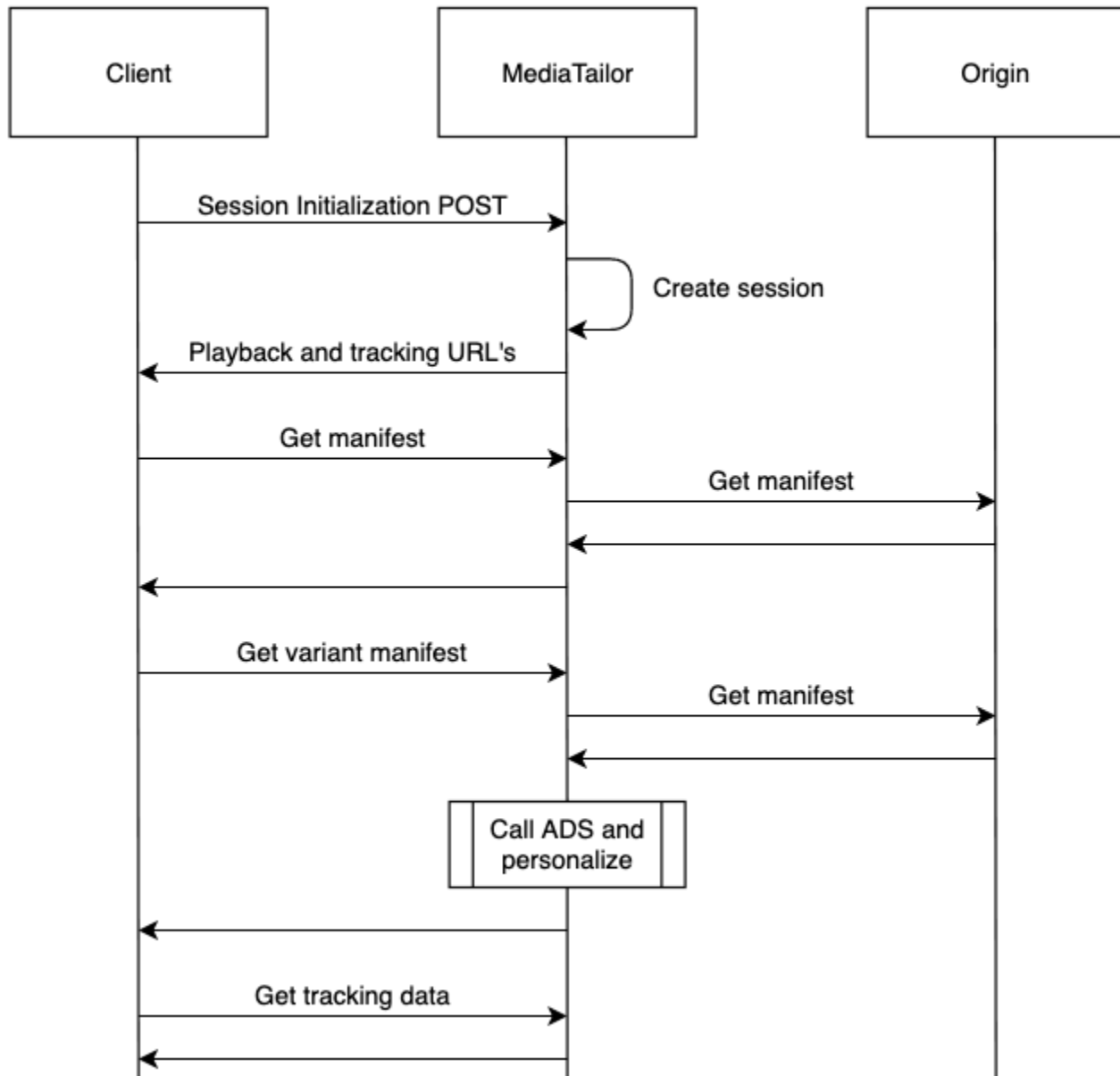
항상 최신 광고 추적 메타데이터를 보유하기 위해 HLS의 모든 대상 기간 또는 DASH의 최소 업데이트 기간과 일치하는 간격으로 추적 엔드포인트를 폴링합니다. 이 간격을 일치시키는 것은 크리에이티브에 대화형 또는 오버레이 구성 요소가 있을 수 있는 워크플로에서 특히 중요합니다.

Note

일부 플레이어는 폴링의 대안으로 사용할 수 있는 이벤트 리스너를 지원합니다. 예를 들어 각 세션에 대해 MediaTailor 광고 ID 장식 기능을 활성화해야 합니다. 자세한 내용은 [광고 ID 장식](#) 단원을 참조하십시오. 이 기능을 사용하면 가용 구간의 각 광고 위에 날짜 범위(HLS) 또는 이벤트 요소(DASH) 식별자가 추가됩니다. 플레이어는 이러한 매니페스트 태그를 프롬프트로 사용하여 세션에 대한 MediaTailor 추적 엔드포인트를 호출할 수 있습니다.

VOD 워크플로

세션 초기화가 성공하고 MediaTailor가 미디어가 포함된 첫 번째 매니페스트를 수신한 후 추적 엔드포인트를 한 번만 호출하면 됩니다.



서버 가이드 광고 삽입

서버 가이드 광고 삽입(SGAI) 세션은 GetTracking API를 사용하지 않습니다. 대신를 사용하면 `aws.reportingMode=CLIENT` MediaTailor는 플레이어가 광고 콘텐츠를 요청할 때 각 자산 목록 응답의 TRACKING 섹션에 추적 정보를 제공합니다. 세션 초기화 응답에는 포함되지 않습니다 `trackingUrl`.

클라이언트 측 추적 SGAI 세션에 대한 자산 목록 응답의 구조는 다음과 같습니다.

```
{
  "ASSETS": [
    { "DURATION": 20.0, "URI": "https://cdn.example.com/ad1/master.m3u8" },
    { "DURATION": 10.0, "URI": "https://cdn.example.com/ad2/master.m3u8" }
  ],
  "TRACKING": {
    ...VAST tracking events and beacon URLs for each ad...
  }
}
```

SGAI 메서드에 대한 클라이언트 측 추적을 구현하는 경우:

- 를 호출하는 대신 자산 목록 응답에서 TRACKING 섹션 구문 분석 GetTracking
- 광고 이벤트 보고를 위해 자산 목록에 제공된 추적 URLs 사용
- 플레이어의 실제 광고 재생 이벤트를 기반으로 하는 발사 추적 비컨
- 자산 목록을 가져올 때 각 광고 시간에 대한 추적을 독립적으로 처리

Important

TRACKING 섹션은이 `aws.reportingMode=CLIENT` 설정된 경우에만 자산 목록에 포함됩니다. 서버 측 보고(SGAI의 기본값)를 사용하는 경우 MediaTailor는 TRACKING 섹션을 생략하고 대신 광고 URIs에 비컨 데이터를 포함합니다. 자세한 내용은 [서버 가이드 광고 삽입\(SGAI\)을 사용한 서버 측 추적](#)을 참조하세요.

클라이언트 측 광고 추적 스키마 및 속성

MediaTailor 클라이언트 측 광고 추적 기능을 사용하면 자세한 클라이언트 측 광고 추적 데이터를 플레이어 환경에 통합할 수 있습니다. 다음 섹션에서는 전체 광고 추적 스키마와 스키마를 구성하는 특정 속성 및 값을 다룹니다.

스키마

다음 JSON 구조는 MediaTailor 클라이언트 측 광고 추적 스키마를 보여줍니다. 이 표현은 여러 부분 간의 관계를 이해하는 데 도움이 되는 스키마의 중첩 구조를 보여줍니다.

각 속성에 대한 자세한 내용은 섹션을 참조하세요 [the section called “속성”](#).


```
{
  "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": {}
}

```

속성

다음 표에는 클라이언트 측 추적 API의 속성, 정의, 값 유형 및 예제가 나열되어 있습니다.

속성	정의	값 유형	예제
adID	- HLS - 광고 시작과 연결된 시퀀스 번호 - DASH - 광고의 기간 ID 경로: /avails/ads/adId VAST 매핑: 없음	문자열	10
adBreakTrackingEvents	VAST 응답에서 VMAP 추적 이벤트를 전달하는 배열입니다. 자세한 내용은 VMAP 1.0 사양의 섹션 2.3.3을 참조하세요. 경로: /avails/ads/adBreakTrackingEvents	배열	[]
adMarkerDuration	매니페스트의 광고 마커에서 관찰된 가용 기간입니다.	문자열	30
adParameters	MediaTailor가 플레이어에게 전달하는 VAST VPAID의 광고 파라미터 문자열입니다. 경로: /avails/ads/adParameters VAST 매핑: VAST/Ad/Inline/Creatives/Creative/Linear/AdParameters	문자열	

속성	정의	값 유형	예제
adProgram DateTime	- HLS - 광고의 첫 번째 미디어 시퀀스를 나타내는 ISO/IEC 8601:2004 형식의 날짜입니다. - DASH - [지정되지 않음]	문자열	
ads	가능 구간을 구성하는 광고 객체가 포함된 배열입니다. 광고는 매니페스트에 나타나는 순서대로 나열됩니다. 경로: /avails/ads	배열	[]
adSystem	광고를 제공하는 시스템의 이름입니다.  Important 값을 제공해야 합니다. 값을 제공하지 않으면 문제가 발생할 수 있습니다.	문자열	myADS
adTitle	광고의 제목입니다.	문자열	ad1

속성	정의	값 유형	예제
adVerifications	크리에이티브 재생을 확인하기 위해 타사 측정 코드를 실행하는 데 필요한 리소스와 메타데이터를 포함합니다. 이 속성에 대한 자세한 내용은 VAST 4.2 사양의 섹션 3.16을 참조하세요. MediaTailor는 VAST 3 확장 노드 adVerifications 로를 지원합니다. 경로: /avails/ads/adVerifications VAST 매핑: VAST/Ad/Inline/AdVerifications	배열	[]
altText	컴패니언 광고 이미지의 대체 텍스트입니다. 이 텍스트를 사용하면 시각 장애가 있는 플레이어가 이미지에 대한 설명을 다시 읽을 수 있습니다. 경로: /avails/ads/companionAds/altText	문자열	video sequence advertising sneakers
apiFramework	이 광고가 VPAID 광고임을 플레이어에게 알리려면 VPAID 로 설정합니다. 스키마의 여러 위치에 나타날 수 있습니다.	문자열	VPAID

속성	정의	값 유형	예제
availID	- HLS - 광고 시간의 시작과 연결된 시퀀스 번호입니다. - DASH - 광고 시간의 기간 ID로, 일반적으로 광고로 대체할 콘텐츠의 기간 ID입니다. 경로: /avails/availID	문자열	34 - PT34S_1
avails	활성 매니페스트 창에 표시되는 광고 중단 객체 또는 가능 구간을 포함하는 배열입니다. 가용 구간은 매니페스트에 나타나는 순서대로 나열됩니다. 경로: /avails	배열	[]
adType	광고의 유형입니다. 경로: /avails/adType 및 /avails/ads/adType	문자열	
dateTime	광고 가능성 또는 광고 시작을 위한 ISO 8601초 형식의 프로그램 날짜 시간입니다. 경로: /avails/dateTime 및 /avails/ads/dateTime	문자열	
duration	길이, ISO 8601초 형식. 응답에는 전체 광고 시간 및 각 광고 및 비컨에 대한 지속 시간이 포함되지만 비컨 지속 시간은 항상 0입니다. 경로: /avails/duration 및 /avails/ads/duration	문자열	15.015

속성	정의	값 유형	예제
durationInSeconds	초 형식의 길이입니다. 경로: /avails/durationInSeconds 및 /avails/ads/durationInSeconds	숫자	
extensions	광고 서버에서 사용하는 VAST의 사용자 지정 확장입니다. 확장에 대한 자세한 내용은 VAST 4.2 사양의 섹션 3.18 을 참조하세요. 경로: /avails/ads/extensions VAST 매핑: VAST/Ad/Inline/Extensions	배열	[]
icons	광고의 아이콘 요소입니다. 경로: /avails/ads/icons VAST 매핑: VAST/Ad/Inline/Creatives/Creative/Linear/Icons	배열	
mediaFiles	플레이어가 광고 시간에 필요한 비디오 및 기타 자산. 경로: /avails/ads/mediaFiles	객체	
nonLinearAvails	비선형 광고 가능 객체 배열입니다. 경로: /nonLinearAvails	배열	

속성	정의	값 유형	예제
<code>executableResource</code>	확인을 위한 실행 가능한 리소스입니다. 경로: <code>/avails/ads/adVerifications/executableResource</code> VAST 매핑: <code>VAST/Ad/Inline/AdVerifications/Verification/ExecutableResource</code>	배열	
<code>javascriptResource</code>	확인을 위한 JavaScript 리소스입니다. 경로: <code>/avails/ads/adVerifications/javascriptResource</code> VAST 매핑: <code>VAST/Ad/Inline/AdVerifications/Verification/JavaScriptResource</code>	배열	
<code>trackingEvents</code>	확인 또는 컴패니언 광고를 위한 이벤트 추적. 경로: <code>/avails/ads/adVerifications/trackingEvents</code> 또는 <code>/avails/ads/companionAds/trackingEvents</code>	배열	

속성	정의	값 유형	예제
vendor	확인 공급업체. 경로: /avails/ads/adVerifications/vendor VAST 매핑: VAST/Ad/Inline/AdVerifications/Verification/@vendor	문자열	
uri	실행 파일 자산, 비디오 자산 또는 추적 엔드포인트를 가리키는 URI입니다. 경로: 스키마의 다양한 위치 VAST 매핑: VAST의 다양한 CDATA 요소	문자열	https://tracking.example.com/impression
verificationParameters	확인 파라미터입니다. 경로: /avails/ads/adVerifications/verificationParameters VAST 매핑: VAST/Ad/Inline/AdVerifications/Verification/VerificationParameters	문자열	
attributes	차원 및 렌더링 모드와 같은 컴패니언 광고 속성. 경로: /avails/ads/companionAds/attributes	객체	

속성	정의	값 유형	예제
companionClickThrough	최종 사용자가 컴패니언 광고를 클릭할 때 미디어 플레이어에서 여는 광고주 페이지의 URL입니다. 경로: /avails/ads/companionAds/companionClickThrough VAST 매핑: VAST/Ad/Inline/Creatives/Creative/CompanionAds/Companion/CompanionClickThrough	문자열	https://aws.amazon.com/
companionClickTracking	companionClickThrough 속성의 추적 URL입니다. 경로: /avails/ads/companionAds/companionClickTracking VAST 매핑: VAST/Ad/Inline/Creatives/Creative/CompanionAds/Companion/CompanionClickTracking	문자열	https://myads.com/beaconing/event=clicktracking

속성	정의	값 유형	예제
htmlResource	스트리밍 공급자의 HTML 페이지 내에 직접 삽입되는 CDATA 인코딩 HTML입니다. 경로: /avails/ads/companionAds/htmlResource VAST 매핑: VAST/Ad/Inline/Creatives/Creative/CompanionAds/Companion/HTMLResource	문자열	<pre><![CDATA[<!doctype html><html><head>< meta name= \"viewport \" content= \"width=1, initial-s cale=1.0, minimum-s cale=1.0, ...]]></pre>
iFrameResource	스트리밍 공급자가 iframe에 로드하는 HTML 리소스 파일의 URL입니다. 경로: /avails/ads/companionAds/iFrameResource VAST 매핑: VAST/Ad/Inline/Creatives/Creative/CompanionAds/Companion/iFrameResource	문자열	
sequence	VAST 응답에서 크리에이티브에 지정된 시퀀스 값입니다. 경로: /avails/ads/companionAds/sequence	문자열	1

속성	정의	값 유형	예제
startTime	ISO 8601초 형식의 시간 위치입니다. HLS의 경우 재생 세션의 시작을 기준으로 합니다. DASH의 경우 이는 매니페스트의 AST(가용성 시작 시간)를 기준으로 합니다. 응답에는 전체 광고 시간과 각 광고 및 비콘에 대한 시작 시간이 포함됩니다. 경로: /avails/startTime 및 /avails/ads/startTime	문자열	PT18.581355S
startTimeInSeconds	초 형식의 시간 위치입니다. HLS의 경우 재생 세션의 시작을 기준으로 합니다. DASH의 경우 이는 매니페스트의 AST(가용성 시작 시간)를 기준으로 합니다. 응답에는 전체 광고 시간과 각 광고 및 비콘에 대한 시작 시간이 포함됩니다. 경로: /avails/startTimeInSeconds 및 /avails/ads/startTimeInSeconds	숫자	18.581
eventId	- HLS - 비콘과 연결된 시퀀스 번호입니다. - DASH - 광고 ptsTime 시작의 입니다.	문자열	

속성	정의	값 유형	예제
event	추적 이벤트의 이름입니다. 경로: /avails/ads/adVerifications/trackingEvents/event 또는 /avails/ads/companionAds/trackingEvents/event	문자열	impression , start, firstQuartile , midpoint, thirdQuartile , complete
beaconUrls	MediaTailor가 광고 비컨을 전송하는 URL입니다. 경로: /avails/ads/mediaFiles/trackingEvents/beaconUrls	배열	
bitrate	비디오 자산의 비트레이트입니다. 이 속성은 일반적으로 실행 가능 자산에 포함되지 않습니다.	문자열	2048
companion Ads	각각 사용할 리소스 파일을 지정하는 하나 이상의 컴패니언 광고 콘텐츠 사양입니다. 컴패니언 광고는 광고 시간과 함께 제공되며 광고 주변의 프레임 또는 배너와 같은 콘텐츠를 제공하여 비디오 근처에 표시합니다. 경로: /avails/ads/companionAds	배열	[]
creativeId	광고에 대한 Creative 태그의 Id 속성 값입니다.	문자열	creative-1
creativeSequence	VAST 응답의 Adid 값에 따라 광고가 재생되어야 하는 시퀀스입니다.	문자열	1

속성	정의	값 유형	예제
dashAvailabilityStartTime	라이브/동적 DASH의 경우 오리진 매니페스트MPD@availabilityStartTime 의 입니다.	문자열	2022-10-05T19:38:39.263Z
delivery	progressive 또는 streaming 프로토콜이 사용 중인지 여부를 나타냅니다.	문자열	progressive
eventType	비컨의 유형입니다. 경로: /avails/ads/mediaFiles/trackingEvents/eventType	문자열	impression
height	비디오 자산의 픽셀 단위 높이입니다.	문자열	360
hlsAnchorMediaSequenceNumber	HLS 오리진 매니페스트에 표시되는 첫 번째/가장 오래된 미디어 시퀀스의 미디어 시퀀스 번호입니다.	문자열	77
maintainAspectRatio	조정 중에 비디오의纵横비를 유지할지 여부를 나타냅니다.	부울	true
mediaFilesList	플레이어가 광고 시간에 필요한 비디오 및 기타 자산을 지정합니다. 경로: /avails/ads/mediaFiles/mediaFilesList	배열	[]
mediaFileUri	실행 파일 자산 또는 비디오 자산을 가리키는 URI입니다.	문자열	https://myad.com/ad/ad134/vpaid.js
mediaType	크리에이티브 또는 컴패니언 자산의 MIME 유형입니다.	문자열	video/mp4

속성	정의	값 유형	예제
meta	광고에 대한 추가 메타데이터입니다.	객체	
mezzanine	VPAID 광고에 포함된 경우 지정된 메자닌 MP4 자산의 URL입니다. 경로: /avails/ads/mediaFiles/mezzanine	문자열	https://gcdn.2mdn.net/video/playback/id/itag/c/k2/file/file.mp4
nextToken	해당 값이 존재할 때 결과의 다음 페이지를 가리키는 토큰의 값입니다.	문자열	UFQz0S44NzNTXzIwMjMtMDctMzFUMTY6NTA6MDYuMzUwNjI2ODQ1Wl8x
nonLinearAds	비디오 콘텐츠와 함께 표시되는 비선형 광고입니다.	배열	[]
nonLinearAdsList	비선형 광고 목록입니다.	배열	[]
scalable	비디오를 다른 차원으로 조정할지 여부를 나타냅니다.	부울	true
skipOffset	플레이어가 사용자가 건너뛰기 제어를 사용할 수 있도록 하는 시기를 식별하는 시간 값입니다.	문자열	00:00:05

속성	정의	값 유형	예제
staticResource	광고 구성 요소에 사용되는 정적 크리에이티브 파일의 URL입니다. 경로: /avails/ads/companionAds/staticResource	문자열	https://very-interactive-ads.com/campaign1/file.json?c=1019113602
vastAdId	Ad 태그의 Id 속성 값입니다.	문자열	ad1
width	비디오 자산의 픽셀 단위 너비입니다.	문자열	640
xPosition	비디오 플레이어 내 아이콘의 가로 위치입니다. 특정 픽셀 값 또는 "왼쪽" 또는 "오른쪽"과 같은 위치일 수 있습니다. 경로: /avails/ads/icons/attributes/xPosition	문자열	left 또는 10
yPosition	비디오 플레이어 내 아이콘의 세로 위치입니다. 특정 픽셀 값 또는 "상단" 또는 "하단"과 같은 위치일 수 있습니다. 경로: /avails/ads/icons/attributes/yPosition	문자열	top 또는 10
iconClicks	아이콘에 대한 클릭 및 추적 정보가 포함되어 있습니다. 경로: /avails/ads/icons/iconClicks	객체	

속성	정의	값 유형	예제
iconClickThrough	최종 사용자가 아이콘을 클릭할 때 미디어 플레이어에 여는 광고자 페이지의 URL입니다. 경로: /avails/ads/icons/iconClicks/iconClickThrough	문자열	https://advertiser.com/landing-page
iconClickTracking	iconClickThrough 속성의 추적 URL입니다. 경로: /avails/ads/icons/iconClicks/iconClickTracking	객체	
iconClickFallbackImages	아이콘을 표시할 수 없는 경우 표시할 폴백 이미지 배열입니다. 경로: /avails/ads/icons/iconClicks/iconClickFallbackImages	배열	
iconViewTracking	아이콘이 표시될 때 추적하기 위한 URL입니다. 경로: /avails/ads/icons/iconViewTracking	문자열	https://tracking.example.com/icon-view
offset	광고 재생 중에 아이콘이 나타나는 시간의 시간 오프셋입니다. 경로: /avails/ads/icons/attributes/offset	문자열	00:00:05

속성	정의	값 유형	예제
program	“AdChoices”와 같은 아이콘과 연결된 프로그램 또는 이니셔티브입니다. 경로: /avails/ads/icons/attributes/program	문자열	AdChoices
pxratio	하이 DPI 디스플레이에 사용되는 아이콘 또는 컴패니언 광고의 픽셀 비율입니다. 경로: /avails/ads/icons/attributes/pxratio 또는 /avails/ads/companionAds/attributes/pxratio	문자열	1 또는 2
type	리소스 또는 확장의 유형입니다. 경로: /avails/ads/extensions/type 또는 /avails/ads/adVerifications/executableResource/type	문자열	text/javascript
content	확장의 콘텐츠입니다. 경로: /avails/ads/extensions/content	문자열	
language	실행 리소스의 프로그래밍 언어입니다. 경로: /avails/ads/adVerifications/executableResource/language	문자열	javascript

속성	정의	값 유형	예제
browserOptional	JavaScript 리소스에 브라우저 지원이 필요한지 여부를 나타냅니다. 경로: /avails/ads/adVerifications/javascriptResource/browserOptional	문자열	true 또는 false
id	스키마의 다양한 요소에 대한 식별자입니다. 경로: /avails/ads/companionAds/attributes/id 또는 /avails/ads/icons/iconClicks/iconClickTracking/id	문자열	companion-1
assetHeight	컴패니언 광고 자산의 높이입니다. 경로: /avails/ads/companionAds/attributes/assetHeight	문자열	250
assetWidth	컴패니언 광고 자산의 너비입니다. 경로: /avails/ads/companionAds/attributes/assetWidth	문자열	300
expandedHeight	확장 시 컴패니언 광고의 높이입니다. 경로: /avails/ads/companionAds/attributes/expandedHeight	문자열	600

속성	정의	값 유형	예제
expandedWidth	확장 시 컴패니언 광고의 너비입니다. 경로: /avails/ads/companionAds/attributes/expandedWidth	문자열	600
renderingMode	컴패니언 광고의 렌더링 모드입니다. 경로: /avails/ads/companionAds/attributes/renderingMode	문자열	default 또는 transparent
adSlotId	컴패니언 광고를 표시해야 하는 광고 슬롯의 ID입니다. 경로: /avails/ads/companionAds/attributes/adSlotId	문자열	banner-1
creativeType	크리에이티브 자산의 MIME 유형입니다. 경로: /avails/ads/icons/staticResource/creativeType	문자열	image/png

광고 추적 활동 타이밍

클라이언트 측 보고를 사용하면 플레이어는 정밀도 수준으로 추적 이벤트(비컨)를 내보내야 합니다. MediaTailor 클라이언트 측 추적 스키마를 사용하면 모든 가용, 광고, 컴패니언, 오버레이 및 추적 이벤트에 대해 타이밍 및 기간 정보가 다양한 형식으로 표시되도록 할 수 있습니다.

플레이어에 대해 다음 MediaTailor 키/값 페어를 사용하여 이벤트 추적과 같은 광고 이벤트 활동을 재생 위치와 정확하게 조정합니다.

- [startTime](#)
- [startTimeInSeconds](#)

- [adProgramDateTime](#)
- [adID/eventId](#)

HLS와 DASH는 및 값을 startTimeInSeconds 다르게 구현startTime합니다.

- HLS - startTime 값은 재생 세션의 시작을 기준으로 합니다. 재생 세션의 시작은 시간 0으로 정의됩니다. 광고의 startTime는 가용 구간까지 이어지는 모든 EXT-INF 세그먼트 지속 시간의 누적 값의 합계입니다. 광고 또는 추적 이벤트가 속하는 세그먼트의 미디어 시퀀스 번호도 클라이언트 측 추적 응답의 adId 또는 eventId에 해당합니다.
- DASH:
 - 라이브/동적 매니페스트 - startTime 값은 DASH 매니페스트MPD@availabilityStartTime의를 기준으로 합니다. MPD@avaibilityStartTime는 스트림을 사용하는 모든 MediaTailor 세션의 타이밍 앵커입니다.
 - VOD/정적 매니페스트 - startTime 값은 재생 세션의 시작을 기준으로 합니다. 재생 세션의 시작은 시간 0으로 정의됩니다. 가능 구간 내의 각 광고는 자체 Period 요소에 포함되어 있습니다. Period 요소에는 클라이언트 측 추적 페이로드의 값과 동일한 startTime 값이 있는 @start 속성이 있습니다. 는 클라이언트 측 추적 응답eventId의 adId 또는 PeriodId 에도 해당합니다.

Example HLS:

다음 예제에서 MediaTailor 세션이 시작되었으며 다음 매니페스트는 클라이언트에 제공된 첫 번째 매니페스트입니다.

```
#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
```

클라이언트 측 추적 JSON 페이로드에는 다음 값이 적용됩니다.

- `startTime`: "PT18.581355S"
- `startTimeInSeconds`: 18.581
- `availProgramDateTime`: "2023-05-03T21:24:41.453Z"
- `adId`: 4603269

Example DASH:

다음 예제에서 MediaTailor 세션은 매니페스트에서 미드롤을 가져옵니다. 광고 기간인 두 번째 기간의 `@start` 속성 값에는 값과 관련된 `MPD@availabilityStartTime` 값이 있습니다. 이 값은 MediaTailor가 모든 세션에 대해 클라이언트 측 추적 응답 `startTime` 필드에 기록하는 값입니다.

```
<?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/eea5485198bf497284559efb8172425e/</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>

```

클라이언트 측 추적 JSON 페이로드에는 다음 값이 적용됩니다.

- startTime: "PT5042H27M59.931S"
- startTimeInSeconds: 18152879.931
- availProgramDateTime: *null*
- adId: 1683151599194_1_1

클라이언트 측 광고 추적을 위한 플레이어 제어 및 기능

MediaTailor 클라이언트 측 추적 메타데이터는 다양한 플레이어 제어 및 기능을 지원합니다. 다음 목록은 인기 있는 플레이어 컨트롤을 설명합니다.

주제

- [스크러빙](#)
- [광고 카운트다운 타이머](#)
- [건너뛰기 가능한 광고](#)
- [광고 클릭](#)
- [کمپانیان 광고](#)

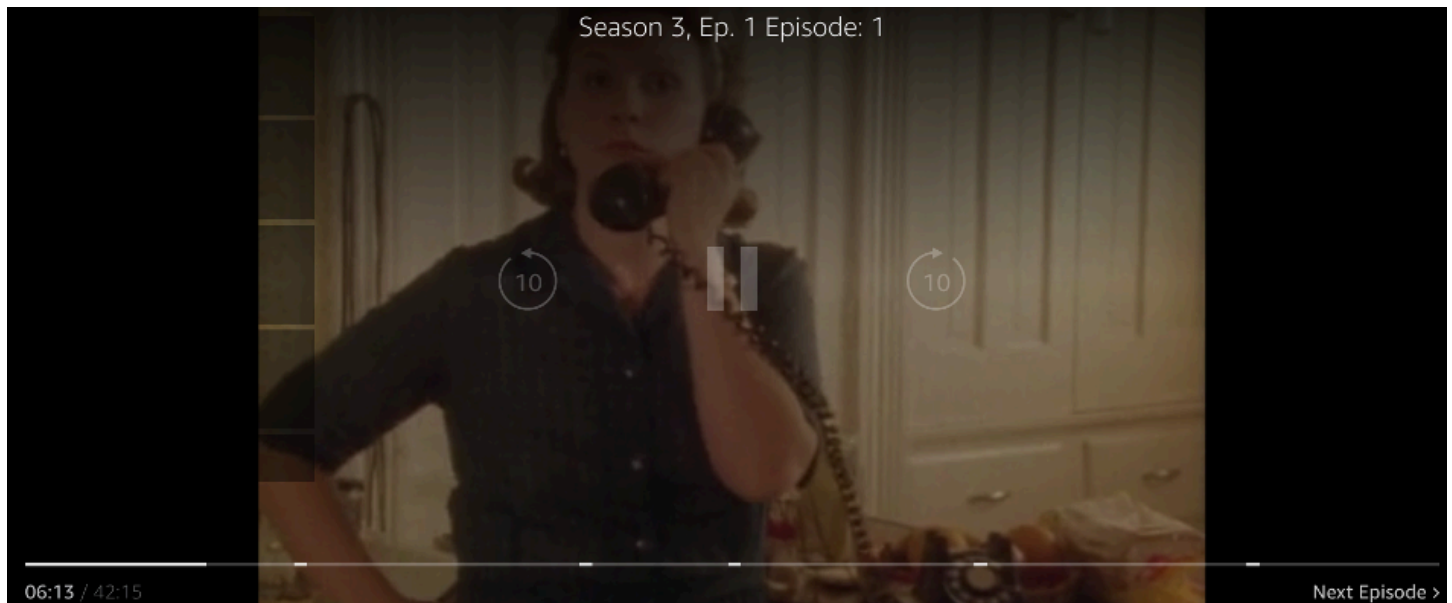
- [대화형 광고\(SIMID\)](#)
- [대화형 광고\(VPAID\)](#)
- [Google WTA\(Why This Ad\)용 아이콘](#)

스크러빙

재생 환경을 개선하기 위해 플레이어는 재생 타임라인에 광고 위치를 표시할 수 있습니다. MediaTailor는 클라이언트 측 추적 응답에서 이러한 광고 위치를 `startTimeInSeconds` 값의 형태로 사용할 수 있도록 합니다.

Note

일부 스트리밍 공급자는 광고 위치를 지나는 스크러빙을 방지합니다.



다음 클라이언트 측 추적 페이로드 JSON 응답은 가용 배열의 루트 JSON 객체 내의 가용(아드 브레이크) 시작 시간을 보여줍니다. 플레이어는 이 데이터를 사용하여 플레이어 타임라인의 광고 시간 위치를 28초로 표시합니다.

```
{
  "avails": [
    {
      "adBreakTrackingEvents": [],

```

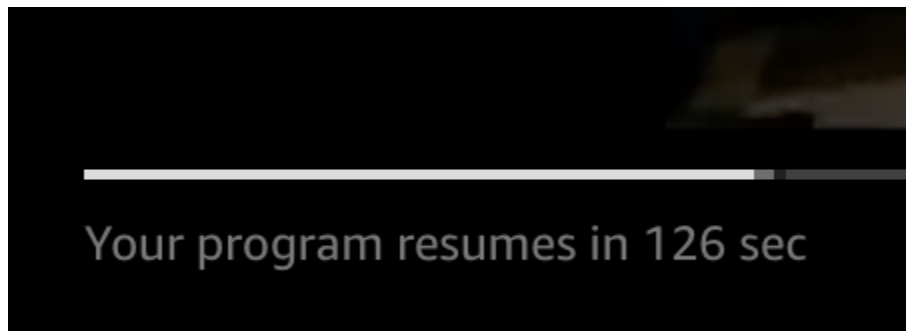
```

    "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": []
}

```

광고 카운트다운 타이머

MediaTailor를 사용하면 광고 카운트다운 타이머를 사용하여 광고 시간을 보는 동안 시청자의 참여를 유지할 수 있습니다. 시청자는 타이머를 사용하여 광고 시간이 종료되고 프로그램이 재개되는 시기를 이해할 수 있습니다.



광고 카운트다운 타이머에서 역할을 수행하는 클라이언트 측 추적 메타데이터의 요소는 `startTime`, `startTimeInSeconds`, 및 `duration`입니다. 플레이어는 이 메타데이터를 개별적으로 추적하는 세션의 경과 시간과 함께 사용하여 타이머를 표시할 시기와 카운트다운해야 하는 값을 결정합니다.

다음 클라이언트 측 추적 페이로드 JSON 응답은 광고 카운트다운 타이머를 표시하는 데 필요한 정보를 보여줍니다.

```
{
```

```

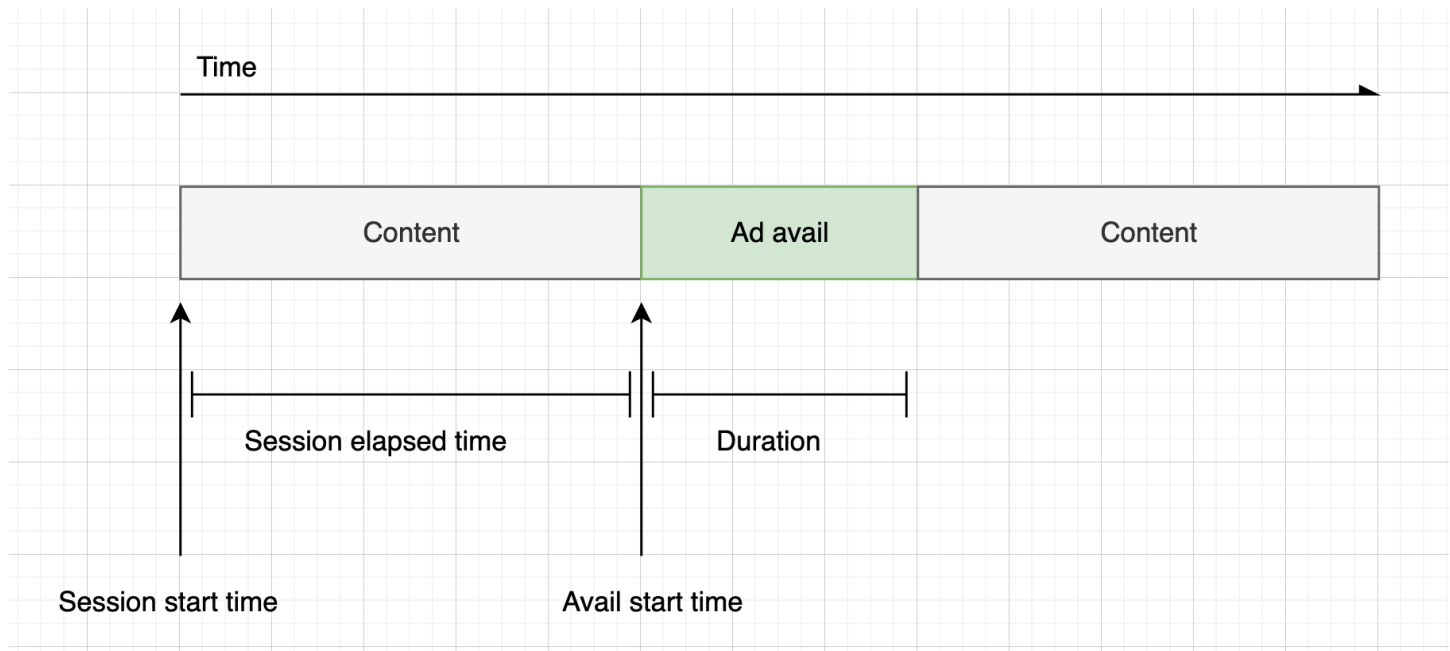
"avails": [
  {
    "adBreakTrackingEvents": [],
    "adMarkerDuration": null,
    "ads": [...],
    "availId": "7",
    "availProgramDateTime": null,
    "duration": "PT30S",
    "durationInSeconds": 30,
    "meta": null,
    "nonLinearAdsList": [],
    "startTime": "PT28S",
    "startTimeInSeconds": 28
  }
],
"dashAvailabilityStartTime": null,
"hlsAnchorMediaSequenceNumber": null,
"nextToken": "UFQxMk0zNC44NjhTXzIwMjMtMDctMjFUMjA6MjM6MDcuNzc1NzE2MzAyWl8x",
"nonLinearAvails": []
}

```

세션의 경과 시간이 가능 구간의 시작 시간에 도달하면 플레이어는 가능 구간의 기간과 일치하는 값과 함께 카운트다운 타이머를 표시합니다. 경과 시간이 가능 구간의 시작 시간을 초과하면 countdown-timer 값이 감소합니다.

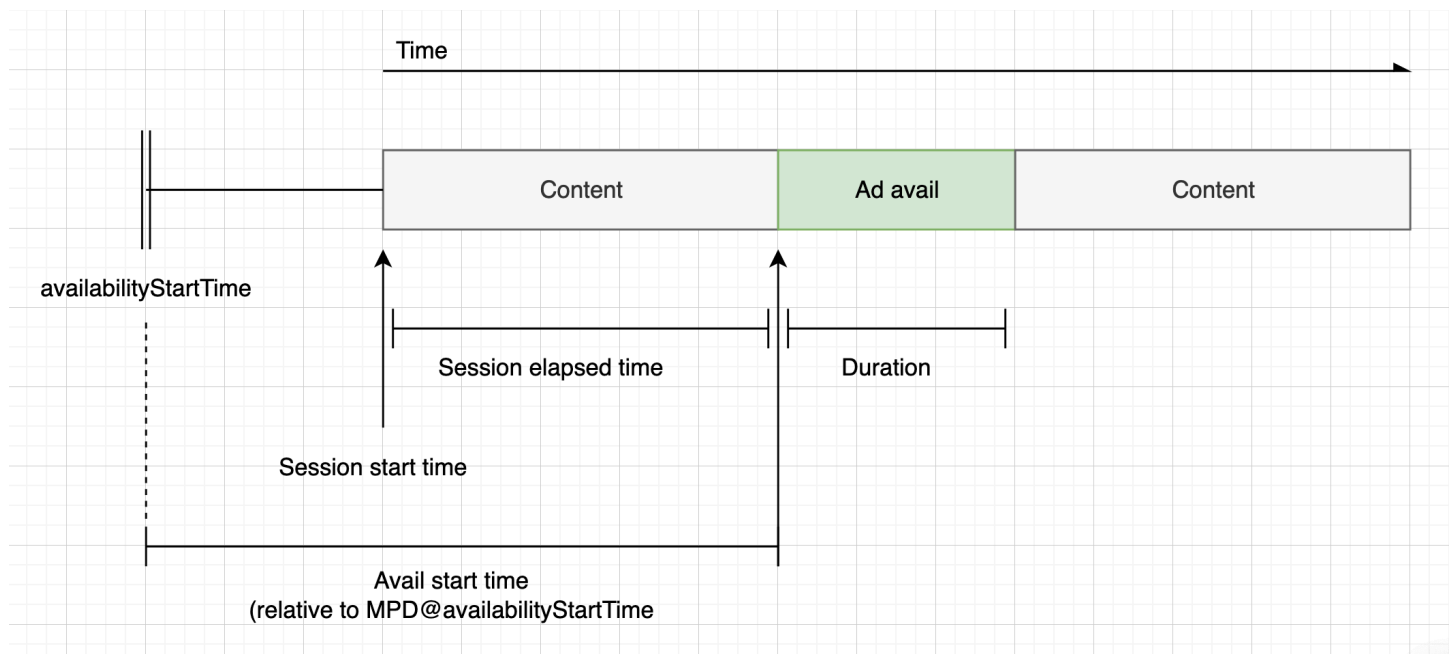
Example 공식: HLS(라이브 및 VOD) 및 DASH(VOD)에 대한 카운트다운 타이머

- $\text{session_start_time} = \text{모든 EXT-INF 지속 시간 값의 합}$
계 - 세 개의 최신 EXT-INF 미디어 시퀀스의 지속 시간 값
- $\text{타이머 값} = \text{duration} - (\text{session_elapsed_time} - \text{startTime})$



Example 공식: DASH의 카운트다운 타이머(실시간)

- $\text{session_start_time} = (\text{최신 세그먼트의 } \text{startTime} + \text{duration}) / \text{timescale} - \text{MPD@suggestedPresentationDelay}$
- 타이머 값 = $\text{duration} - (\text{session_elapsed_time} - \text{startTime})$



건너뛰기 가능한 광고

건너뛰기 가능 광고는 최종 사용자가 일부 광고를 건너뛰어 프로그램 보기를 재개할 수 있는 광고 스팟입니다. VAST에서 `Linear@skipOffset` 속성은 건너뛰기 가능한 광고를 식별합니다.

다음 VAST 응답은 건너뛰기 가능한 광고를 사용하는 방법을 보여줍니다.

```
<?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>
```

다음 클라이언트 측 추적 페이로드 JSON 응답은 ads 배열 내의 광고 메타데이터를 보여줍니다. 배열에는 MediaTailor가 VAST 응답에서 얻은 `skipOffset` 값이 포함됩니다.

```
{
  "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": []  
}
```

광고 클릭

클릭 URIs 사용하면 광고주가 광고가 시청자의 주의를 끄는 데 얼마나 성공했는지 측정할 수 있습니다. 시청자가 진행 중인 광고의 활성 비디오 프레임을 클릭하면 웹 브라우저가 광고주의 홈 페이지 또는 캠페인 랜딩 페이지에 대한 URI를 엽니다. 플레이어 개발자는 클릭하여 자세히 알아볼 수 있는 메시지와 함께 광고 비디오의 버튼 또는 레이블 오버레이와 같은 클릭 동작을 결정합니다. 플레이어 개발자는 최종 사용자가 활성 비디오 프레임을 클릭한 후 광고의 비디오를 일시 중지하는 경우가 많습니다.



Click here for deals on Amazon.com

MediaTailor는 VAST 응답에 반환된 모든 선형 비디오 클릭 이벤트 URLs을 구문 분석하고 사용할 수 있도록 할 수 있습니다. 다음 VAST 응답은 광고 클릭 예제를 보여줍니다.


```
<?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>
```

다음 클라이언트 측 추적 페이로드 JSON 응답은 MediaTailor가 trackingEvents 배열 내에서 클릭 및 클릭 추적 URLs을 표시하는 방법을 보여줍니다. clickThrough 이벤트 유형은 클릭-스루 광고를 나타내고 clickTracking 이벤트 유형은 클릭-추적 URL을 나타냅니다.

```
{
  "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": "UFQz0S4zMz1TXzIwMjMtMDctMzFUMTY6NTQ6MDQu0DA1Mzk2NTI5W18x",
  "nonLinearAvails": []
}

```

컴패니언 광고

동반 광고는 선형 크리에이티브와 함께 표시됩니다. 컴패니언 광고를 사용하면 제품, 로고 및 브랜드 정보를 표시하여 광고 스팟의 효과를 높일 수 있습니다. 디스플레이 광고에는 빠른 응답(QR) 코드와 클릭 가능한 영역이 있어 시청자의 참여를 촉진할 수 있습니다.

MediaTailor는 VAST 응답에서 컴패니언 광고를 지원합니다. 각각 `StaticResource`, `iFrameResource` 및 `HTMLResource` 노드에서 메타데이터를 전달할 수 있습니다.

다음 VAST 응답은 선형 광고 및 컴패니언 광고의 위치와 형식의 예를 보여줍니다.

```

<?xml version="1.0" encoding="UTF-8"?>
<VAST xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="vast.xsd" version="3.0">

```

```

<Ad>
  <InLine>
    ...
    <Creatives>
      <Creative id="1" sequence="1">
        <Linear>
          <Duration>00:00:10</Duration>
          <MediaFiles>
            <MediaFile id="EMT" delivery="progressive" width="640" height="360"
type="video/mp4" bitrate="143" scalable="true" maintainAspectRatio="true"><![
CDATA[https://ads.com/file.mp4]]></MediaFile>
          </MediaFiles>
        </Linear>
      </Creative>
      <Creative id="2" sequence="1">
        <CompanionAds>
          <Companion id="2" width="300" height="250">
            <StaticResource creativeType="image/png"><![CDATA[https://emt.com/
companion/9973499273]]></StaticResource>
            <TrackingEvents>
              <Tracking event="creativeView"><![CDATA[https://beacon.com/1]]></
Tracking>
            </TrackingEvents>
            <CompanionClickThrough><![CDATA[https://beacon.com/2]]></
CompanionClickThrough>
          </Companion>
          <Companion id="3" width="728" height="90">
            <StaticResource creativeType="image/png"><![CDATA[https://emt.com/
companion/1238901823]]></StaticResource>
            <TrackingEvents>
              <Tracking event="creativeView"><![CDATA[https://beacon.com/3]]></
Tracking>
            </TrackingEvents>
            <CompanionClickThrough><![CDATA[https://beacon.com/4]]></
CompanionClickThrough>
          </Companion>
        </CompanionAds>
      </Creative>
    </Creatives>
    ...
  </InLine>
</Ad>
</VAST>

```

데이터는 /avails/x/ads/y/companionAds 목록의 클라이언트 측 추적 응답에 표시됩니다. 각 선형 크리에이티브에는 최대 6개의 컴패니언 광고가 포함될 수 있습니다. 아래 예제와 같이 컴패니언 광고가 목록에 표시됩니다.

Note

애플리케이션 개발자는 크리에이티브가 끝날 때 컴패니언 광고를 명시적으로 제거하거나 언로드하는 로직을 구현하는 것이 가장 좋습니다.

```
{
  "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": "UFQxMFNFmJyAyMy0wNy0wNlQyMToxMDowOC42NzQ4NDA1NjJaXzE%3D",
"nonLinearAvails": []
```

}

대화형 광고(SIMID)

SecureInteractive Media Interface Definition(SIMID)은 대화형 광고국(IAB)의 VAST 4.x 표준에 도입된 대화형 광고의 표준입니다. SIMID는 플레이어의 기본 선형 크리에이티브에서 대화형 요소의 로드를 분리하여 VAST 응답의 두 가지를 모두 참조합니다. MediaTailor는 기본 크리에이티브에 연결하여 재생 환경을 유지하고 클라이언트 측 추적 응답에 대화형 구성 요소의 메타데이터를 배치합니다.

다음 예제 VAST 4 응답에서 SIMID 페이로드는 InteractiveCreativeFile 노드 내에 있습니다.

```
<?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>
```



```

        </Linear>
    </Creative>
</Creatives>
</InLine>
</Ad>
</VAST>

```

다음 VAST 3 응답에서 SIMID 페이로드는 Extensions 노드 내에 있습니다.

```

<?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>

```

다음 클라이언트 측 추적 응답에서 SIMID 데이터가 /avails/x/ads/y/extensions 목록에 나타납니다.

```

{
  "avails": [
    {
      "adBreakTrackingEvents": [],
      "adMarkerDuration": null,
      "ads": [
        {
          "adId": "1",
          "adParameters": "",
          "adProgramDateTime": "2023-07-31T16:53:40.577Z",
          "adSystem": "2.0",
          "adTitle": "Linear SIMID Example",
          "adVerifications": [],
          "companionAds": [],
          "creativeId": "1",
          "creativeSequence": "1",
          "duration": "PT14.982S",
          "durationInSeconds": 14.982,
          "extensions": [
            {
              "content": "<InteractiveCreativeFile type=\"text/html\" apiFramework=\\\"SIMID\\\" variableDuration=\\\"true\\\">\nhttps://interactive-ads.com/interactive-media-ad-sample/sample\_simid.html</InteractiveCreativeFile>",
              "type": "InteractiveCreativeFile"
            }
          ],
          "mediaFiles": {
            "mediaFilesList": [],
            "mezzanine": ""
          },
          "skipOffset": null,
          "startTime": "PT39.339S",

```

```
"startTimeInSeconds": 39.339,
"trackingEvents": [
  {
    "beaconUrls": [
      "https://myads.com/beaconing/event=impression"
    ],
    "duration": "PT14.982S",
    "durationInSeconds": 14.982,
    "eventId": "2698188",
    "eventProgramDateTime": null,
    "eventType": "impression",
    "startTime": "PT39.339S",
    "startTimeInSeconds": 39.339
  },
  {
    "beaconUrls": [
      "https://aws.amazon.com"
    ],
    "duration": "PT14.982S",
    "durationInSeconds": 14.982,
    "eventId": "2698188",
    "eventProgramDateTime": null,
    "eventType": "clickThrough",
    "startTime": "PT39.339S",
    "startTimeInSeconds": 39.339
  },
  {
    "beaconUrls": [
      "https://myads.com/beaconing/event=clicktracking"
    ],
    "duration": "PT14.982S",
    "durationInSeconds": 14.982,
    "eventId": "2698795",
    "eventProgramDateTime": null,
    "eventType": "clickTracking",
    "startTime": "PT39.339S",
    "startTimeInSeconds": 39.339
  }
],
"vastAdId": ""
}
],
"availId": "2698188",
"availProgramDateTime": "2023-07-31T16:53:40.577Z",
```

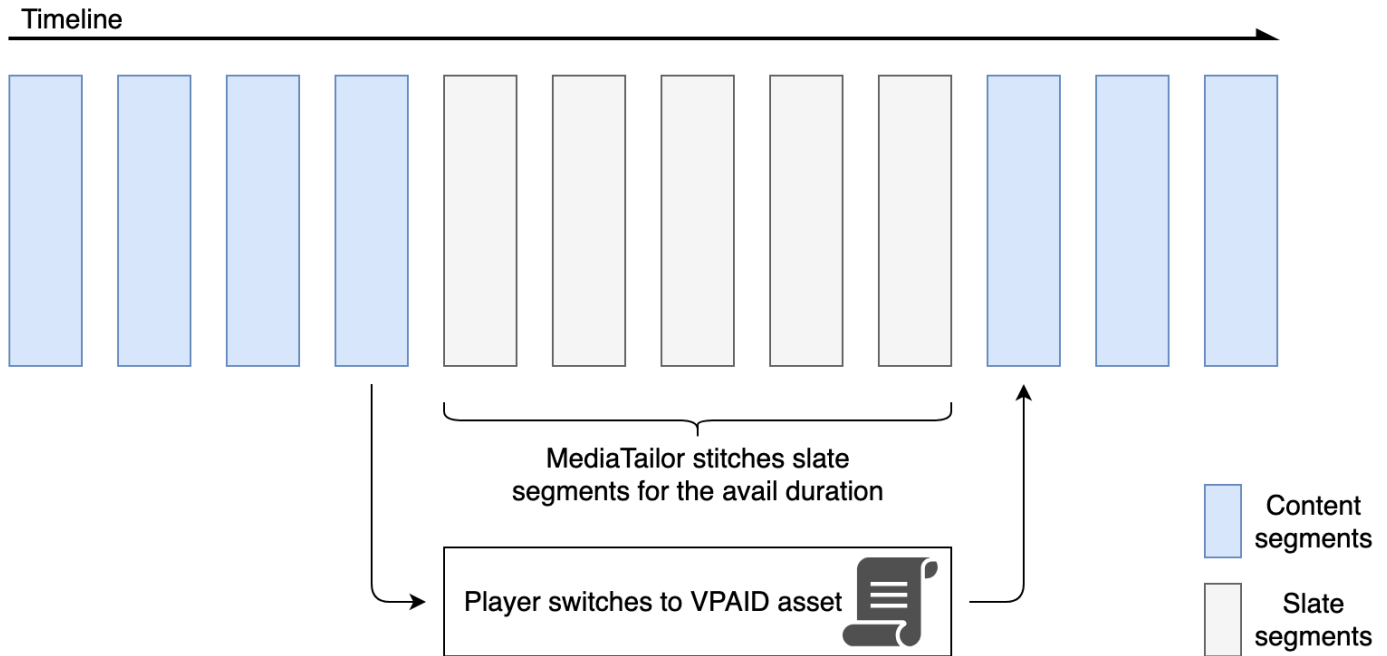
```
    "duration": "PT14.982S",
    "durationInSeconds": 14.982,
    "meta": null,
    "nonLinearAdsList": [],
    "startTime": "PT39.339S",
    "startTimeInSeconds": 39.339
  }
],
"dashAvailabilityStartTime": null,
"hlsAnchorMediaSequenceNumber": null,
"nextToken": "UFQz0S4zMz1TXzIwMjMtMDctMzFUMTY6NTQ6MDQu0DA1Mzk2NTI5Wl8x",
"nonLinearAvails": []
}
```

대화형 광고(VPAID)

비디오 플레이어 광고 인터페이스 정의(VPAID)는 광고와 비디오 플레이어 간의 프로토콜을 지정하여 광고 상호 작용 및 기타 기능을 활성화합니다. 라이브 스트림의 경우 MediaTailor는 가용 시간 동안에 슬레이트 세그먼트를 스티칭하고 비디오 플레이어가 사용하는 클라이언트 측 추적 응답에 VPAID 크리에이티브에 대한 메타데이터를 배치하여 VPAID 형식을 지원합니다. 플레이어는 VPAID 파일을 다운로드하고 선형 크리에이티브를 재생하고 클라이언트의 스크립트를 실행합니다. 플레이어는 슬레이트 세그먼트를 재생해서는 안 됩니다.

Note

VPAID는 VAST 4.1부터 더 이상 사용되지 않습니다.



다음 예제는 VAST 응답의 VPAID 콘텐츠를 보여줍니다.

```
<?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>

```

다음 예제에서는 추적 정보를 보여줍니다.

```

{
  "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": "UFQz0S4zMzlTXzIwMjMtMDctMzFUMTY6NTQ6MDQu0DA1Mzk2NTI5Wl8x",
"nonLinearAvails": []
}{
  "avails": [
    {
      "adBreakTrackingEvents": [],
      "adMarkerDuration": null,
      "ads": [
        {
          "adId": "2922274",
          "adParameters": "",
          "adProgramDateTime": "2023-08-14T19:49:53.998Z",
          "adSystem": "Innovid Ads",
          "adTitle": "VPAID",
          "adVerifications": [],
          "companionAds": [],
          "creativeId": "",
          "creativeSequence": "",
          "duration": "PT16.016S",
          "durationInSeconds": 16.016,

```



```

    "extensions": [],
    "mediaFiles": {
      "mediaFilesList": [
        {
          "apiFramework": "VPAID",
          "bitrate": 0,
          "codec": null,
          "delivery": "progressive",
          "height": 9,
          "id": "",
          "maintainAspectRatio": false,
          "maxBitrate": 0,
          "mediaFileUri": "http://my-ads.com/mobileapps/js/vpaid/1h41kg?
cb=178344c0-8e67-281a-58ca-962e4987cd60&deviceid=&ivc=",
          "mediaType": "application/javascript",
          "minBitrate": 0,
          "scalable": false,
          "width": 16
        }
      ],
      "mezzanine": "http://my-ads.com/mobileapps/js/vpaid/1h41kg?
cb=178344c0-8e67-281a-58ca-962e4987cd60&deviceid=&ivc="
    },
    "skipOffset": null,
    "startTime": "PT8M42.289S",
    "startTimeInSeconds": 522.289,
    "trackingEvents": [
      {
        "beaconUrls": [
          "about:blank"
        ],
        "duration": "PT16.016S",
        "durationInSeconds": 16.016,
        "eventId": "2922274",
        "eventProgramDateTime": null,
        "eventType": "impression",
        "startTime": "PT8M42.289S",
        "startTimeInSeconds": 522.289
      }
    ],
    "vastAdId": "1h41kg"
  }
],
  "availId": "2922274",

```

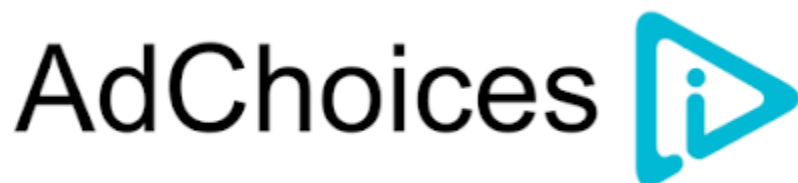
```

    "availProgramDateTime": "2023-08-14T19:49:53.998Z",
    "duration": "PT16.016S",
    "durationInSeconds": 16.016,
    "meta": null,
    "nonLinearAdsList": [],
    "startTime": "PT8M42.289S",
    "startTimeInSeconds": 522.289
  }
],
"dashAvailabilityStartTime": null,
"hlsAnchorMediaSequenceNumber": null,
"nextToken": "UFQ4TTQyLjI4OVNfMjAyMy0wOC0xNFQxOT01MDo0MS4zOTc5MjAzODVaXzE%3D",
"nonLinearAvails": []
}

```

Google WTA(Why This Ad)용 아이콘

AdChoices는 시청자에게 광고의 대상 지정 방식을 포함하여 시청자가 보는 광고에 대한 정보를 제공하는 업계 표준입니다.



MediaTailor 클라이언트 측 추적 API는 VAST 응답의 VAST 확장 노드에 전달되는 아이콘 메타데이터를 지원합니다. VAST 응답의 WTA에 대한 자세한 내용은 [이 샘플 VAST XML 응답](#)을 참조하세요.

Note

MediaTailor는 현재 VAST 버전 3만 지원합니다.

```

<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>

```

다음 예제는 /avails/x/ads/y/extensions 목록의 클라이언트 측 추적 응답을 보여줍니다.

```

{
  "avails": [
    {
      "adBreakTrackingEvents": [],
      "adMarkerDuration": null,
      "ads": [
        {
          "adId": "0",
          "adParameters": "",
          "adProgramDateTime": null,
          "adSystem": "GDFP",
          "adTitle": "Google Why This Ad VAST 3 Sample",
          "adVerifications": [],
          "companionAds": [],
          "creativeId": "7891011",
          "creativeSequence": "1",

```

```

    "duration": "PT10S",
    "durationInSeconds": 10,
    "extensions": [
      {
        "content": "<IconClickFallbackImages program=\"GoogleWhyThisAd\">
          <IconClickFallbackImage height=\"150\" width=\"400\">
            <AltText>Alt icon fallback</AltText>
            <StaticResource creativeType=\"image/png\"><![CDATA[https://storage.googleapis.com/interactive-media-ads/images/wta_dialog.png]]>
          </StaticResource>
        </IconClickFallbackImage>
        </IconClickFallbackImages>
        <IconClickFallbackImages program=\"AdChoices\">
          <IconClickFallbackImage height=\"150\" width=\"400\">
            <AltText>Alt icon fallback</AltText>
            <StaticResource creativeType=\"image/png\"><![CDATA[https://storage.googleapis.com/interactive-media-ads/images/wta_dialog.png?size=1x]]>
          </StaticResource>
        </IconClickFallbackImage>
        <IconClickFallbackImage height=\"300\" width=\"800\">
          <AltText>Alt icon fallback</AltText>
          <StaticResource creativeType=\"image/png\"><![CDATA[https://storage.googleapis.com/interactive-media-ads/images/wta_dialog.png?size=2x]]>
        </StaticResource>
        </IconClickFallbackImage>
        </IconClickFallbackImages>\"",
        "type": "IconClickFallbackImages"
      }
    ],
    "mediaFiles": {
      "mediaFilesList": [],
      "mezzanine": ""
    },
    "skipOffset": "00:00:03",
    "startTime": "PT0S",
    "startTimeInSeconds": 0,
    "trackingEvents": [
      {
        "beaconUrls": [
          "https://example.com/view"
        ],
        "duration": "PT10S",
        "durationInSeconds": 10,
        "eventId": "0",

```

```

        "eventProgramDateTime": null,
        "eventType": "impression",
        "startTime": "PT0S",
        "startTimeInSeconds": 0
      }
    ],
    "vastAdId": "123456"
  }
],
"availId": "0",
"availProgramDateTime": null,
"duration": "PT10S",
"durationInSeconds": 10,
"meta": null,
"nonLinearAdsList": [],
"startTime": "PT0S",
"startTimeInSeconds": 0
}
],
"dashAvailabilityStartTime": null,
"hlsAnchorMediaSequenceNumber": null,
"nextToken": "UFQxMFNfMjAyMy0wNy0wNlQyMDo0MT0xNy45NDE4MDM0NDhaXzE%3D",
"nonLinearAvails": []
}

```

클라이언트 측 비컨

클라이언트 측 추적 `startTimeInSeconds` 요소를 사용하면 MediaTailor를 사용하여 비컨 타이밍을 지원할 수 있습니다.

다음 JSON 응답은 노출, 시작, 사분위수 및 완료와 같은 주요 비컨 유형을 보여줍니다.

Note

대화형 광고국(IAB) 비디오 노출 측정 지침에는 노출 시 광고 콘텐츠가 클라이언트 측에 로드되고 최소한 플레이어에 렌더링되기 시작하는 시간이 필요하다고 명시되어 있습니다. 자세한 내용은 IAB 웹 사이트의 [Digital Video Ad Serving Template\(VAST\)](#)을 참조하세요.

```

{
  "avails": [
    {

```

```
"ads": [
  {
    "adId": "8104385",
    "duration": "PT15.100000078S",
    "durationInSeconds": 15.1,
    "startTime": "PT17.817798612S",
    "startTimeInSeconds": 17.817,
    "trackingEvents": [
      {
        "beaconUrls": [
          "http://exampleleadserver.com/tracking?event=impression"
        ],
        "duration": "PT15.100000078S",
        "durationInSeconds": 15.1,
        "eventId": "8104385",
        "eventType": "impression",
        "startTime": "PT17.817798612S",
        "startTimeInSeconds": 17.817
      },
      {
        "beaconUrls": [
          "http://exampleleadserver.com/tracking?event=start"
        ],
        "duration": "PT0S",
        "durationInSeconds": 0.0,
        "eventId": "8104385",
        "eventType": "start",
        "startTime": "PT17.817798612S",
        "startTimeInSeconds": 17.817
      },
      {
        "beaconUrls": [
          "http://exampleleadserver.com/tracking?event=firstQuartile"
        ],
        "duration": "PT0S",
        "durationInSeconds": 0.0,
        "eventId": "8104386",
        "eventType": "firstQuartile",
        "startTime": "PT21.592798631S",
        "startTimeInSeconds": 21.592
      },
      {
        "beaconUrls": [
          "http://exampleleadserver.com/tracking?event=midpoint"
        ]
      }
    ]
  }
]
```

```

    ],
    "duration": "PT0S",
    "durationInSeconds": 0.0,
    "eventId": "8104387",
    "eventType": "midpoint",
    "startTime": "PT25.367798651S",
    "startTimeInSeconds": 25.367
  },
  {
    "beaconUrls": [
      "http://exampleleadserver.com/tracking?event=thirdQuartile"
    ],
    "duration": "PT0S",
    "durationInSeconds": 0.0,
    "eventId": "8104388",
    "eventType": "thirdQuartile",
    "startTime": "PT29.14279867S",
    "startTimeInSeconds": 29.142
  },
  {
    "beaconUrls": [
      "http://exampleleadserver.com/tracking?event=complete"
    ],
    "duration": "PT0S",
    "durationInSeconds": 0.0,
    "eventId": "8104390",
    "eventType": "complete",
    "startTime": "PT32.91779869S",
    "startTimeInSeconds": 32.917
  }
]
}
],
"availId": "8104385",
"duration": "PT15.100000078S",
"durationInSeconds": 15.1,
"startTime": "PT17.817798612S",
"startTimeInSeconds": 17.817
}
]
}

```

서버 측 광고 비컨이 있는 하이브리드 모드

MediaTailor는 세션 추적을 위한 하이브리드 모드를 지원합니다. 이 모드에서 서비스는 재생 관련 광고 추적 이벤트를 내보내지만 세션에 대해 전체 클라이언트 측 추적 페이로드를 사용할 수 있도록 합니다.

재생 접두사를 사용하여 하이브리드 추적을 활성화하려면 플레이어에서 프로토콜에 따라 다음 형식 중 하나의 요청을 사용하여 새 MediaTailor 재생 세션을 초기화합니다.

Example: HLS 형식

```
POST master.m3u8
{
  "adsParams": {
    "deviceType": "ipad"
  },
  "reportingMode": "server"
}
```

Example: DASH 형식

```
POST manifest.mpd
{
  "adsParams": {
    "deviceType": "ipad"
  },
  "reportingMode": "server"
}
```

MediaTailor는 하이브리드 모드에서 다음 추적 이벤트를 유지합니다.

- 노출
- 시작
- 첫 번째 사분위수
- Midpoint
- 3사분위수
- 완료
- breakStart (vmap)
- breakEnd (vmap)

클라이언트 측 광고 추적 통합

이 섹션에서는 MediaTailor와 다양한 클라이언트 측 광고 추적 서버 간의 통합에 대해 설명합니다.

주제

- [Open 측정 SDK](#)
- [Datazoom 프리 플레이어 SDKs](#)
- [Roku 광고 프레임워크\(RAF\)](#)
- [TheoPlayer](#)
- [MediaTailor SDK](#)

Open 측정 SDK

Interactive Advertising Bureau(IAB) Open Measurement SDK(OM SDK)는 웹 비디오 및 네이티브 앱 환경에 제공되는 광고에 대한 타사 가시성 및 확인 측정을 용이하게 합니다.

이전 VAST 버전 3 문서의 경우 확장 유형이 인 확장 노드로 확인 코드를 로드해야 합니다 AdVerifications. 확장 노드의 루트는 VAST 4.1 요소와 스키마가 동일한 AdVerifications 노드입니다.

OM SDK를 더 쉽게 채택할 수 있도록 MediaTailor는 Datazoom과 협력하여 Open Measurement용으로 구성되고 확인된 무료 플레이어 SDKs를 제공합니다. 자세한 내용은 [Datazoom 프리 플레이어 SDKs](#) 단원을 참조하십시오.

Note

MediaTailor는 현재 VAST 버전 3만 지원합니다.

Example: 버전 4.1 이전 VAST 3의 확인 노드

```
...
<Extensions>
  <Extension type="AdVerifications">
    <AdVerifications>
      <Verification vendor="company.com-omid">
        <JavaScriptResource apiFramework="omid" browserOptional="true">
          <![CDATA[https://verification.com/omid_verification.js]]>
        </JavaScriptResource>
      </Verification>
    </AdVerifications>
  </Extension>
</Extensions>
```

```

        <TrackingEvents>
            <Tracking event="verificationNotExecuted">
                <![CDATA[https://verification.com/trackingurl]]>
            </Tracking>
        </TrackingEvents>
        <VerificationParameters>
            <![CDATA[verification params key/value pairs]]>
        </VerificationParameters>
    </Verification>
</AdVerifications>
</Extension>
</Extensions>

```

MediaTailor는 <Extensions> 노드에서 AdVerifications 데이터를 추출하여 클라이언트 측 추적 응답의 adVerifications 배열에 배치합니다.

Example: 클라이언트 측 추적 응답의 adVerifications 배열

```

{
  "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",

```

[illegible]

```

        }
      ],
      "vastAdId": ""
    }
  ],
  "availId": "3062770",
  "availProgramDateTime": "2023-08-23T16:25:40.914Z",
  "duration": "PT14.982S",
  "durationInSeconds": 14.982,
  "meta": null,
  "nonLinearAdsList": [],
  "startTime": "PT10.11S",
  "startTimeInSeconds": 10.11
}
],
"dashAvailabilityStartTime": null,
"hlsAnchorMediaSequenceNumber": null,
"nextToken": "UFQxMC4xMVNfMjAyMy0wOC0yM1QxNjoyNjoyNC4yNDYxMDIxOTBaXzE%3D",
"nonLinearAvails": []
}

```

Note

IAB Tech Lab과 협력하여 애플리케이션이 규정 준수를 보장하기 위해 매년 인증을 받도록 합니다.

OM SDK에 대한 자세한 내용은 IAB Tech Lab 웹 사이트의 [Open Measurement SDK](#)를 참조하세요.

Datazoom 프리 플레이어 SDKs

플레이어 SDKs를 쉽게 채택할 수 있도록 MediaTailor는 Datazoom과 협력하여 [Google Ad Manager와의 클라이언트 측 AWS Elemental MediaTailor 통합](#) 및 IAB Tech 로 구성 및 테스트된 무료 플레이어 SDKs를 제공합니다. [Open 측정 SDK](#).

Datazoom 플레이어 SDK는 다음 기능을 지원합니다.

- 라이브 및 VOD 재생 목록
- DASH 및 HLS 사양

- Bitmovin, exoplayer, Android 미디어 플레이어, Apple AVPlayer, Brightcove, Chromecast Receiver, Dash.js, hls.js, JWPlayer, Shaka player, THEO player, Video.js, Roku 등에 대한 플레이어 공급업체 지원
- 선택한 디바이스에서 사용 가능한 경우 IAB Tech Lab Open Measurement 인증
- 클릭스루 이벤트 처리
- 광고 카운트다운 타이머, 광고 오버레이 및 비선형 이벤트, 광고 중단 시작, 광고 중단 종료와 같은 광고 이벤트 디스패처
- 클라이언트 측 광고 비컨
- Google Programmatic Access Library(PAL) SDK, 선택적 구성 설정

또한 Datazoom은 플레이어 SDKs 지원하는 유료 분석 및 원격 측정 서비스를 제공합니다. 고객은 Datazoom 관리 콘솔에서 플레이어 SDK 원격 측정을 옵트인하고 제어할 수 있습니다. Datazoom 플레이어 SDKs에 액세스하고 부가가치 원격 측정 및 분석 서비스에 대해 자세히 알아보려면 [Datazoom 사이트의](#) 연락처 정보를 사용하세요.

Roku 광고 프레임워크(RAF)

Roku Ad Framework(RAF)는 Roku 플랫폼 전체에서 일관된 광고 환경을 유지합니다. 비디오 광고를 포함한 모든 채널은 RAF에 대한 Roku의 인증 요구 사항을 충족해야 합니다. 특히 앱은 항상 RAF를 통한 클라이언트 측 이벤트 실행을 사용해야 합니다. MediaTailor는 서버 측 광고 삽입(SSAI) 공급자로서 클라이언트 측 이벤트 실행을 지원합니다. RAFX SSAI 어댑터는 SSAI 매니페스트 서버 또는 스티처 및 RAF 모두에 인터페이스를 제공합니다. 이러한 인터페이스에는 다음이 포함됩니다.

- masterURL 응답을 구문 분석하고 playURL, AdURL 및 광고 메타데이터를 추출합니다.
- MediaTailor SSAI 광고 메타데이터를 RAF 사용 가능 광고 메타데이터로 변환하고 재생을 위해 RAF를 구성합니다.
- 스트림 이벤트 및 시간 지정된 메타데이터 관찰.
- 스트림 이벤트, 광고 메타데이터 및 발사 이벤트 픽셀을 정시에 일치시킵니다.
- MediaTailor SSAI 매니페스트 서버에서 요구하는 AdURL대로를 핑/폴링한 다음 RAF를 구문 분석하고 재구성합니다.

RAF용 SSAI 어댑터에 대한 자세한 내용은 [Roku 웹 사이트의 Roku 어댑터를 사용하여 서버 측 광고 삽입 구현을 참조하세요.](#)

TheoPlayer

MediaTailor와의 TheoPlayer 통합은 다음을 수행합니다.

- VOD 및 라이브 워크플로 모두에 대해 HLS 및 DASH에 대한 MediaTailor 클라이언트 측 이벤트를 추적하는 기능을 제공합니다.
- 선형 광고에 대해서만 추적 비컨 전송을 지원합니다.
- 광고 중에 탐색을 비활성화합니다. 그러나 사용자가 광고 시간을 지나려고 할 때 광고를 재생하기 위한 로직은 없습니다.

TheoPlayer의 SSAI에 대한 자세한 내용과 MediaTailor용 웹, Android, iOS 및 tvOS SDKs를 검토하려면 TheoPlayer 웹 사이트의 [MediaTailor](#)를 참조하세요.

MediaTailor SDK

AWS Elemental은 JavaScript 기반 소프트웨어 개발 키트(SDK)를 유지 관리합니다. AWS Elemental은 묵시적 보증 없이 SDK를 있는 그대로 제공합니다. SDK를 참조 데모로 사용하여 MediaTailor를 사용하여에 대한 온보딩을 간소화합니다. SDK는 MediaTailor 클라이언트 측 추적 API와 상호 작용하는 방법을 보여줍니다. SDK는 HTML5-based 플레이어에 대한 클라이언트 측 광고 추적 및 보고를 구현합니다. SDK는 MediaTailor 클라이언트 측 보고 세션을 초기화한 다음 주기적으로 광고 추적 정보를 요청합니다. 재생 중에 SDK는 새 광고 이벤트가 감지되면 광고 추적 이벤트를 내보냅니다.

MediaTailor SDK는 다음 기능을 지원합니다.

- 라이브 및 VOD 재생 목록
- DASH 및 HLS 사양
- 클릭스루 이벤트 처리
- Ad-event 디스패처
- 사용자 지정 이벤트 후크
- 클라이언트 측 광고 비컨. 광고 비컨 전송에 대한 자세한 내용은 섹션을 참조하세요 [클라이언트 측 비컨](#).

Note

MediaTailor용 JavaScript SDK 샘플을 받으려면 AWS 지원 티켓을 제출하세요. 패키지 및 해당 파일에 대한 다운로드 링크를 받게 됩니다.

GetTracking을 사용하여 광고 비컨 페이징

GetTracking 엔드포인트를 사용하여 플레이어에게 반환되는 광고 수를 좁힙니다. 예를 들어 매니페스트 기간이 넓어 시간이 많이 걸리는 경우 반환된 광고 비컨 수가 플레이어 성능에 영향을 미칠 수 있습니다.

GetTracking는 반환된 비컨 목록을 페이징하여 반환된 비컨 수를 좁히는 데 사용할 수 있는 NextToken 값을 반환합니다. NextToken 값을 순환하여 광고 비컨 StartTimeInSeconds 필드의 원하는 값을 찾을 수 있습니다.

- 에 대한 첫 번째 호출에서 GetTracking매니페스트 창에 속하는 각의 NextToken 및 값을 포함하여 가능한 모든 광고가 반환됩니다.
- GetTracking 요청에가 포함되지 않은 경우 매니페스트 창의 NextToken모든 광고가 반환됩니다.
- GetTracking 요청에가 포함되어 NextToken 있지만 반환할 새 비컨이 없는 경우 MediaTailor는 원래 요청에서 전송NextToken한 값과 동일한 값을 반환합니다.
- 광고에 해당하는 비컨이 더 이상 없는 경우는 해당 응답에서 광고를 GetTracking 제거합니다.
- 의 토큰은 24시간 후에 GetTracking 만료됩니다. NextToken 값이 24시간보다 오래된 경우에 대한 다음 호출은 null 값을 GetTracking 반환합니다NextToken.

플레이어에서 GetTracking의 일반화된 호출 시퀀스

클라이언트 플레이어의 GetTracking 요청은 토큰과 관련된 NextToken 및 광고와 비컨이 포함된 요청 본문이 있는 POST입니다.

```
https://YouMediaTailorUrl/v1/tracking
{
    "NextToken": "value"
    .
    .
    .
}
```

GetTracking와 함께 사용하는 일반적인 순서는 다음과 NextToken 같습니다.

1. 를 처음 호출합니다GetTracking.

- 모든 광고 및 비컨과 후속 호출을 NextToken 위한 첫 번째가 반환됩니다.
2. 값이 null이면 MediaTailorNextToken는 모든 광고 비컨을 반환합니다.
 3. 이 만료되면 MediaTailorNextToken는 HTTP 반환 코드 400 오류 메시지를 반환합니다.

를 새로 호출GetTracking하여 유효한 NextToken를 검색합니다.

4. 전체 응답을 스캔하여 원하는 범위에 있는 광고 비컨StartTimeInSeconds의를 찾습니다.
5. 원하는와 NextToken 연결된 값을 GetTracking 사용하여 새로 호출합니다StartTimeInSeconds.
6. 필요한 경우 재생하려는 광고를 정확히 찾을 때까지 반환된 광고를 다시 순환합니다.

확장 예제

이 예제에서는 GetTracking의를 사용하여 플레이어NextToken에게 반환되는 광고 비컨 수를 제한하는 방법을 보여줍니다.

MediaTailor는 GetTracking 요청을 받습니다. 응답에는 ID가 9935407인 광고와 StartTimeInSeconds 값이 52.286 및 48.332초인 비컨 2개가 포함됩니다.

MediaTailor는 다음과 NextToken 같이를 사용하여 JSON 응답을 전송합니다.

```
{
  "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
}
]
}

```

다음 GetTracking 요청에서 MediaTailor는

JF57ITE48t1441mv7TmLKuZLroxDzflsp6BiSNL1IJmzPVMDN0lqrBYycgMbKEb NextToken 값으로 응답합니다.

MediaTailorStartTimeInSeconds는 이전 호출NextToken의에 설정된와 일치하는 광고 및 비컨으로 응답합니다.

이제 응답에 ID가 9235407인 이전 광고 외에 ID가 9935407인 다른 광고가 포함되어 있다고 가정합니다. 광고 ID 9235407의 비컨에는 StartTimeInSeconds 132.41 및 70.339가 있습니다.

MediaTailor는 세션의 모든 비컨을 반복하여 ID가 9235407인 광고의 비컨 3 및 비컨 4인 52.286초보다 StartTimeInSeconds 큰 비컨을 선택합니다.

```
{
  "NextToken": "ZkfknvbfsgdfbsDFRdffg12EdffecFRvhjyjfhdhfnjtsG5SDGN
  "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"
              ],
```

```

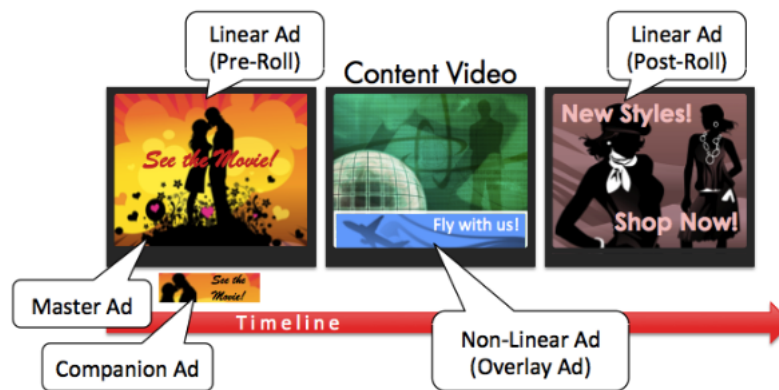
        "duration": "PT0S",
        "durationInSeconds": 0,
        "eventId": "8935412",
        "eventType": "secondQuartile",
        "startTime": "PT2M12.41S",
        "StartTimeInSeconds": 132.41
    },
    ],
    "vastAdId": ""
},
],
"startTime": "PT36.47S",
"StartTimeInSeconds": 36.47
}
]
}

```

오버레이 광고

미드를 광고로 시청 환경을 중단하지 않고 수익 창출을 늘리려는 라이브 스트리밍 워크플로의 경우 현재 AWS Elemental MediaTailor 통합을 활용하여 클라이언트 측에서 렌더링되는 광고 형식을 안내할 수 있습니다. 이러한 유형의 광고를 오버레이 광고라고 합니다. 오버레이 광고는 'L-대역 광고', '비선형 비디오 광고', 'picture-in-picture 광고', '모션 오버레이', '콘텐츠 내 광고' 또는 '프레임 광고'의 형태로 나타나는 비선형 비디오 광고입니다.

MediaTailor는 분할 유형이 오버레이 광고 삽입 기회에 대한 대역 내 신호id=0x38인 SCTE-35 마커를 감지합니다. SCTE-35 마커를 사용하면 MediaTailor가 Ad Decision Server(ADS)에 요청을 보낸 다음 VAST 응답에서 비선형 광고 페이로드로 응답합니다. MediaTailor는 오버레이 광고 삽입을 지원하기 위해 VAST 응답을 구문 분석합니다. MediaTailor는 선형 광고를 스티칭하지 않고 플레이어에게 재생할 수 있는 비선형 오버레이 광고가 있음을 알립니다. 이 신호를 통해 플레이어는 클라이언트 측 추적 엔드포인트에서 재생할 비선형 광고를 가져오고 상호 연관시킬 수 있습니다. 그러면 플레이어가 해당 광고와 관련된 디스플레이, 보고 및 기타 작업을 처리합니다. 예를 들어 플레이어의 개발자는 오버레이 광고 형식을 지원하는 공급업체의 디바이스 SDK를 사용할 수 있습니다. 클라이언트 측 추적 통합에 대한 자세한 내용은 [섹션을 참조하세요](#) [클라이언트 측 광고 추적 통합](#).



주제

- [MediaTailor에서 오버레이 광고를 사용하기 위한 사전 조건](#)
- [MediaTailor에서 오버레이 광고 사용 시작하기](#)
- [MediaTailor에서 오버레이 광고에 대한 로깅 및 지표](#)
- [MediaTailor의 오버레이 광고에 대한 결제](#)

MediaTailor에서 오버레이 광고를 사용하기 위한 사전 조건

MediaTailor에서 오버레이 광고를 사용할 때 다음 사전 조건이 적용됩니다.

- 워크플로는 온디맨드 비디오(VOD)가 아닌 라이브여야 합니다.
- Ad Decision Server(ADS) 응답은 VAST 응답에서 비선형 광고만 반환하도록 구성해야 합니다. MediaTailor는 광고 스티칭을 위해 선형 광고를 무시합니다.
- 매니페스트는 오버레이 광고 기능을 호출id=0x38하려면 분할 유형이 있는 SCTE-35 시간 신호 메시지를 사용해야 합니다.
- 스트리밍 공급자는 클라이언트 디바이스 애플리케이션을 제어하고 MediaTailor 클라이언트 측 추적 API와 통합되어야 합니다.

MediaTailor에서 오버레이 광고 사용 시작하기

이 섹션에서는 MediaTailor의 오버레이 광고 기능을 사용하여 시작하는 방법을 설명합니다. SCTE-35 신호 전송을 설정하고, Ad Decision Server(ADS) 응답을 구성하고, 세션 수준 제어를 설정합니다.

주제

- [오버레이 광고 활성화](#)
- [클라이언트 측 메타데이터를 사용하여 오버레이 광고 추적](#)

오버레이 광고 활성화

오버레이 광고에 대한 MediaTailor 지원은 기본적으로 활성화되어 있습니다. 매니페스트의 특정 SCTE-35 광고 마커 유형은 오버레이 광고 삽입을 트리거합니다. 일부 플레이어는 오버레이 광고의 클라이언트 측 렌더링을 지원하지 않을 수 있으므로 세션 수준에서 기능을 비활성화할 수 있습니다.

HLS 또는 DASH 재생 접두사를 사용하여 오버레이 광고 지원을 비활성화하려면:

- 플레이어에서 프로토콜에 따라 다음 형식 중 하나로 요청을 사용하여 새 MediaTailor 재생 세션을 초기화합니다.

- 예: HLS 형식

```
GET mediatailorURL/v1/master/hashed-account-id/origin-id/asset-id?
aws.overlayAvails=off
```

- 예: DASH 형식

```
GET mediatailorURL/v1/master/hashed-account-id/origin-id/asset-id?
aws.overlayAvails=off
```

세션 초기화 접두사를 사용하여 오버레이 광고 지원을 비활성화하려면:

- 플레이어에서 MediaTailor에 대한 세션 초기화 요청에 대한 JSON 메시지 본문을 구성합니다.
 - 광고 오버레이 지원을 비활성화하려면 `overlays` 객체를 값이 `in`의 최상위 키로 추가합니다 `off`. 기본 `overlays` 값은 `on`입니다.
 - (선택 사항) MediaTailor가 `adsParams` 객체 내의 ADS로 전달하는 파라미터를 제공합니다. 이러한 파라미터는 MediaTailor 구성의 ADS 템플릿 URL에 있는 `[player_params.param]` 설정에 해당합니다.

Example HLS:

```
POST master.m3u8
{
```

```

    "adsParams": {
      "deviceType": "ipad"
    },
    "overlayAvails": "off"
  }

```

Example DASH:

```

POST manifest.mpd
{
  "adsParams": {
    "deviceType": "androidmobile"
  },
  "overlayAvails": "off"
}

```

매니페스트 신호

MediaTailor 트리거 오버레이 광고는 매니페스트에 특정 SCTE-35 마커가 표시되면 지원합니다. 필수 신호는 공급자 오버레이 광고 시작 신호인 스플라이스 명령 유형 6 또는 시간 신호입니다. 이 신호의 세분화 유형 ID는 0x38입니다.

다음 예제는 JSON 객체의 0x38 SCTE-35 마커를 보여줍니다.

```

{
  "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
}

```

다음 예제는 바이너리(베이스 32/16진수) 값으로 표시되는 SCTE-35 신호를 보여줍니다.

```
0xfc303500000000000000ff00506fe001b78c8001f021d435545490970d4717fd0000dbba009076f7665726c617
```

다음 예제는 HLS 매니페스트와 DASH 매니페스트 모두에서 SCTE-35 마커를 보여줍니다.

Example: HLS 매니페스트

#EXTM3U

```
#EXT-X-VERSION:6
#EXT-X-TARGETDURATION:7
#EXT-X-MEDIA-SEQUENCE:419
#EXT-X-DISCONTINUITY-SEQUENCE:3
#EXT-X-PROGRAM-DATE-TIME:2023-08-15T04:30:09.231Z
#EXTINF:6.02,
https://aws.cloudfront.net/media/asset1/index1_00007.ts
#EXT-X-DISCONTINUITY
#EXT-X-KEY:METHOD=NONE
#EXT-X-PROGRAM-DATE-TIME:2023-08-15T04:30:15.251Z
#EXTINF:6.0,
https://aws.cloudfront.net/media/asset1/index1_00001.ts
#EXT-X-PROGRAM-DATE-TIME:2023-08-15T04:30:21.251Z
#EXTINF:4.0,
https://aws.cloudfront.net/media/asset1/index1_00002.ts
#EXT-X-DISCONTINUITY
#EXT-X-DATERANGE:ID="1692073825251-30-1",START-
DATE="2023-08-15T04:30:25.251Z",DURATION=10.0,PLANNED-DURATION=10.0,SCTE35-
OUT=0xfc30350000000000000000ffff00506fe001b78c8001f021d435545490970d4717fdf00000dbba009076f7665726
#EXT-X-PROGRAM-DATE-TIME:2023-08-15T04:30:25.251Z
#EXTINF:2.0,
https://aws.cloudfront.net/media/asset1/index1_00003.ts
#EXT-X-PROGRAM-DATE-TIME:2023-08-15T04:30:27.251Z
#EXTINF:6.0,
https://aws.cloudfront.net/media/asset1/index1_00004.ts
#EXT-X-PROGRAM-DATE-TIME:2023-08-15T04:30:33.251Z
#EXTINF:2.0,
https://aws.cloudfront.net/media/asset1/index1_00005.ts
#EXT-X-DISCONTINUITY
#EXT-X-PROGRAM-DATE-TIME:2023-08-15T04:30:35.251Z
#EXTINF:4.0,
https://aws.cloudfront.net/media/asset1/index1_00006.ts
#EXT-X-PROGRAM-DATE-TIME:2023-08-15T04:30:39.251Z
#EXTINF:6.02,
https://aws.cloudfront.net/media/asset1/index1_00007.ts
```

Example: DASH 매니페스트

```
<?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>
```

Ad Decision Server(ADS) 응답

ADS 응답에는 하나의 유효한 추적 이벤트가 포함되어야 합니다. 최소한 추적 이벤트는 Impression 추적 이벤트일 수 있습니다. 추적 이벤트에는 하나 이상의 NonLinear 광고가 포함되어야 합니다. 이 광고는 정적, HTML 또는 iFrame 리소스의 형태를 취하는 오버레이 광고입니다.

```
<vmap AdBreak breaktype="linear" breakId="csoverlay"
```

VAST 응답이 nonlinearbreakType가 인 VMAP인 경우 가용 메타데이터는 nonLinearAvails 루트 객체 내에 있습니다. VAST 응답이 breakType가 인 VMAPlinear이거나 VMAP가 없는 일반 VAST 응답인 경우 가용 메타데이터는 avails 루트 객체 내에 있습니다.

다음 VAST 응답은 breakType 값이 인 래핑된 VMAP 응답입니다linear.

래핑된 VMAP 응답 외에도 MediaTailor는 breakType 값이 인 래핑된 VMAP 응답nonlinear과 일반 VAST 응답도 지원합니다.

```
<?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: MediaTailor에 대한 DASH 매니페스트 소스

```

<?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: 광고 ID 장식을 포함하는 MediaTailor 맞춤형 DASH 매니페스트

```
<?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:smp: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>
```

클라이언트 측 메타데이터를 사용하여 오버레이 광고 추적

MediaTailor는 오버레이 광고를 사용 가능 nonLinearAdsList 구간 의에 배치합니다. MediaTailor 클라이언트 측 추적 API에는 avails 및 라는 두 개의 루트 객체가 있습니다nonLinearAvails. VAST 응답이 nonlinearbreakType가 인 VMAP인 경우 가용 메타데이터는 nonLinearAvails 루트 객체 내에 있습니다. VAST 응답이 breakType가 인 VMAPlinear이거나 VMAP가 없는 일반 VAST 응답인 경우 가용 메타데이터는 avails 루트 객체 내에 있습니다.

클라이언트 측 추적에 대한 자세한 내용은 섹션을 참조하세요 [클라이언트 측 광고 추적](#).

다음 예제에서는 breakType 값이 인 일반 VAST 응답 또는 VMAP 응답을 보여줍니다linear.

```
{
  "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": []
}

```

다음 예제에서는 breakType 값이 인 일반 VMAP 응답을 보여줍니다 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
}
]
}

```

MediaTailor에서 오버레이 광고에 대한 로깅 및 지표

이 섹션에서는 MediaTailor의 오버레이 광고에 대한 로깅 및 지표를 설명합니다. 로깅 설정에 대한 자세한 내용은 섹션을 참조하세요 [AWS Elemental MediaTailor 리소스 모니터링 및 태그 지정](#).

주제

- [CloudWatch 로그](#)

- [CloudWatch 지표](#)

CloudWatch 로그

CloudWatch는 오버레이 광고에 대한 다음 로그 정보를 수집합니다.

- VAST_RESPONSE - 비선형 광고 목록에 대한 정보를 표시합니다.
- FILLED_PROVIDER_OVERLAY - 비선형 광고에 대한 정보를 표시합니다.

Note

는 ADS의 원래 응답을 보여주는 선택적 이벤트 RAW_ADS_RESPONSE입니다. 이 이벤트를 사용하면 스테이징 및 테스트 환경에서 특히 유용합니다. 구성 또는 계정에서이 이벤트를 활성화하려면 AWS Support에 티켓을 제출합니다.

CloudWatch 지표

MediaTailor는 오버레이 광고 지표를 다른 ADS 지표와 별도로 수집합니다. MediaTailor는 ADS에서 광고를 성공적으로 가져온 후 이러한 지표를 수집합니다. 지표를 수집하기 위해 GetTracking API를 폴링할 필요가 없습니다.

다음 표에서는 오버레이 광고에 대한 CloudWatch 지표를 설명합니다.

지표	설명
AdDecisionServer.OverlayAds	지정한 CloudWatch 기간 내에 ADS 응답에 포함된 오버레이 광고 수입니다.
AdDecisionServer.OverlayErrors	지정한 CloudWatch 기간 내에 MediaTailor가 ADS로부터 수신한 비 HTTP 200 상태 코드 응답, 빈 응답 및 시간 초과 응답 수입니다.
AdDecisionServer.OverlayFilled	하나 이상의 오버레이 광고로 성공적으로 채워진 가용 시간 수: • 1 - 유효한 광고가 하나 이상 있습니다.

지표	설명
	0 - MediaTailor에서 오버레이 광고를 가져오지 못했거나 다른 오류가 발생했습니다. SampleCount 는 채워진 가용 수를 추적합니다. Sum는 성공적으로 채워진 오버레이 가용 수를 추적합니다.
AdDecisionServer.OverlayMinSuggestedDuration	지정한 CloudWatch 기간 내에 MediaTailor 가 ADS에서 수신한 모든 광고의 minSuggestedDuration 지속 시간 합계입니다. minSuggestedDuration 이 지정되지 않은 경우 표시된 기간은 계획된 기간입니다.
AdDecisionServer.OverlayLatency	MediaTailor가 ADS에 보내는 요청에 대한 밀리초 단위의 응답 시간입니다.
AdDecisionServer.OverlayTimeouts	지정한 CloudWatch 기간에 ADS에 대한 시간 초과 요청 수입니다.
AdsBilled	청구된 광고에 대한 자세한 내용은 섹션을 참조하세요 MediaTailor의 오버레이 광고에 대한 결제 .
Avail.*	MediaTailor는 오버레이 광고를 계획하지 않으므로 CloudWatch는 Avail.X 지표를 표시하지 않습니다.
SkippedReason.*	MediaTailor는 오버레이 광고를 계획하지 않으므로 CloudWatch는 SkippedReason.X 지표를 표시하지 않습니다.

MediaTailor의 오버레이 광고에 대한 결제

MediaTailor는 ADS 응답의 비선형 광고 수를 기준으로 고객에게 요금을 청구합니다. 이 숫자에는 휴식 시간을 지나 연장된 비선형 광고가 포함됩니다. MediaTailor는 가용 구간을 채운 후 채운 광고에 대해 요금을 청구합니다.

프리페치 워크플로의 경우 MediaTailor는 프리페치를 검색할 때 광고에 대해 요금을 청구하지 않고 해당 세션의 소비 기간에 호환 가능한 광고 가용성이 표시되는 경우 요금을 청구합니다.

추가 결제 정보는 섹션을 참조하세요 <https://aws.amazon.com/mediatailor/pricing/>.

광고 ID 장식

AWS Elemental MediaTailor 는 콘텐츠에서 광고 시간으로 전환할 때 서버 측 광고 스티칭을 수행합니다. MediaTailor는 스티칭된 광고와 연결된 메타데이터로 매니페스트를 조건을 지정할 수 있습니다. 이렇게 하면 다음과 같은 이점이 있습니다.

Important

헤더를 채우는 필드는 캐시 가능한 매니페스트가 처음 작성될 때가 아니라 광고 재생 중에 자산 목록을 가져올 때 X-AD-CREATIVE-SIGNALING 만 알려져 있기 때문에 Ad-ID 장식은 서버 가이드 광고 삽입(SGAI) 방법과 호환되지 않습니다.

- 비디오 시작 시간(VST) 개선
- MediaTailor는 서버 측 광고 삽입 및 서버 안내 광고 삽입의 하이브리드 모델을 지원할 수 있습니다.
- 서버 측 세션은 광고 위치 마커를 사용하여 재생 타임라인을 구축할 수 있습니다.
- MediaTailor API를 사용하여 재생 타임라인을 이미 빌드한 클라이언트 측 세션의 경우 세션이 타임라인을 빌드하기 위해 추적 API를 호출하는 데 의존하지 않으므로 세션 VST가 개선됩니다.
- 서버 측 광고 삽입과 장면에 표시되는 클라이언트 측 렌더링 광고에 MediaTailor를 활용할 수 있습니다. 이렇게 하면 플레이어의 소프트웨어 개발 키트(SDK)가 클라이언트 측 광고를 위해 광고 제공 엔터티를 직접 호출하기 위해 별도의 통합을 가질 필요가 없습니다. MediaTailor는 매니페스트와 클라이언트 측 추적 API를 통해 광고를 판매할 수 있습니다.

각 크리에이티브 광고 자산을 고유 식별자와 연결하는 표준이 있습니다. 이 연결을 통해 광고주, 기관, 공급업체 및 게시자는 독립적인 워크플로에서 창의적인 광고 자산을 연결할 수 있습니다. 스트림의 지

표와 모니터링이 계속 개선되고 더 많은 배포자가 서버 기반 삽입 아키텍처를 활용함에 따라 개인화된 매니페스트 내에서와 같이 인터리브/스티치 프레젠테이션 내에서 개별 크리에이티브 자산에 할당된 식별자를 정확하게 전달해야 합니다.

주제

- [세션에 대한 광고 ID 신호 활성화](#)
- [매니페스트 및 광고 메타데이터 삽입](#)
- [Ad Decision Server\(ADS\) 상호 작용](#)
- [클라이언트 측 추적 API](#)

세션에 대한 광고 ID 신호 활성화

광고 ID 신호 기능은 세션 초기화 중에 활성화해야 합니다. 기능을 활성화하는 프로세스는 HLS/DASH 재생 접두사(암시적 세션 초기화)를 사용하여 세션을 생성하는 것과 세션 초기화 접두사(명시적 세션 초기화)를 사용하는 것과 다릅니다.

HLS/DASH 재생 접두사를 사용하여 세션에 대한 광고 ID를 활성화하려면

- 플레이어에서 프로토콜에 따라 다음 형식 중 하나의 요청을 사용하여 새 MediaTailor 재생 세션을 초기화합니다.
- 예: HLS 형식

```
GET <mediatailorURL>/v1/master/<hashed-account-id>/<origin-id>/<asset-id>?
aws.adSignalingEnabled=true
```

- 예: DASH 형식

```
GET <mediatailorURL>/v1/dash/<hashed-account-id>/<origin-id>/<asset-id>?
aws.adSignalingEnabled=true
```

세션 초기화 접두사를 사용하여 세션에 대한 광고 ID를 활성화하려면

- 플레이어에서 MediaTailor에 대한 세션 초기화 요청에 대한 JSON 메시지 본문을 구성합니다.
- adsParams 객체 내에서 MediaTailor가 ADS에 전달해야 하는 파라미터를 제공합니다. 이러한 파라미터는 MediaTailor 구성의 ADS 템플릿 URL에 있는 [player_params.param] 설정에 해당합니다.

- 광고 ID 신호를 활성화하려면 `adSignaling` 객체를 최상위 객체로 추가하고 내부에 `enabled` 및 `값` 이라는 파라미터를 추가합니다 `true`. 기본 `adSignaling` 값은 `disabled`.
- 예: HLS 형식

```
POST master.m3u8
{
  "adsParams": {
    "deviceType": "ipad"
  },
  "adSignaling": {
    "enabled": "true"
  },
  "reportingMode": "client"
}
```

- 예: DASH 형식

```
POST manifest.mpd
{
  "adsParams": {
    "deviceType": "ipad"
  },
  "adSignaling": {
    "enabled": "true"
  },
  "reportingMode": "client"
}
```

매니페스트 및 광고 메타데이터 삽입

광고 스티칭 프로세스 중에 MediaTailor는 스티칭되는 각 크리에이티브와 연결된 고유 ID를 매니페스트에 추가합니다. MediaTailor는 VAST 응답에서 해당 크리에이티브의 `id` 속성 값에서 크리에이티브의 고유 ID를 가져옵니다. 크리에이티브에 ID 속성 값이 없는 경우 MediaTailor는 빈 값()을 게시합니다 `id=""`.

MediaTailor는 매니페스트 내 메타데이터 신호를 사용하여 광고 크리에이티브 메타데이터를 위한 클라이언트 추적 API와 전체 타임라인 내의 타이밍/위치 지정 간의 종속성을 분리합니다. 이 연결 해제는

재생을 초기화하기 전에 플레이어의 사용자 인터페이스(UI)가 타임라인의 광고 중단 위치를 렌더링하는 재생 지연 시간을 줄입니다(특히 VOD 시나리오에서).

추가된 메타데이터의 형식은 다음과 같습니다.

- HLS 매니페스트의 경우 추가된 메타데이터는 가용 기간 동안 각 광고에 대한 DATERANGE 태그의 형태를 취합니다.
- DASH 매니페스트의 경우 추가된 메타데이터는 각 광고 기간 내에 Event 요소의 형태를 취합니다.

다음 JSON 메시지 본문은 VAST 응답의 예를 보여줍니다.

```
{
  "version": 1,
  "identifiers": [
    {
      "scheme": "urn:smp:ul:060E2B34.01040101.01200900.00000000",
      "value": "creativeId",
      "ad_position": "adId",
      "ad_type": "adType",
      "tracking_uri": "trackingUri",
      "custom_vast_data": "customVastData"
    }
  ]
}
```

이전 예제에서:

- *creativeId*는 광고에 대한 Creative 요소의 Id 속성 값입니다.
- *adId*는 광고의 시작과 연결된 HLS 시퀀스 번호 또는 광고의 DASH 기간 ID입니다.
- *adType*은 VAST 응답을 overlay기반으로 avail 또는 입니다.
- *trackingUri*는 MediaTailor 세션의 상대 추적 엔드포인트로, 형식입니다. `.../.../.../.../tracking/hashed-account-id/origin-id/session-id`
- *customVastData*는 MediaTailor가 creative_signaling VAST 확장에서 추출하는 값입니다. MediaTailor는 CDATA 노드가 있는 경우 해당 노드의 콘텐츠를 사용합니다. 자세한 내용과 샘플 VAST 응답은 [Ad Decision Server\(ADS\) 상호 작용](#) 섹션을 참조하세요.

광고 메타데이터를 사용하여 HLS 매니페스트 개인화

라이브 HLS 스트림의 경우 MediaTailor는 스트림에 PROGRAM-DATE-TIME 태그가 포함된 경우에만 매니페스트 기간당 한 번 이상 메타데이터를 추가합니다. 온디맨드 비디오(VOD) 스트림의 경우 MediaTailor는 개인화된 매니페스트에서 하나 이상의 세그먼트PROGRAM-DATE-TIME에를 추가합니다. 여기서 각 VOD 자산의 시작 시간은 에포크 0()입니다1970-01-01T00:00:00Z. 오리진 매니페스트에 기존 PROGRAM-DATE-TIME 콘텐츠가 있는 경우 MediaTailor는 해당 콘텐츠를 보존합니다.

MediaTailor는 광고 결정 서버(ADS)에서 반환한 크리에이티브로 매니페스트를 개인화합니다. 각 광고에 대해 MediaTailor에는 광고 기간에 걸친 DATERANGE 태그도 포함되어 있습니다. DATERANGE 태그 형식은 SVA 기술 간행물 2023 버전의 [DASH 및 HLS의 광고 크리에이티브 신호](#) 섹션에 설명된 것과 유사합니다.

MediaTailorDATERANGE가 생성하는 에는 고유한 ID 값이 있습니다. 고유성을 보장하기 위해([SCTE-35를 EXT-X-DATERANGE로 매핑](#)에 지정된 지침에 따라) MediaTailor는 가용 구간의 첫 번째 광고 세그먼트 MEDIA-SEQUENCE 수를 가용 구간 내의 광고 시퀀스 번호와 결합합니다.

슬레이트가 활성화된 구성에서 과소 채워진 광고 시간의 경우 MediaTailor는 DISCONTINUITY 태그로 구분되지만 DATERANGE 메타데이터 없이 슬레이트 세그먼트를 가용 구간 끝에 추가합니다.

맞춤형 매니페스트에 연결된 각 광고에 대해 MediaTailor는 사용자 지정 DATERANGE 태그에 base64로 인코딩된 데이터로 표시되는 창의적인 메타데이터를 추가합니다.

Example선형 HLS 오리진(#EXT-X-CUE-OUT):

```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-TARGETDURATION:7
#EXT-X-MEDIA-SEQUENCE:398
#EXT-X-PROGRAM-DATE-TIME:2023-02-10T19:20:01.397Z
#EXTINF:6.006,
index_1_398.ts?m=1676054627
#EXTINF:5.873,
index_1_399.ts?m=1676054627
#EXT-OATCLS-SCTE35:/DA1AAAAAyiYAP/wFAUAAAACf+//jP197P4AUmNiAAEBAQAase4/gA==
#EXT-X-CUE-OUT:59.993
#EXTINF:6.139,
index_1_400.ts?m=1676054627
#EXT-X-CUE-OUT-CONT:ElapsedTime=6.139,Duration=59.993,SCTE35=/DA1AAAAAyiYAP/wFAUAAAACf+//jP197P4AUmNiAAEBAQAase4/gA==
#EXTINF:6.006,
index_1_401.ts?m=1676054627
```



```
#EXT-X-CUE-OUT-CONT:ElapsedTime=12.145,Duration=59.993,SCTE35=/DA1AAAAAyiYAP/wFAUAAAACf
+//jP197P4AUmNiAAEBAQAase4/gA==
#EXTINF:6.006,
index_1_402.ts?m=1676054627
#EXT-X-CUE-OUT-CONT:ElapsedTime=18.151,Duration=59.993,SCTE35=/DA1AAAAAyiYAP/wFAUAAAACf
+//jP197P4AUmNiAAEBAQAase4/gA==
#EXTINF:6.006,
index_1_403.ts?m=1676054627
#EXT-X-CUE-OUT-CONT:ElapsedTime=24.157,Duration=59.993,SCTE35=/DA1AAAAAyiYAP/wFAUAAAACf
+//jP197P4AUmNiAAEBAQAase4/gA==
#EXTINF:6.006,
index_1_404.ts?m=1676054627
#EXT-X-CUE-OUT-CONT:ElapsedTime=30.163,Duration=59.993,SCTE35=/DA1AAAAAyiYAP/wFAUAAAACf
+//jP197P4AUmNiAAEBAQAase4/gA==
#EXTINF:6.006,
index_1_405.ts?m=1676054627
#EXT-X-CUE-OUT-CONT:ElapsedTime=36.169,Duration=59.993,SCTE35=/DA1AAAAAyiYAP/wFAUAAAACf
+//jP197P4AUmNiAAEBAQAase4/gA==
#EXTINF:6.006,
index_1_406.ts?m=1676054627
#EXT-X-CUE-OUT-CONT:ElapsedTime=42.175,Duration=59.993,SCTE35=/DA1AAAAAyiYAP/wFAUAAAACf
+//jP197P4AUmNiAAEBAQAase4/gA==
#EXTINF:6.006,
index_1_407.ts?m=1676054627
#EXT-X-CUE-OUT-CONT:ElapsedTime=48.181,Duration=59.993,SCTE35=/DA1AAAAAyiYAP/wFAUAAAACf
+//jP197P4AUmNiAAEBAQAase4/gA==
#EXTINF:6.006,
index_1_408.ts?m=1676054627
#EXT-X-CUE-OUT-CONT:ElapsedTime=54.187,Duration=59.993,SCTE35=/DA1AAAAAyiYAP/wFAUAAAACf
+//jP197P4AUmNiAAEBAQAase4/gA==
#EXTINF:5.806,
index_1_409.ts?m=1676054627
#EXT-X-CUE-IN
#EXTINF:6.206,
index_1_410.ts?m=1676054627
#EXTINF:6.006,
index_1_411.ts?m=1676054627
#EXTINF:6.006,
index_1_412.ts?m=1676054627
```

Example선형 HLS 오리진(#EXT-X-DATERANGE):

```
#EXTM3U
```

```
#EXT-X-VERSION:3
#EXT-X-TARGETDURATION:7
#EXT-X-MEDIA-SEQUENCE:25
#EXT-X-PROGRAM-DATE-TIME:2023-02-10T19:19:53.389Z
#EXTINF:6.006,
index_1_25.ts?m=1676056675
#EXTINF:6.006,
index_1_26.ts?m=1676056675
#EXTINF:6.006,
index_1_27.ts?m=1676056675
#EXTINF:1.869,
index_1_28.ts?m=1676056675
#EXT-X-DATERANGE:ID="2",START-DATE="2023-02-10T19:20:13.276Z",PLANNED-
DURATION=59.993,SCTE35-
OUT=0xFC302500000003289800FFF01405000000027FEFFF8CF97DECFE00526362000101010000B1EE3F80
#EXTINF:6.139,
index_1_29.ts?m=1676056675
#EXTINF:6.006,
index_1_30.ts?m=1676056675
#EXTINF:6.006,
index_1_31.ts?m=1676056675
#EXTINF:6.006,
index_1_32.ts?m=1676056675
#EXTINF:6.006,
index_1_33.ts?m=1676056675
#EXTINF:6.006,
index_1_34.ts?m=1676056675
#EXTINF:6.006,
index_1_35.ts?m=1676056675
#EXTINF:6.006,
index_1_36.ts?m=1676056675
#EXTINF:6.006,
index_1_37.ts?m=1676056675
#EXTINF:5.806,
index_1_38.ts?m=1676056675
#EXT-X-DATERANGE:ID="2",START-DATE="2023-02-10T19:20:13.276Z",END-
DATE="2023-02-10T19:21:13.269Z",DURATION=59.993
#EXTINF:6.206,
index_1_39.ts?m=1676056675
#EXTINF:6.006,
index_1_40.ts?m=1676056675
```

Example선형 HLS 개인화된 매니페스트(창의적 광고 신호 사용):

MediaTailorDATERANGE가 생성하는 에는 고유한 ID 값이 있습니다. 고유성을 보장하기 위해([SCTE-35](#)를 [EXT-X-DATERANGE](#)로 매핑에 지정된 지침에 따라) MediaTailor는 가용 구간의 첫 번째 광고 세그먼트 MEDIA-SEQUENCE 수를 가용 구간 내의 광고 시퀀스 번호와 결합합니다.

다음 예제에서 MediaTailor는 MEDIA-SEQUENCE 421을 광고 위치 번호와 연결합니다.

```
#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 오리진(SCTE 신호 포함):

```

#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 오리진:

```
#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 개인 맞춤형 매니페스트:

MediaTailorPROGRAM-DATE-TIME는 광고 위치를 나타내는 HLS DATERANGE 요소의 앵커로 사용하기 위해 VOD 매니페스트에를 추가합니다.

MediaTailorDATERANGE가 생성하는 에는 고유한 ID 값이 있습니다. 고유성을 보장하기 위해([SCTE-35](#)를 [EXT-X-DATERANGE로 매핑](#)에 지정된 지침에 따라) MediaTailor는 가용 구간의 첫 번째 광고 세그먼트 MEDIA-SEQUENCE 수를 가용 구간 내의 광고 시퀀스 번호와 결합합니다.

다음 예제에서 MediaTailor는 MEDIA-SEQUENCE 421을 광고 위치 번호와 연결합니다.

```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-PLAYLIST-TYPE:VOD
#EXT-X-TARGETDURATION:7
#EXT-X-MEDIA-SEQUENCE:1
#EXT-X-DISCONTINUITY-SEQUENCE:0
#EXT-X-PROGRAM-DATE-TIME:1970-01-01T00:00:00Z
#EXTINF:6.0,
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SK0-22/asset-1/hls/
index_720p1500k_00001.ts
#EXTINF:6.0,
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SK0-22/asset-1/hls/
index_720p1500k_00002.ts
#EXTINF:6.0,
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SK0-22/asset-1/hls/
index_720p1500k_00003.ts
#EXTINF:6.0,
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SK0-22/asset-1/hls/
index_720p1500k_00004.ts
#EXTINF:4.0,
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SK0-22/asset-1/hls/
index_720p1500k_00005.ts
#EXT-X-DISCONTINUITY
#EXTINF:2.002,
../../../../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/vod-
variations/9810d863-8736-45fa-866e-be6d2c2bfa20/0/28
#EXTINF:2.002,
../../../../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/vod-
variations/9810d863-8736-45fa-866e-be6d2c2bfa20/0/29
#EXTINF:2.002,
../../../../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/vod-
variations/9810d863-8736-45fa-866e-be6d2c2bfa20/0/30
```



```

#EXTINF:2.002,
../.../.../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/vod-
variations/9810d863-8736-45fa-866e-be6d2c2bfa20/0/31
#EXTINF:2.002,
../.../.../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/vod-
variations/9810d863-8736-45fa-866e-be6d2c2bfa20/0/32
#EXTINF:2.002,
../.../.../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/vod-
variations/9810d863-8736-45fa-866e-be6d2c2bfa20/0/33
#EXTINF:2.002,
../.../.../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/vod-
variations/9810d863-8736-45fa-866e-be6d2c2bfa20/0/34
#EXTINF:1.001,
../.../.../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/vod-
variations/9810d863-8736-45fa-866e-be6d2c2bfa20/0/35
#EXT-X-DISCONTINUITY
#EXTINF:2.002,
../.../.../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/vod-
variations/9810d863-8736-45fa-866e-be6d2c2bfa20/0/36
#EXTINF:2.002,
../.../.../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/vod-
variations/9810d863-8736-45fa-866e-be6d2c2bfa20/0/37
#EXTINF:2.002,
../.../.../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/vod-
variations/9810d863-8736-45fa-866e-be6d2c2bfa20/0/38
#EXTINF:2.002,
../.../.../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/vod-
variations/9810d863-8736-45fa-866e-be6d2c2bfa20/0/39
#EXTINF:2.002,
../.../.../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/vod-
variations/9810d863-8736-45fa-866e-be6d2c2bfa20/0/40
#EXTINF:2.002,
../.../.../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/vod-
variations/9810d863-8736-45fa-866e-be6d2c2bfa20/0/41
#EXTINF:2.002,
../.../.../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/vod-
variations/9810d863-8736-45fa-866e-be6d2c2bfa20/0/42
#EXTINF:1.001,
../.../.../segment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/vod-
variations/9810d863-8736-45fa-866e-be6d2c2bfa20/0/43
#EXT-X-DISCONTINUITY
#EXTINF:2.0,
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SK0-22/asset-1/hls/
index_720p1500k_000006.ts

```

```
#EXTINF:6.0,
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SK0-22/asset-1/hls/
index_720p1500k_00007.ts
#EXTINF:6.0,
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SK0-22/asset-1/hls/
index_720p1500k_00008.ts
#EXTINF:6.0,
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SK0-22/asset-1/hls/
index_720p1500k_00009.ts
#EXTINF:6.0,
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SK0-22/asset-1/hls/
index_720p1500k_00010.ts
#EXTINF:6.0,
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SK0-22/asset-1/hls/
index_720p1500k_00011.ts
#EXTINF:6.0,
https://d3fch9e2fcarly.cloudfront.net/cunsco-media/SK0-22/asset-1/hls/
index_720p1500k_00012.ts
#EXT-X-ENDLIST
#EXT-X-DATERANGE:ID="5-1",START-DATE="1970-01-01T00:00:28.000Z",END-
DATE="1970-01-01T00:00:43.015Z",DURATION=15.015
#EXT-X-DATERANGE:ID="5-2",START-DATE="1970-01-01T00:00:43.015Z",END-
DATE="1970-01-01T00:00:58.030Z",DURATION=15.01
```

광고 메타데이터를 사용하여 DASH 매니페스트 개인화

MediaTailor는 광고 결정 서버(ADS)에서 반환한 크리에이티브로 매니페스트를 개인화합니다. 각 광고에 대해 MediaTailor에는 광고 기간에 걸친 EventStream 요소도 포함됩니다. Event 요소 형식은 SVA 기술 간행물 2023 버전의 [DASH 및 HLS의 광고 크리에이티브 신호](#) 섹션에 설명된 형식과 유사합니다.

슬레이트가 활성화된 구성에서 과소 채워진 광고 시간의 경우 MediaTailor는 슬레이트 기간을 가용 기간 종료에 추가하지만 EventStream 메타데이터는 추가하지 않습니다.

개인화된 매니페스트에 연결된 각 광고에 대해 MediaTailor는 Event 요소 내에 CDATA 요소로 표시되는 창의적인 메타데이터를 추가합니다.

Example선형 DASH 오리진(인라인 SCTE 속성):

```
<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"/>
  </AdaptationSet>
</Period>
</EventStream>
</EventStream>

```

```

    <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>

```

Example선형 DASH 개인화된 매니페스트(창의적 광고 신호 사용):

```

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

```

```

        <SegmentTemplate initialization="index_audio_${RepresentationID}_0_init.mp4?
m=1676062374" media="index_audio_${RepresentationID}_0_${Number}.mp4?m=1676062374"
presentationTimeOffset="3846790126" startNumber="151" timescale="48000">
            <SegmentTimeline>
                <S d="287744" t="3859981294"/>
                <S d="288768" t="3860269038"/>
                <S d="287744" t="3860557806"/>
                <S d="288768" t="3860845550"/>
                <S d="252928" t="3861134318"/>
            </SegmentTimeline>
        </SegmentTemplate>
        <Representation audioSamplingRate="48000" bandwidth="193007"
codecs="mp4a.40.2" id="2">
            <AudioChannelConfiguration
schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
        </Representation>
        <Representation audioSamplingRate="48000" bandwidth="193007"
codecs="mp4a.40.2" id="4">
            <AudioChannelConfiguration
schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
        </Representation>
        <Representation audioSamplingRate="48000" bandwidth="193007"
codecs="mp4a.40.2" id="6">
            <AudioChannelConfiguration
schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
        </Representation>
    </AdaptationSet>
    <SupplementalProperty schemeIdUri="urn:scte:dash:utc-time"
value="2023-02-10T21:02:31.007Z"/>
</Period>
<Period id="155_1" start="PT22H20M45.56S">
    <BaseURL>https://777788889999.mediatailor.us-west-2.amazonaws.com/
v1/dashsegment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/
emt/672ed481-4ffd-4270-936f-7c8403947f2e/155/155_1/</BaseURL>
    <EventStream schemeIdUri="urn:sva:advertising-wg:ad-id-signaling"
timescale="90000">
        <Event presentationTime="xxxxx" duration="1351350">
            <![CDATA[{"version": 1,"identifiers": [{"scheme":
"urn:smp:ul:060E2B34.01040101.01200900.00000000","value": "155_1","ad_position":
"155_1", "ad_type":"avail","creative_id": "123","tracking_uri": "../..../v1/
tracking/hashed-account-id/origin-id/session-id","custom_vast_data":"123abc"}]]]>
        </Event>
    </EventStream>
</Period>

```



```

    <AdaptationSet bitstreamSwitching="false" frameRate="30000/1001"
    mimeType="video/mp4" segmentAlignment="true" startWithSAP="1"
    subsegmentAlignment="true" subsegmentStartsWithSAP="1">
        <SegmentTemplate startNumber="1" timescale="90000"/>
        <Representation bandwidth="1800000" codecs="avc1.64001f" height="540"
    id="1" width="960">
            <SegmentTemplate initialization="asset_540_1_2init.mp4"
    media="asset_540_1_2_${Number%09d$.mp4" startNumber="1" timescale="90000">
                <SegmentTimeline>
                    <S d="180180" r="6" t="0"/>
                    <S d="90090" t="1261260"/>
                </SegmentTimeline>
            </SegmentTemplate>
        </Representation>
        <Representation bandwidth="1200000" codecs="avc1.64001e" height="360"
    id="3" width="640">
            <SegmentTemplate initialization="asset_360_1_1init.mp4"
    media="asset_360_1_1_${Number%09d$.mp4" startNumber="1" timescale="90000">
                <SegmentTimeline>
                    <S d="180180" r="6" t="0"/>
                    <S d="90090" t="1261260"/>
                </SegmentTimeline>
            </SegmentTemplate>
        </Representation>
        <Representation bandwidth="800000" codecs="avc1.640015" height="270" id="5"
    width="480">
            <SegmentTemplate initialization="asset_270_0_0init.mp4"
    media="asset_270_0_0_${Number%09d$.mp4" startNumber="1" timescale="90000">
                <SegmentTimeline>
                    <S d="180180" r="6" t="0"/>
                    <S d="90090" t="1261260"/>
                </SegmentTimeline>
            </SegmentTemplate>
        </Representation>
    </AdaptationSet>
    <AdaptationSet lang="eng" mimeType="audio/mp4" segmentAlignment="0">
        <SegmentTemplate initialization="asset_audio_128_3init.mp4"
    media="asset_audio_128_3_${Number%09d$.mp4" startNumber="1" timescale="48000"/>
        <Label>eng</Label>
        <Representation audioSamplingRate="48000" bandwidth="128000"
    codecs="mp4a.40.2" id="6">
            <SegmentTemplate initialization="asset_audio_128_3init.mp4"
    media="asset_audio_128_3_${Number%09d$.mp4" startNumber="1" timescale="48000">
                <SegmentTimeline>

```

```

        <S d="98304" t="0"/>
        <S d="96256" r="1" t="98304"/>
        <S d="95232" t="290816"/>
        <S d="96256" r="2" t="386048"/>
        <S d="48128" t="674816"/>
    </SegmentTimeline>
</SegmentTemplate>
<AudioChannelConfiguration
schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
</Representation>
</AdaptationSet>
</Period>
<Period id="155_2" start="PT22H21M0.575S">
    <BaseURL>https://777788889999.mediatailor.us-west-2.amazonaws.com/
v1/dashsegment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/
emt/672ed481-4ffd-4270-936f-7c8403947f2e/155/155_2/</BaseURL>
    <EventStream schemeIdUri="urn:sva:advertising-wg:ad-id-signaling"
timescale="90000">
        <Event presentationTime="0" duration="1351350">
            <![CDATA[{"version": 1,"identifiers": [{"scheme":
"urn:smp:ul:060E2B34.01040101.01200900.00000000","value": "155_2","ad_position":
"155_2", "ad_type":"avail","creative_id": "234","tracking_uri": "../.../v1/
tracking/hashed-account-id/origin-id/session-id","custom_vast_data":"123abc"}]]]>
        </Event>
    </EventStream>
    <AdaptationSet bitstreamSwitching="false" frameRate="30000/1001"
mimeType="video/mp4" segmentAlignment="true" startWithSAP="1"
subsegmentAlignment="true" subsegmentStartsWithSAP="1">
        <SegmentTemplate startNumber="1" timescale="90000"/>
        <Representation bandwidth="1800000" codecs="avc1.64001f" height="540"
id="1" width="960">
            <SegmentTemplate initialization="asset_540_1_2init.mp4"
media="asset_540_1_2_${Number%09d$.mp4" startNumber="1" timescale="90000">
                <SegmentTimeline>
                    <S d="180180" r="6" t="0"/>
                    <S d="90090" t="1261260"/>
                </SegmentTimeline>
            </SegmentTemplate>
        </Representation>
        <Representation bandwidth="1200000" codecs="avc1.64001e" height="360"
id="3" width="640">
            <SegmentTemplate initialization="asset_360_1_1init.mp4"
media="asset_360_1_1_${Number%09d$.mp4" startNumber="1" timescale="90000">
                <SegmentTimeline>

```

```

        <S d="180180" r="6" t="0"/>
        <S d="90090" t="1261260"/>
    </SegmentTimeline>
</SegmentTemplate>
</Representation>
<Representation bandwidth="800000" codecs="avc1.640015" height="270" id="5"
width="480">
    <SegmentTemplate initialization="asset_270_0_0init.mp4"
media="asset_270_0_0_$Number%09d$.mp4" startNumber="1" timescale="90000">
        <SegmentTimeline>
            <S d="180180" r="6" t="0"/>
            <S d="90090" t="1261260"/>
        </SegmentTimeline>
    </SegmentTemplate>
</Representation>
</AdaptationSet>
<AdaptationSet lang="eng" mimeType="audio/mp4" segmentAlignment="0">
    <SegmentTemplate initialization="asset_audio_128_3init.mp4"
media="asset_audio_128_3_$Number%09d$.mp4" startNumber="1" timescale="48000"/>
    <Label>eng</Label>
    <Representation audioSamplingRate="48000" bandwidth="128000"
codecs="mp4a.40.2" id="6">
        <SegmentTemplate initialization="asset_audio_128_3init.mp4"
media="asset_audio_128_3_$Number%09d$.mp4" startNumber="1" timescale="48000">
            <SegmentTimeline>
                <S d="98304" t="0"/>
                <S d="96256" r="1" t="98304"/>
                <S d="95232" t="290816"/>
                <S d="96256" r="2" t="386048"/>
                <S d="48128" t="674816"/>
            </SegmentTimeline>
        </SegmentTemplate>
        <AudioChannelConfiguration
schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
    </Representation>
</AdaptationSet>
</Period>
<Period id="155_3" start="PT22H21M15.59S">
    <BaseUrl>https://777788889999.mediatailor.us-west-2.amazonaws.com/
v1/dashsegment/94063eadf7d8c56e9e2edd84fdf897826a70d0df/
emt/672ed481-4ffd-4270-936f-7c8403947f2e/155/155_3/</BaseUrl>
    <EventStream schemeIdUri="urn:sva:advertising-wg:ad-id-signaling"
timescale="90000">
        <Event presentationTime="0" duration="1351350">

```

```

        <![CDATA[{"version": 1,"identifiers": [{"scheme":
"urn:smppte:ul:060E2B34.01040101.01200900.00000000","value": "155_3","ad_position":
"155_3", "ad_type":"avail","creative_id": "345","tracking_uri": "../v1/
tracking/hashed-account-id/origin-id/session-id","custom_vast_data":"123abc"}]]]>
    </Event>
</EventStream>
    <AdaptationSet bitstreamSwitching="false" frameRate="30000/1001"
mimeType="video/mp4" segmentAlignment="true" startWithSAP="1"
subsegmentAlignment="true" subsegmentStartsWithSAP="1">
        <SegmentTemplate startNumber="1" timescale="90000"/>
        <Representation bandwidth="1800000" codecs="avc1.64001f" height="540"
id="1" width="960">
            <SegmentTemplate initialization="asset_540_1_2init.mp4"
media="asset_540_1_2_${Number%09d$.mp4" startNumber="1" timescale="90000">
                <SegmentTimeline>
                    <S d="180180" r="6" t="0"/>
                    <S d="90090" t="1261260"/>
                </SegmentTimeline>
            </SegmentTemplate>
        </Representation>
        <Representation bandwidth="1200000" codecs="avc1.64001e" height="360"
id="3" width="640">
            <SegmentTemplate initialization="asset_360_1_1init.mp4"
media="asset_360_1_1_${Number%09d$.mp4" startNumber="1" timescale="90000">
                <SegmentTimeline>
                    <S d="180180" r="6" t="0"/>
                    <S d="90090" t="1261260"/>
                </SegmentTimeline>
            </SegmentTemplate>
        </Representation>
        <Representation bandwidth="800000" codecs="avc1.640015" height="270" id="5"
width="480">
            <SegmentTemplate initialization="asset_270_0_0init.mp4"
media="asset_270_0_0_${Number%09d$.mp4" startNumber="1" timescale="90000">
                <SegmentTimeline>
                    <S d="180180" r="6" t="0"/>
                    <S d="90090" t="1261260"/>
                </SegmentTimeline>
            </SegmentTemplate>
        </Representation>
    </AdaptationSet>
    <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>

```

Ad Decision Server(ADS) 상호 작용

MediaTailor는 VAST 응답의 크리에이티브 id 속성 값을 광고 ID 신호의 값으로 사용합니다. id 속성 값이 비어 있거나 VAST 응답에 없는 경우 MediaTailor는 광고 ID 신호에 빈 값을 배치합니다.

Example VAST 응답:

다음 샘플 VAST 응답에는 인라인 선형에 대한 id 속성 값이 포함됩니다Creative. MediaTailor는 사용자 지정 VAST Extension 요소에서 값을 추출하고 매니페스트의 크리에이티브 메타데이터에 해당 값을 배치합니다.

```

<?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>
    </Inline>
  </Ad>
</VAST>

```

```

    <Impression><![CDATA[https://n8ljfs0xxx.execute-api.us-
west-2.amazonaws.com/v1/impression]]></Impression>
    <Creatives>
      <Creative sequence="3" apiFramework="inLine" id="1234">
        <Linear>
          <Duration>00:00:15</Duration>
          <MediaFiles>
            <MediaFile id="00002" delivery="progressive" type="video/
mp4" width="1280" height="720"><![CDATA[https://d3re4i3vgppxxx.cloudfront.net/Media/
Bumpers/AD-caribbean2-15-HD.mp4]]></MediaFile>
          </MediaFiles>
        </Linear>
      </Creative>
    </Creatives>
    <Extensions>
      <Extension type="creative_signaling"><![CDATA[999999|
TVN1DDNpFTchtpRj,E5TfTtcYd5IEzvEt,ChA050HcvWRGFY6Zp5VSS1xUEJ2B9p8GGhQIDzIQkFeQC-
Ho67FR3P9qNa6khSAGKgAyAA]]></Extension>
    </Extensions>
  </InLine>
</Ad>
</VAST>

```

클라이언트 측 추적 API

다음 예제는 플레이어 SDK가 creativeId 및를 사용하여 클라이언트 측 추적 응답 페이로드의 전체 추적 이벤트 데이터와 매니페스트의 광고 메타데이터를 연결하는 방법을 보여줍니다adId.

Example JSON 메시지:

```

{
  "avails": [
    {
      "adBreakTrackingEvents": [],
      "ads": [
        {
          "adId": "5",
          "adParameters": "",
          "adProgramDateTime": null,
          "adSystem": "2.0",
          "adTitle": "AD-caribbean2-15",
          "adVerifications": [],
          "companionAds": [],

```

```
    "creativeId": "1234",
    "creativeSequence": "2",
    "duration": "PT15S",
    "durationInSeconds": 15,
    "extensions": [],
    "mediaFiles": {
      "mediaFilesList": [],
      "mezzanine": ""
    },
    "skipOffset": null,
    "startTime": "PT30S",
    "startTimeInSeconds": 30,
    "trackingEvents": [
      {
        "beaconUrls": [
          "https://myServer/impression"
        ],
        "duration": "PT15S",
        "durationInSeconds": 15,
        "eventId": "5",
        "eventProgramDateTime": null,
        "eventType": "impression",
        "startTime": "PT30S",
        "startTimeInSeconds": 30
      }
    ],
    "vastAdId": ""
  },
  "availId": "5",
  "availProgramDateTime": null,
  "duration": "PT15S",
  "durationInSeconds": 15,
  "meta": null,
  "nonLinearAdsList": [],
  "startTime": "PT30S",
  "startTimeInSeconds": 30
}
],
"nextToken": "UFQ1TTM0Ljk2N1NfMjAyMi0xMS0xOFQwNDZMzo1Mi4yNDUxOTdaXzE%3D",
"nonLinearAvails": []
}
```


AWS Elemental MediaTailor 를 사용하여 선형 어셈블 스트림 생성

AWS Elemental MediaTailor 채널 어셈블리는 라이브 콘텐츠와 혼합된 기존 온디맨드 비디오(VOD) 콘텐츠를 사용하여 선형 스트리밍 채널을 생성할 수 있는 매니페스트 전용 서비스입니다. MediaTailor는 오리진 서버에서 직접 제공되는 콘텐츠 세그먼트에 절대 접촉하지 않습니다. 대신 MediaTailor는 오리진에서 매니페스트를 가져오고 이를 사용하여 기본 콘텐츠 세그먼트를 참조하는 라이브 슬라이딩 매니페스트 창을 조합합니다. 채널 어셈블리는 자산에서 자산으로 원활하게 재생하는 데 필요한 미디어 시퀀스 번호와 같은 항목을 추적합니다. 선형 조립 스트림은 기존의 다중 비트레이트 인코딩 및 패키징 VOD 콘텐츠를 사용하여 낮은 실행 비용으로 생성됩니다.

SCTE-35 마커로 콘텐츠를 조건화하지 않고도 프로그램에 광고 시간을 삽입하여 채널 어셈블리 선형 스트림을 쉽게 수익화할 수 있습니다. MediaTailor 광고 삽입 서비스 또는 서버 측 광고 삽입 서비스와 함께 채널 어셈블리를 사용할 수 있습니다.

채널 어셈블리를 시작하려면 섹션을 참조하세요 [the section called “MediaTailor 채널 어셈블리 시작하기”](#).

채널 어셈블리를 콘텐츠 전송 네트워크(CDN)와 통합하는 방법에 대한 자세한 내용은 섹션을 참조하세요 [채널 어셈블리 및 CDN을 사용하여 MediaTailor 선형 채널 구축](#).

주제

- [채널 어셈블리 워크플로 유형](#)
- [소스 위치 작업](#)
- [채널 작업](#)
- [채널 일정에 프로그램 추가](#)
- [채널 스트림에 개인화된 광고 및 광고 시간 삽입](#)
- [채널 재생 시간 전환](#)

채널 어셈블리 워크플로 유형

AWS Elemental MediaTailor 채널 어셈블리는 선형 스트리밍 채널을 생성하기 위한 세 가지 워크플로 유형을 지원합니다. 각 워크플로 유형은 특정 사용 사례를 충족하기 위해 서로 다른 소스 콘텐츠 조합을 사용합니다. 이 주제에서는 각 워크플로 유형, 사용 시기, 자세한 설정 절차 링크에 대해 설명합니다.

워크플로 유형	설명	재생 모드	사용 사례:
VOD-to-live	사전 레코딩된 VOD 자산을 라이브 선형 스트림으로 어셈블합니다.	루프 또는 선형	콘텐츠 라이브러리의 24/7 채널
Live-to-live	라이브 소스 피드를 예약된 채널의 프로그램으로 전달합니다.	선형 전용	뉴스, 스포츠, 리전 프로그래밍
VOD + Live-to-live(혼합)	VOD와 라이브 소스를 단일 채널 일정으로 결합합니다.	선형 전용	예약된 라이브 이벤트를 포함하는 대부분 VOD 콘텐츠가 있는 채널

VOD-to-live

VOD-to-live 채널은 사전 인코딩되고 사전 패키징된 VOD 콘텐츠에서 선형 스트림을 완전히 생성합니다. 스트림은 라이브 슬라이딩 매니페스트 창이 있는 라이브 브로드캐스트처럼 보이지만 VOD 자산만 사용합니다. 프로그램을 예약할 때 광고 시간을 정의하므로 소스 콘텐츠에 SCTE-35 마커가 필요하지 않습니다.

사용해야 하는 경우

- VOD 자산 라이브러리가 있고 연중무휴 선형 채널을 생성하려고 합니다.
- 광고 시간을 통해 콘텐츠를 수익화하고 싶지만 소스 콘텐츠에 SCTE-35 마커가 없습니다.
- 라이브 인코딩 인프라가 필요하지 않은 간단하고 저렴한 선형 스트림이 필요합니다.

작동 방식

1. 콘텐츠 준비 - VOD 자산을 인코딩하고 패키징합니다. 오리진 서버(Amazon S3 AWS Elemental MediaPackage 또는 HTTP 오리진)에서 호스팅합니다.
2. 소스 위치 생성 - MediaTailor에 오리진 서버를 등록합니다.
3. VOD 소스 추가 - 각 콘텐츠를 등록하고 패키지 구성 및 소스 그룹을 정의합니다.
4. 채널 생성 - 재생 모드(루프 또는 선형)를 선택하고 출력을 구성합니다.

5. 일정 프로그램 - 재생할 순서대로 채널을 일정에 추가합니다.
6. 광고 시간 구성 - 각 프로그램에 광고 시간을 추가합니다. 맞춤형 광고를 위해 SSAI 구성에 연결합니다.
7. 채널을 시작하고 CDN 구성 - 채널을 시작합니다. 선택적으로 확장 및 비용 절감을 위해 앞에 CDN을 추가합니다.

재생 모드 선택

VOD-to-live 채널은 두 가지 재생 모드를 지원합니다.

- 루프(기본 티어) - 프로그램은 무한 루프에서 back-to-back 재생됩니다. 마지막 프로그램이 완료되면 재생이 첫 번째 프로그램으로 돌아갑니다. 24/7 채널에 대한 가장 간단한 옵션입니다. 루프 모드는 라이브 소스 또는 필러 슬레이트를 지원하지 않습니다.
- 선형(표준 티어) - 프로그램은 한 번 순차적으로 재생됩니다. 선형 모드는 프로그램 간 간격에 대해 시간 기반 예약 및 필러 슬레이트를 지원합니다.

콘텐츠 요구 사항

원활한 전환을 위해 동일한 채널의 모든 VOD 자산에는 다음이 있어야 합니다.

- 소스 그룹의 모든 소스에서 동일한 수의 변환(비디오, 오디오, 자막 등)
- 모든 자산에서 일관된 세그먼트 지속 시간. 2~6초를 권장합니다.
- 해당 변환에 대한 매칭 비트 전송률 및 해상도

Important

채널 어셈블리는 제목별 인코딩 또는 자동 ABR 인코딩을 지원하지 않습니다. 이러한 메서드는 변환 수와 지속 시간이 서로 다른 자산을 생성하므로 프로그램 전환 시 재생이 실패합니다.

광고 시간 및 수익 창출

일정에 프로그램을 추가할 때 광고 시간을 정의합니다. 각 광고 시간에는 프로그램 시작 시점부터 밀리초 단위의 오프셋이 필요합니다. 오프셋은 VOD 소스의 모든 트랙에 있는 세그먼트 경계와 일치해야 합니다.

VOD 소스 매니페스트에 광고 마커가 포함될 수 있습니다. HLS의 경우 지속 시간이 지정된 EXT-X-DATERANGE 또는 EXT-X-CUE-OUT 태그입니다. DASH의 경우를 사용합니다. EventStream. MediaTailor가 이러한 마커를 감지하면 광고 시간을 설정할 때 오프셋이 옵션으로 표시됩니다. 이는 가장 정확한 광고 배치를 제공합니다.

개인 맞춤형 광고를 제공하려면 채널을 MediaTailor SSAI 재생 구성과 페어링합니다. 콘텐츠 소스를 채널의 재생 URL로 설정합니다.

팁

Topic	세부 정보
광고 시간 배치	세그먼트 경계와 일치하지 않는 광고 중단 오프셋은 가장 가까운 경계로 조정됩니다. 보다 정확한 광고 배치를 위해 더 짧은 세그먼트(2~4초)를 사용합니다.
서명된 URL 오리진	MediaTailor는 소스를 처음 추가할 때 VOD 소스 매니페스트를 캐싱합니다. 만료 시간이 짧은 서명된 URLs 사용하는 경우 캐시된 URLs 재생 전에 만료될 수 있습니다. UpdateVodSource API를 호출하여 MediaTailor가 매니페스트를 다시 가져오도록 합니다.
변환 불일치	프로그램이 원활하게 전환되지 않는 경우 채널의 모든 VOD 소스의 변환 수가 동일한지 확인하고 동일한 ABR 래더를 사용합니다.
미리 예약	대기 시간 최소 10분 전에 실행 중인 선형 채널에 프로그램을 추가합니다. 채널 어셈블리는 이 기간 내에 프로그램 처리를 시작합니다.

첫 번째 채널 생성에 대한 지침은 섹션을 참조하세요 [the section called “MediaTailor 채널 어셈블리 시작하기”](#).

Live-to-live

live-to-live 채널은 라이브 소스 피드를 예약된 채널의 프로그램으로 전달합니다. 채널은 일정에 따라 서로 다른 라이브 소스 간에 전환됩니다. 모든 콘텐츠가 라이브 상태이고 여러 라이브 피드를 단일 선형 채널로 어셈블하려는 경우 이 워크플로를 사용합니다.

사용해야 하는 경우

- 라이브 인코더/패키저가 여러 개 있고 일정에 따라 라이브 인코더/패키저 간에 전환하려고 합니다.

- 하루 중 서로 다른 시간에 국가 라이브 피드와 지역 라이브 피드 간에 전환하는 리전 채널을 구축하고 있습니다.
- 24/7 라이브 피드(뉴스, 스포츠 네트워크)가 있으며 여러 소스에서 끌어오는 엄선된 채널을 생성하려고 합니다.

작동 방식

1. 라이브 인코딩 파이프라인 설정 - AWS Elemental MediaLive AWS Elemental MediaPackage 또는 라이브 패키징 오리진을 사용하여 라이브 콘텐츠를 인코딩하고 패키징합니다.
2. 소스 위치 생성 - 라이브 콘텐츠의 오리진 서버를 MediaTailor에 등록합니다.
3. 라이브 소스 추가 - 각 라이브 피드를 등록하고 패키지 구성 및 소스 그룹을 정의합니다.
4. 채널 생성 - 선형 재생 모드(라이브 소스에 필요)를 선택하고 출력을 구성합니다.
5. 일정 프로그램 - 채널 일정에 라이브 소스 프로그램을 추가합니다. 각 라이브 프로그램의 기간을 지정합니다.
6. 채널을 시작하고 CDN 구성 - 채널을 시작합니다. 선택적으로 확장 및 비용 절감을 위해 앞에 CDN을 추가합니다.

재생 모드

Live-to-live 채널에는 선형(표준 티어) 재생 모드가 필요합니다. 루프 모드는 라이브 소스를 지원하지 않습니다.

콘텐츠 요구 사항

에 나열된 일반 요구 사항 외에도 [the section called “소스 위치 작업”](#):

- HLS 라이브 소스 - 매니페스트 창 의 첫 번째 세그먼트와 모든 불연속성에 #EXT-X-PROGRAM-DATE-TIME 태그를 제공해야 합니다.
- HLS 광고 마커 - 광고 마커를 로 구성해야 합니다 DATERANGE.
- 소스 매니페스트 기간 - 채널 어셈블리 채널의 매니페스트 기간보다 30초 이상 긴 매니페스트 기간 기간을 사용합니다.
- 대상 기간 - 대상 기간을 채널의 기존 소스 기간과 일치시킵니다.
- 하위 재생 목록 - 하위 재생 목록의 수를 채널에 있는 다른 소스의 수와 일치시킵니다.

MediaPackage를 사용한 라이브 소스 구성

를 패키징 오리진 AWS Elemental MediaPackage 으로 사용하는 경우 다음 구성을 사용하는 것이 좋습니다.

HLS 설정:

설정	값	필요성	참고
엔드포인트 유형	Apple HLS 또는 CMAF	필수	TS 세그먼트Apple HLS에 사용, MP4 세그먼트CMAF에 사용
ProgramDateTimeIntervalSeconds	1	필수	불연속성 시 재생 문제를 방지합니다.
PlaylistWindowSeconds	CA 매니페스트 기간보다 30초 더 깁니다.	필수	
AdMarkers	DATERANGE	필수	광고 마커를 통과할 때 필요합니다.
IncludeIframeOnlyStream	비활성화됨	권장	

DASH 설정:

설정	값	필요성	참고
ManifestLayout	FULL	권장	
SegmentTemplateFormat	NUMBER_WITH_TIMELINE 또는 TIME_WITH_TIMELINE	권장	NUMBER_WITH_DURATION 는 지원되지 않습니다.

설정	값	필요성	참고
ManifestWindowSeconds	CA 매니페스트 기간보다 30초 더 길니다.	필수	
PeriodTriggers	ADS	필수	광고 마커를 통과할 때 필요합니다.

광고 시간

라이브 소스의 경우 채널 어셈블리 일정에서 광고 시간을 정의하지 않습니다. 대신 라이브 소스 매니페스트의 광고 중단은 자동으로 전달됩니다.

- HLS - EXT-X-DATERANGE 태그를 사용한 광고 중단이 전달됩니다.
- DASH -를 사용한 광고 중단EventStream이 전달됩니다.

전달된 광고 시간을 수익화하려면 채널을 MediaTailor SSAI 재생 구성과 페어링합니다.

팁

Topic	세부 정보
프로그램 기간	각 라이브 소스 프로그램에 기간을 지정해야 합니다. 기간이 만료되면 채널 어셈블리가 일정의 다음 프로그램으로 전환됩니다.
필러 슬레이트	일정에 프로그램 간에 간격이 있는 경우 MediaTailor는 필러 슬레이트를 재생합니다. 슬레이트 지속 시간이 간격 지속 시간보다 짧으면 MediaTailor는 슬레이트를 루프합니다. 채널을 생성할 때 필러 슬레이트를 구성합니다.
리전 채널	라이브 소스를 사용하여 리전 채널을 생성합니다. 국가 콘텐츠에 대해 인코더/패키저 하나를 실행한 다음 해당 리전이 가동되면 리전 인코더를 실행합니다. 리전별로 고유한 일정을 가진 별도의 채널을 생성합니다.

Topic	세부 정보
전환 타이밍	MediaTailor는 공통 세그먼트 경계에서 프로그램 재생을 시작합니다. 절대 전환의 경우 예약된 시간과 실제 전환 간에 최대 하나의 분산 세그먼트를 예상합니다.
비용 인코딩	콘텐츠가 라이브 상태일 때만 인코더를 실행하여 비용을 최소화합니다. 라이브 이벤트 간의 간격을 포함하도록 채널에서 필러 슬레이트를 구성합니다.

VOD + Live-to-live(혼합)

혼합 채널은 VOD 소스와 라이브 소스를 단일 일정으로 결합합니다. 가장 유연한 워크플로 유형입니다. 채널이 예약된 라이브 이벤트가 분산된 주로 VOD 콘텐츠이거나 라이브 이벤트와 사전 레코딩된 콘텐츠 간의 간격을 연결하려는 경우에 사용합니다.

사용해야 하는 경우

- 채널은 대부분 VOD 콘텐츠이지만 라이브 이벤트(야간 뉴스, 주간 스포츠, 특별 방송)가 예약되어 있습니다.
- 하루 종일 VOD 필러 콘텐츠와 라이브 피드 간에 전환하려고 합니다.
- 라이브 이벤트 기간 사이에 사전 레코딩된 콘텐츠를 사용하는 채널을 구축하고 있습니다.

작동 방식

1. 콘텐츠 파이프라인 설정 - VOD 자산과 라이브 콘텐츠를 모두 인코딩하고 패키징합니다. 오리진 서버에서 호스팅합니다.
2. 소스 위치 생성 - 오리진 서버를 MediaTailor에 등록합니다. VOD 및 라이브 콘텐츠에 대해 별도의 소스 위치를 사용하거나 오리진을 공유하는 경우 단일 소스 위치를 사용할 수 있습니다.
3. VOD 소스 및 라이브 소스 추가 - 각각 패키지 구성 및 소스 그룹을 사용하여 VOD 자산 및 라이브 피드를 등록합니다.
4. 채널 생성 - 선형 재생 모드(라이브 소스를 사용할 때 필요)를 선택하고 출력을 구성합니다.
5. 일정 프로그램 - 채널 일정에 VOD 및 라이브 소스 프로그램을 혼합하여 추가합니다. 특정 시간에 시작해야 하는 라이브 프로그램에 절대 전환을 사용합니다. 격차를 채우는 VOD 프로그램에 상대적 전환을 사용합니다.

6. 광고 시간 구성 - VOD 프로그램에 광고 시간을 추가합니다. 라이브 프로그램의 경우 광고 중단은 소스 매니페스트에서 전달됩니다.
7. 채널을 시작하고 CDN 구성 - 채널을 시작합니다. 선택적으로 확장 및 비용 절감을 위해 앞에 CDN을 추가합니다.

재생 모드

혼합 채널에는 선형(표준 계층) 재생 모드가 필요합니다. 루프 모드는 라이브 소스를 지원하지 않습니다.

콘텐츠 요구 사항

혼합 채널은 VOD 및 라이브 소스 모두에 대한 콘텐츠 요구 사항을 충족해야 합니다.

- 소스 그룹 내의 모든 소스(VOD 및 라이브)는 변환 수가 동일해야 합니다.
- VOD 및 라이브 소스에는 일치하는 대상 세그먼트 기간이 있어야 합니다.
- 라이브 소스는 HLS에 대한 EXT-X-PROGRAM-DATE-TIME 태그를 포함하여 라이브 소스에 대한 요구 사항을 충족해야 합니다.

패턴 예약

혼합 채널은 프로그램에 대해 두 가지 전환 유형을 지원합니다.

- 상대 - 프로그램이 일정의 다른 프로그램 앞이나 뒤에 재생됩니다. VOD 소스 프로그램에만 사용할 수 있습니다.
- 절대 - 프로그램은 특정 벽시계 시간에 재생됩니다. 선형 모드에서 VOD 및 라이브 소스 프로그램 모두에 사용할 수 있습니다.

일반적인 예약 패턴:

패턴	설명	예제
예정된 라이브 윈도우가 있는 VOD 필러	VOD 프로그램은 특정 시간에 삽입된 라이브 프로그램으로 일정을 채웁니다.	오후 8시에 라이브 스포츠 이벤트로 전환되는 영화 채널입니다.

패턴	설명	예제
VOD 중간 광고가 있는 라이브 기본	라이브 피드는 대부분의 시간 동안 실행되며, 휴식 시간이나 업무 외 시간에는 VOD 프로그램을 사용합니다.	라이브 브로드캐스트 간에 사전 레코딩된 세그먼트를 재생하는 뉴스 채널입니다.
Time-of-day 전환	서로 다른 소스(라이브 또는 VOD)가 하루 중 서로 다른 시간에 재생됩니다.	프라임 타임 동안의 국가 피드, 오전의 리전별 라이브 피드, 야간 VOD 콘텐츠.

광고 시간

광고 중단 동작은 각 프로그램의 소스 유형에 따라 다릅니다.

- VOD 프로그램 - 일정에서 광고 시간을 밀리초 단위로 오프셋하여 정의합니다. 맞춤형 광고를 위해 SSAI 구성에 연결합니다.
- 라이브 프로그램 - 광고 중단은 라이브 소스 매니페스트에서 자동으로 전달됩니다. HLS는 EXT-X-DATERANGE 태그를 사용하고 DASH는 EventStream를 사용합니다.

팁

Topic	세부 정보
VOD에서 라이브로 전환	절대 시간에 VOD 프로그램에서 라이브 프로그램으로 전환할 때 MediaTailor는 라이브 프로그램의 시작 시간에 가장 가까운 공통 세그먼트 경계에서 VOD 프로그램을 잘라냅니다.
라이브에서 VOD로 전환	라이브 프로그램의 기간이 만료되면 MediaTailor는 일정의 다음 프로그램으로 전환합니다. 다음 프로그램이 VOD인 경우 콘텐츠 요구 사항이 충족되는 한 재생이 원활합니다.
필러 슬레이트	채널에서 필러 슬레이트를 구성합니다. MediaTailor는 프로그램 간의 간격 동안 슬레이트를 재생합니다. 프로그램 중간에 라이브 소스를 사용할 수 없게 되면 채널 어셈블리는 프로그램 기간이 만료될 때까지 마지막으로 알려진 세그먼트를 계속 제공합니다.

Topic	세부 정보
혼합 광고 수익화	전체 채널에 대해 단일 SSAI 구성을 사용할 수 있습니다. MediaTailor는 동일한 스트림에서 예약된 광고 시간(VOD 프로그램)과 전달된 광고 시간(라이브 프로그램)을 모두 처리합니다.
일정 유연성	대기 시간 최소 10분 전에 VOD 프로그램을 추가합니다. 절대 전환을 사용하는 라이브 프로그램의 경우 실행 중인 채널에서 최소 15분 전에 예약합니다.

올바른 워크플로 선택

다음 결정 가이드를 사용하여 사용 사례에 적합한 워크플로 유형을 선택합니다.

- VOD 콘텐츠만 있고 시간 기반 예약이 필요하지 않은 경우 → 루프 재생 모드에서 VOD-to-live를 사용합니다.
- VOD 콘텐츠만 있고 시간 기반 예약이 필요한 경우 → 선형 재생 모드에서 VOD-to-live를 사용합니다.
- 라이브 콘텐츠만 있는 경우 → Live-to-live를 사용합니다.
- VOD 콘텐츠와 라이브 콘텐츠가 모두 있는 경우 → VOD + Live-to-live(혼합)를 사용합니다.

기능 비교

기능	VOD-to-live(루프)	VOD-to-live(선형)	Live-to-live	혼합형
VOD 소스	예	예	아니요	예
라이브 소스	아니요	아니요	예	예
절대 전환	아니요	예	예	예
상대 전환	예	예	아니요	예(VOD만 해당)
예약된 광고 시간	예	예	아니요	예(VOD만 해당)

기능	VOD-to-live(루프)	VOD-to-live(선형)	Live-to-live	혼합형
패스스루 광고 시간	아니요	아니요	예	예(라이브만 해당)
필러 슬레이트	아니요	예	예	예
연중무휴 반복	예	아니요	아니요	아니요
시간 기반 예약	아니요	예	예	예
채널 티어	기본	표준	표준	표준

다음 단계

- [the section called “MediaTailor 채널 어셈블리 시작하기”](#) Step-by-step 자습서
- [the section called “소스 위치 작업”](#) - 소스 위치 및 콘텐츠 관리를 위한 세부 절차
- [the section called “채널 작업”](#) - 채널 구성 옵션 및 재생 모드
- [the section called “프로그램 추가”](#) - VOD 및 라이브 소스를 사용하여 프로그램 예약
- [the section called “광고 및 광고 시간 삽입”](#) - SSAI를 통한 수익화
- [the section called “CDN을 사용한 채널 어셈블리”](#) - 규모 및 비용 절감을 위한 CDN 통합
- [the section called “채널 재생 시간 전환”](#) - 캐치업 및 스타트오버 기능 활성화

소스 위치 작업

소스 위치는 소스 콘텐츠가 저장되는 오리진 서버를 나타냅니다. 소스 위치는 Amazon S3, 표준 웹 서버, Amazon CloudFront와 같은 콘텐츠 전송 네트워크(CDN) 또는와 같은 패키징 오리진일 수 있습니다 AWS Elemental MediaPackage. MediaTailor는 소스 위치에서 콘텐츠 매니페스트를 검색하고 이를 사용하여 채널의 선형 스트림을 어셈블합니다.

이 주제에서는 AWS Elemental MediaTailor 콘솔을 사용하여 소스 위치를 생성하고 삭제하는 방법과 VOD 소스로 작업하는 방법을 설명합니다.

주제

- [소스 위치 생성](#)

- [소스 위치에 대한 인증 구성](#)
- [VOD 소스 작업](#)
- [라이브 소스 작업](#)
- [패키지 구성 사용](#)
- [매니페스트 캐싱](#)

소스 위치 생성

다음 절차에서는 MediaTailor 콘솔을 사용하여 소스 위치를 생성하는 방법을 설명합니다. MediaTailor API를 사용하여 소스 위치를 생성하는 방법에 대한 자세한 내용은 API 참조의 [CreateSourceLocation](#)을 AWS Elemental MediaTailor 참조하세요.

소스 위치를 생성하려면

1. <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.
2. 탐색 창에서 채널 어셈블리 > 소스 위치를 선택합니다.
3. 탐색 모음에서 소스 위치 생성을 선택합니다.
4. 소스 위치 구성에서 오리진 서버의 이름과 기본 URL을 입력합니다.
 - 이름: my-origin과 같은 소스 위치의 식별자입니다.
 - 기본 URL: <https://111111111111.cloudfront.net> 같이 콘텐츠가 저장된 오리진 서버의 프로토콜 및 기본 URL입니다. URL은 http:// 또는 https:// 접두사가 붙은 표준 HTTP URL 형식이어야 합니다.

소스 위치가 Amazon S3 버킷이고 Amazon S3 액세스 인증에 서명 버전 4를 사용SigV44 사용을 선택합니다. Amazon S3 AWS Amazon S3 고급 정보는 단원을 참조하십시오 [소스 위치에 대한 인증 구성](#).
5. 액세스 구성에서 선택적으로 소스 위치에 대한 인증을 구성합니다.
 - 액세스 유형: MediaTailor가 소스 위치의 오리진에 저장된 콘텐츠에 액세스하는 데 사용하는 인증 유형을 선택합니다.
 - Amazon S3용 SigVSigV4 - MediaTailor는 Amazon 서명 버전 4(SigV4)를 사용하여 오리진에 대한 요청을 승인합니다. 자세한 내용은 [the section called “SigV4를 사용하여 Amazon S3에 대한 요청 인증”](#) 단원을 참조하십시오.

- Secrets Manager 액세스 토큰 인증 - MediaTailor는 사용자가 생성, 소유 및 관리하는 Secrets Manager 및 AWS KMS 고객 관리형 키를 사용하여 MediaTailor와 오리진 간의 액세스 토큰 인증을 용이하게 합니다. Secrets Manager 액세스 토큰 인증을 구성하는 방법에 대한 자세한 내용은 섹션을 참조하세요 [the section called “AWS Secrets Manager 액세스 토큰 인증 작업”](#).
- 헤더 이름 - HTTP 헤더 이름을 지정합니다. MediaTailor는 HTTP 헤더를 사용하여 콘텐츠 매니페스트 요청의 오리진에 액세스 토큰을 전송합니다. x-amz- 또는 로 시작하지 않는 한 헤더 이름을 사용할 수 있습니다 x-amzn-. [MediaPackage CDN 권한](#) 부여와 통합하는 경우 헤더 값은 여야 합니다 X-MediaPackage-CDNIdentifier.
- 보안 암호 문자열 키 - Secrets Manager 보안 암호에 지정한 SecretString 키입니다. 예를 들어 아래와 같은 키 및 값 페어가 SecretString 포함된 경우 {"MyHeaderName": "11111111-2222-3333-4444-111122223333"} MyHeaderName는 이 필드에 입력하는 SecretString 키입니다.
- 보안 암호 ARN - 액세스 토큰을 보유한 보안 암호의 ARN입니다. step-by-step 가이드는 단원을 참조하십시오 [2단계: AWS Secrets Manager 보안 암호 생성](#).

6. 세그먼트 전송 서버 구성에서 필요에 따라 콘텐츠 세그먼트를 전송하도록 서버를 구성합니다.

- 기본 세그먼트 전송 서버 사용: CDN과 같은 콘텐츠 세그먼트를 전송하는 데 사용되는 서버의 기본 URL을 입력합니다. 소스 위치 서버와 다른 서버를 사용하여 콘텐츠 세그먼트를 제공하려는 경우 기본 세그먼트 호스트 이름을 구성합니다. 예를 들어, 기본 HTTP URL(MediaTailor가 매니페스트에 액세스하는 데 사용하는 것)과 기본 세그먼트 기본 URL(플레이어가 콘텐츠 세그먼트에 액세스하는 데 사용하는 것)에 대해 다른 CDN 구성을 사용하여 플레이어의 오리진 매니페스트에 대한 액세스를 제한할 수 있습니다. 값을 입력하지 않으면 MediaTailor는 세그먼트 전송을 위해 기본적으로 소스 위치 서버로 설정됩니다.
- 명명된 세그먼트 전송 서버 사용: 기본 세그먼트 전송 서버를 구성한 경우 추가 세그먼트 전송 서버를 구성할 수도 있습니다. 각 이름에는 고유한 이름과 기본 URL이 있어야 합니다. 기본 URL은 전체 HTTP URL이거나 /some/path/와 같은 상대 경로일 수 있습니다. 이름은 MediaTailor가 콘텐츠 세그먼트에 대한 요청을 수신할 때 사용해야 하는 서버를 식별하는 데 사용됩니다. 요청에 헤더가 포함되어 X-MediaTailor-SegmentDeliveryConfigurationName 있고 헤더 값이 이름과 일치하는 경우 해당 기본 URL이 콘텐츠를 제공하는 데 사용됩니다. 헤더가 요청에 포함되어 있지 않거나 이름과 일치하지 않는 경우 기본 세그먼트 전송 서버가 사용됩니다.

7. 소스 위치 생성을 선택합니다.

8. 소스 위치를 더 추가하려면 2~6단계를 반복합니다.

소스 위치에 대한 인증 구성

액세스 구성을 사용하여 소스 위치에 대한 인증을 구성합니다. 액세스 구성이 켜져 있는 경우 MediaTailor는 MediaTailor와 오리진 간에 요청이 승인된 경우에만 오리진에서 소스 매니페스트를 검색합니다. 액세스 구성은 기본적으로 꺼져 있습니다.

MediaTailor는 다음 인증 유형을 지원합니다.

- Amazon S3 인증을 위한 SigVSigV4
- AWS Secrets Manager 액세스 토큰
- MediaPackage 버전 2(v2) 인증을 위한 SigV4

이 장에서는 소스 위치 인증을 위해 Amazon S3용 SigVSigV4, MediaPackage v2 및 AWS Secrets Manager 액세스 토큰을 사용하는 방법을 설명합니다.

자세한 내용은 해당 주제를 선택합니다.

주제

- [SigV4를 사용하여 Amazon S3에 대한 요청 인증](#)
- [MediaPackage 버전 2용 SigV4 작업](#)
- [AWS Secrets Manager 액세스 토큰 인증 작업](#)

SigV4를 사용하여 Amazon S3에 대한 요청 인증

Amazon S3용 서명 버전 4(SigV4)는 HTTPS를 통해 Amazon S3에 대한 요청을 인증하는 데 사용되는 서명 프로토콜입니다. Amazon S3용 SigV4를 사용하는 경우 MediaTailor는 오리진으로 사용되는 Amazon S3 버킷에 대한 HTTPS 요청에 서명된 권한 부여 헤더를 포함합니다. 서명된 권한 부여 헤더가 유효하면 오리진이 요청을 이행합니다. 유효하지 않으면 요청이 실패합니다.

용 SigV4에 대한 일반적인 정보는 Amazon S3 API AWS Key Management Service참조의 [요청 인증 \(AWS 서명 버전 4\)](#) 주제를 참조하세요.

Note

MediaTailor는 항상 SigV4를 사용하여 이러한 오리진에 대한 요청에 서명합니다.

요구 사항

소스 위치에 대해 SigV4 for Amazon S3 인증을 활성화하는 경우 다음 요구 사항을 충족해야 합니다.

- MediaTailor가 IAM에서 mediatailor.amazonaws.com 보안 주체 액세스 권한을 부여하여 Amazon S3 버킷에 액세스하도록 허용해야 합니다. IAM에서 액세스를 구성하는 방법에 대한 자세한 내용은 AWS Identity and Access Management 사용 설명서의 [액세스 관리](#)를 참조하세요.
- mediatailor.amazonaws.com 서비스 보안 주체는 VOD 소스 패키지 구성에서 참조하는 모든 다변량 재생 목록을 읽을 수 있는 권한이 있어야 합니다.
- API 호출자는 MediaTailor VOD 소스 패키지 구성에서 참조하는 모든 다변량 재생 목록을 읽을 수 있는 s3:GetObject IAM 권한이 있어야 합니다.
- MediaTailor 소스 위치 기본 URL은 Amazon S3 가상 호스팅 스타일 요청 URL 형식을 따라야 합니다. 예: [https://*bucket-name*.s3.*Region*.amazonaws.com/*key-name*](https://bucket-name.s3.Region.amazonaws.com/key-name). Amazon S3 호스팅 가상 방식 액세스에 대한 자세한 내용은 [가상 호스팅 방식 요청을 참조](#)하세요.

MediaPackage 버전 2용 SigV4 작업

MediaPackage v2용 서명 버전 4(SigV4)는 HTTP를 통해 MediaPackage v2에 대한 요청을 인증하는 데 사용되는 서명 프로토콜입니다. MediaPackage v2용 SigV4를 사용하는 경우 MediaTailor는 오리지널로 사용되는 MediaPackage v2 엔드포인트에 대한 HTTP 요청에 서명된 권한 부여 헤더를 포함합니다. 서명된 권한 부여 헤더가 유효하면 오리지널이 요청을 이행합니다. 유효하지 않으면 요청이 실패합니다.

MediaPackage v2용 SigV4에 대한 일반적인 정보는 MediaPackage v2 API 참조의 [요청 인증\(AWS 서명 버전 4\)](#) 주제를 참조하세요.

요구 사항

소스 위치에 대해 MediaPackage v2 인증을 위해 SigV4를 활성화하는 경우 다음 요구 사항을 충족해야 합니다.

- 엔드포인트의 오리지널 액세스 정책에서 mediatailor.amazonaws.com 보안 주체 액세스 권한을 부여하여 MediaTailor가 MediaPackage v2 엔드포인트에 액세스하도록 허용해야 합니다.
- MediaTailor 소스 위치 기본 URL은 MediaPackage v2 엔드포인트여야 합니다.
- API 호출자는 MediaTailor 소스 패키징 구성에서 참조하는 모든 다변량 재생 목록을 읽을 수 있는 mediapackagev2:GetObject IAM 권한이 있어야 합니다. MediaTailor

AWS Secrets Manager 액세스 토큰 인증 작업

MediaTailor는 Secrets Manager 액세스 토큰 인증을 지원합니다. AWS Secrets Manager 액세스 토큰 인증을 통해 MediaTailor는 AWS Key Management Service (AWS KMS) 고객 관리형 키와 사용자가 생성, 소유 및 관리하는 보안 암호를 사용하여 오리 AWS Secrets Manager 진에 대한 요청을 인증합니다.

이 섹션에서는 Secrets Manager 액세스 토큰 인증의 작동 방식을 설명하고 Secrets Manager 액세스 토큰 인증을 구성하는 방법에 대한 step-by-step 정보를 제공합니다. 에서 Secrets Manager 액세스 토큰 인증을 AWS Management Console 사용하거나 AWS APIs.

주제

- [AWS Secrets Manager 액세스 토큰 인증 구성](#)
- [CDN 인증을 사용하는 MediaPackage 엔드포인트와 통합](#)
- [MediaTailor Secrets Manager 액세스 토큰 인증 작동 방식](#)

AWS Secrets Manager 액세스 토큰 인증 구성

AWS Secrets Manager 액세스 토큰 인증을 사용하려면 다음 단계를 수행합니다.

1. [고객 관리형 AWS Key Management Service 키를 생성합니다.](#)
2. [AWS Secrets Manager 보안 암호를 생성합니다.](#) 보안 암호에는 액세스 토큰이 포함되어 있으며, 이 토큰은 Secrets Manager에 암호화된 보안 암호 값으로 저장됩니다. MediaTailor는 AWS KMS 고객 관리형 키를 사용하여 보안 암호 값을 해독합니다.
3. Secrets Manager 액세스 토큰 인증을 사용하도록 AWS Elemental MediaTailor 소스 위치를 구성합니다.

다음 섹션에서는 AWS Secrets Manager 액세스 토큰 인증을 구성하는 방법에 대한 step-by-step 지침을 제공합니다.

주제

- [1단계: AWS KMS 대칭 고객 관리형 키 생성](#)
- [2단계: AWS Secrets Manager 보안 암호 생성](#)
- [3단계: 액세스 토큰 인증을 사용하여 MediaTailor 소스 위치 구성](#)

1단계: AWS KMS 대칭 고객 관리형 키 생성

AWS Secrets Manager 를 사용하여 보안 암호에 SecretString 저장된 형태의 액세스 토큰을 저장합니다. 는 사용자가 생성, 소유 및 관리하는 AWS KMS 대칭 고객 관리형 키를 사용하여 암호화 SecretString됩니다. MediaTailor는 대칭 고객 관리형 키를 사용하여 권한 부여를 통해 보안 암호에 쉽게 액세스하고 보안 암호 값을 암호화 및 해독합니다.

고객 관리형 키를 사용하면 다음과 같은 작업을 수행할 수 있습니다.

- 키 정책 수립 및 유지
- IAM 정책 및 권한 부여 수립 및 유지
- 키 정책 활성화 및 비활성화
- 암호화 키 구성 요소 교체
- 태그 추가

Secrets Manager가 AWS KMS 를 사용하여 보안 암호를 보호하는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의가 [AWS Secrets Manager 사용하는 방법을 AWS KMS](#) 참조하세요.

고객 관리형 키에 대한 자세한 내용은 AWS Key Management Service 개발자 가이드의 [고객 관리형 키](#)를 참조하세요.

Note

AWS KMS 고객 관리형 키 사용에 요금이 적용됩니다. 요금에 대한 자세한 내용은 [AWS Key Management Service 요금](#) 페이지를 참조하세요.

를 사용하거나 API를 AWS Management Console 사용하여 프로그래밍 방식으로 AWS KMS 대칭 고객 관리형 키를 생성할 수 있습니다. AWS KMS APIs

대칭 고객 관리형 키를 만들려면

개발자 안내서의 [대칭 고객 관리형 키 생성](#) 단계를 따릅니다. AWS Key Management Service

에서 필요한 Amazon 리소스 이름(ARN) 키를 기록해 둡니다 [2단계: AWS Secrets Manager 보안 암호 생성](#).

암호화 컨텍스트

암호화 컨텍스트는 데이터에 대한 추가 컨텍스트 정보를 포함하는 선택적 키-값 페어 세트입니다.

Secrets Manager에는 [암호화하고 해독할 때 암호화 컨텍스트](#)가 포함됩니다 SecretString. 암호화 컨텍스트에는 암호화를 특정 보안 암호로 제한하는 보안 암호 ARN이 포함됩니다. 추가된 보안 조치로 MediaTailor는 사용자를 대신하여 AWS KMS 권한 부여를 생성합니다. MediaTailor는 Secrets Manager 암호화 컨텍스트에 포함된 보안 암호 ARN과 SecretString 연결된 만 해독하도록 허용하는 [GrantConstraints](#) 작업을 적용합니다.

Secrets Manager가 암호화 컨텍스트를 사용하는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 [암호화 컨텍스트](#) 주제를 참조하세요.

키 정책 설정

키 정책에서는 고객 관리형 키에 대한 액세스를 제어합니다. 모든 고객 관리형 키에는 키를 사용할 수 있는 사람과 키를 사용하는 방법을 결정하는 문장이 포함된 정확히 하나의 키 정책이 있어야 합니다. 고객 관리형 키를 생성할 때 기본 키 정책을 사용할 수 있습니다. 자세한 내용은 AWS Key Management Service 개발자 가이드의 [Authentication and access control for AWS KMS](#)를 참조하세요.

MediaTailor 소스 위치 리소스와 함께 고객 관리형 키를 사용하려면 [CreateSourceLocation](#) 또는 [UpdateSourceLocation](#)을 호출하는 IAM 보안 주체에 다음 API 작업을 사용할 수 있는 권한을 부여해야 합니다.

- kms:CreateGrant - 고객 관리형 키에 권한 부여를 추가합니다. MediaTailor는 키를 사용하여 액세스 토큰 인증으로 구성된 소스 위치를 생성하거나 업데이트할 수 있도록 고객 관리형 키에 대한 권한 부여를 생성합니다. [에서 권한 부여 AWS KMS](#)사용에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서를 참조하세요.

이렇게 하면 MediaTailor가 다음을 수행할 수 있습니다.

- [GetSecretValue](#)를 호출할 때 Secrets Manager 보안 암호를 성공적으로 검색할 수 Decrypt 있도록 호출합니다.
- 소스 위치가 삭제되거나 보안 암호에 대한 액세스가 취소되면 호출RetireGrant하여 권한 부여를 사용 중지합니다.

다음은 MediaTailor에 추가할 수 있는 정책 설명의 예입니다.

```
{
```

```

    "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"
        }
    }
}

```

정책에서 권한을 지정하고 키 액세스 문제를 해결하는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 [의 권한 부여 AWS KMS](#)를 참조하세요.

2단계: AWS Secrets Manager 보안 암호 생성

Secrets Manager를 사용하여 AWS KMS 고객 관리형 키로 SecretString 암호화된 형식으로 액세스 토큰을 저장합니다. MediaTailor는 키를 사용하여를 복호화합니다SecretString. Secrets Manager가 AWS KMS 를 사용하여 보안 암호를 보호하는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의가 [AWS Secrets Manager 사용하는 방법을 AWS KMS](#) 참조하세요.

를 소스 위치 오리진 AWS Elemental MediaPackage 으로 사용하고 MediaTailor Secrets Manager 액세스 토큰 인증을 사용하려는 경우 절차를 따릅니다[the section called “CDN 인증을 사용하는 MediaPackage 엔드포인트와 통합”](#).

를 사용하거나 Secrets Manager API를 AWS Management Console 사용하여 프로그래밍 방식으로 Secrets Manager 보안 암호를 생성할 수 있습니다. APIs

보안 암호 생성

AWS Secrets Manager 사용 설명서의 [를 사용하여 보안 암호 생성 및 관리 AWS Secrets Manager](#) 단계를 따릅니다.

보안 암호를 생성할 때 다음 사항을 고려해야 합니다.

- [KmsKeyId](#)는 1단계에서 생성한 고객 관리형 키의 키 ARN이어야 합니다.

- [SecretString](#)을 제공해야 합니다. 는 액세스 토큰이 포함된 키와 값을 포함하는 유효한 JSON 객체 SecretString여야 합니다. 예: {"MyAccessTokenIdentifier":"112233445566"}. 값은 8~128자여야 합니다.

액세스 토큰 인증으로 소스 위치를 구성할 때 SecretString 키를 지정합니다. MediaTailor는 키를 사용하여 저장된 액세스 토큰을 조회하고 검색합니다 SecretString.

보안 암호 ARN과 SecretString 키를 기록해 둡니다. 액세스 토큰 인증을 사용하도록 소스 위치를 구성할 때 이를 사용합니다.

리소스 기반 보안 암호 정책 연결

MediaTailor가 보안 암호 값에 액세스하도록 하려면 리소스 기반 정책을 보안 암호에 연결해야 합니다. 자세한 내용은 AWS Secrets Manager 사용 설명서의 [Secrets Manager 보안 암호에 관한 정책 연결을](#) 참조하세요.

다음은 MediaTailor에 추가할 수 있는 정책 설명 예제입니다.

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"
    }
  ]
}
```

3단계: 액세스 토큰 인증을 사용하여 MediaTailor 소스 위치 구성

를 사용하거나 MediaTailor API를 사용하여 AWS Management Console 프로그래밍 방식으로 Secrets Manager 액세스 토큰 인증을 구성할 수 있습니다. APIs

Secrets Manager 액세스 토큰 인증을 사용하여 소스 위치를 구성하려면

AWS Elemental MediaTailor 사용 설명서의 [Access configuration](#)에 대한 단계를 따릅니다.

CDN 인증을 사용하는 MediaPackage 엔드포인트와 통합

를 소스 위치 오리진 AWS Elemental MediaPackage 으로 사용하는 경우 MediaTailor는 CDN 인증을 사용하는 MediaPackage 엔드포인트와 통합할 수 있습니다.

CDN 권한 부여를 사용하는 MediaPackage 엔드포인트와 통합하려면 다음 절차를 사용합니다.

MediaPackage와 통합하려면

1. 아직 [CDN 권한 부여를 설정하지 않은 경우 사용 설명서](#) AWS Elemental MediaPackage 의 CDN 권한 부여 설정 단계를 완료합니다.
2. [the section called “1단계: AWS KMS 대칭 고객 관리형 키 생성”](#)의 절차를 완료합니다.
3. MediaPackage CDN 권한 부여를 설정할 때 생성한 보안 암호를 수정합니다. 다음 값으로 보안 암호를 수정합니다.
 - 에서 생성한 고객 관리형 키 ARNKmsKeyId으로 업데이트합니다 [the section called “1단계: AWS KMS 대칭 고객 관리형 키 생성”](#).
 - (선택 사항)의 경우 UUID를 새 값으로 SecretString교체하거나와 같은 표준 JSON 형식의 키 및 값 페어인 경우 기존의 암호화된 보안 암호를 사용할 수 있습니다 {"MediaPackageCDNIdentifier": "112233445566778899"}.
4. [the section called “리소스 기반 보안 암호 정책 연결”](#)의 단계를 수행합니다.
5. [the section called “3단계: 액세스 토큰 인증을 사용하여 MediaTailor 소스 위치 구성”](#)의 단계를 수행합니다.

MediaTailor Secrets Manager 액세스 토큰 인증 작동 방식

액세스 토큰 인증을 사용하도록 소스 위치를 생성하거나 업데이트한 후 MediaTailor는 오리진에서 소스 콘텐츠 매니페스트를 요청할 때 HTTP 헤더에 액세스 토큰을 포함합니다.

다음은 MediaTailor가 소스 위치 오리진 인증에 Secrets Manager 액세스 토큰 인증을 사용하는 방법에 대한 개요입니다.

1. 액세스 토큰 인증을 사용하는 MediaTailor 소스 위치를 생성하거나 업데이트하면 MediaTailor는 Secrets Manager에 [DescribeSecret](#) 요청을 보내 보안 암호와 연결된 AWS KMS 키를 확인합니다. 소스 위치 액세스 구성에 보안 암호 ARN을 포함합니다.
2. MediaTailor는 고객 관리형 키에 대한 [권한 부여](#)를 생성하므로 MediaTailor는 키를 사용하여 SecretString에 저장된 액세스 토큰에 액세스하고 해독할 수 있습니다. 권한 부여 이름은 `입니 다MediaTailor-SourceLocation-your AWS ## ID-source location name`.

언제든지 권한 부여에 대한 액세스를 취소하거나 고객 관리형 키에 대한 MediaTailor의 액세스를 제거할 수 있습니다. 자세한 내용은 API 참조의 [RevokeGrant](#)를 참조하세요. AWS Key Management Service

3. VOD 소스가 생성 또는 업데이트되거나 프로그램에서 사용되는 경우 MediaTailor는 소스 위치에 HTTP 요청을 보내 소스 위치의 VOD 소스와 연결된 소스 콘텐츠 매니페스트를 검색합니다. VOD 소스가 액세스 토큰이 구성된 소스 위치와 연결된 경우 요청은 액세스 토큰을 HTTP 헤더 값으로 포함합니다.

VOD 소스 작업

VOD 소스는 소스 위치에 추가하는 비디오 또는 팟캐스트 에피소드와 같은 단일 콘텐츠를 나타냅니다. 하나 이상의 VOD 소스를 소스 위치에 추가한 다음 채널을 생성한 후 각 VOD 소스를 프로그램에 연결합니다.

각 VOD 소스에는 패키지 구성이 하나 이상 있어야 합니다. 패키지 구성은 VOD 소스의 패키지 형식, 매니페스트 위치 및 소스 그룹을 지정합니다. 채널을 생성할 때 패키지 구성의 소스 그룹을 사용하여 채널에 해당 출력을 생성합니다. 예를 들어 소스가 HLS와 DASH의 두 가지 형식으로 패키징된 경우 DASH용과 HLS용의 두 가지 패키지 구성을 생성합니다. 그런 다음 각 패키지 구성에 대해 하나씩 두 개의 채널 출력을 생성합니다. 각 채널 출력은 재생 요청에 사용되는 엔드포인트를 제공합니다. 따라서 앞의 예제를 사용하여 채널은 HLS 재생 요청에 대한 엔드포인트와 DASH 재생 요청에 대한 엔드포인트를 제공합니다.

매니페스트의 광고 마커 오프셋을 자동으로 감지하려면 각 광고 마커가 모든 패키지 구성에서 동일한 오프셋에 나타나고 지속 시간이 0이어야 합니다. HLS의 경우 MediaTailor는 DATERANGE 및 EXT-X-CUE-OUT 태그를 감지합니다. DASH의 경우 HLS는 각 태그 내에서 첫 번째 이벤트 EventStream 태그를 감지합니다.

다음 예제에서는 지속 시간이 0.0인 DATERANGE 태그로 인해 12,000ms의 오프셋에서 광고 중단 기회가 감지됩니다. 오프셋이 0ms인 첫 번째 DATERANGE 태그는 지속 시간이 10.0이므로 감지되지 않습니다.

```
#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
```

다음 예제에서는 EXT-X-CUE-OUT 태그의 지속 시간이 0이고 EXT-X-CUE-IN 태그가 바로 이어지기 때문에 0ms의 오프셋에서 광고 중단 기회가 감지됩니다. 두 번째 EXT-X-CUE-OUT/EXT-X-CUE-IN 페어는 지속 시간이 10이므로 감지되지 않습니다.

```
#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
```

다음 예제에서는의 첫 번째 이벤트가 PT0.000S부터 시작되는 기간에 EventStream 발생하기 때문에 0ms의 오프셋에서 광고 중단 기회가 감지됩니다. Event의 두 번째 EventStream는 감지되지 않습니다.

```
<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>

```

소스 위치에 VOD 소스 추가

다음 절차에서는 소스 위치에 VOD 소스를 추가하고 MediaTailor 콘솔을 사용하여 패키지 구성을 설정하는 방법을 설명합니다. MediaTailor API를 사용하여 VOD 소스를 추가하는 방법에 대한 자세한 내용은 API 참조의 [CreateVodSource](#)를 AWS Elemental MediaTailor 참조하세요.

Important

VOD 소스를 추가하기 전에 다음 요구 사항을 충족하는지 확인합니다.

- 소스 변형의 길이는 소스 매니페스트에 따라 모두 동일해야 합니다.
- 패키지 구성 내에서 각 소스는 동일한 수의 하위 스트림을 가져야 합니다.

이러한 요구 사항으로 인해 제목 또는 자동 ABR당을 지원하지 않습니다. 이러한 인코딩 방법은 다양한 매니페스트 길이와 하위 스트림을 생성할 수 있기 때문입니다.

인코딩된 소스가 이러한 요구 사항을 충족하는지 확인하려면 최소 세그먼트 길이가 포함된 인코딩 템플릿을 사용하는 것이 좋습니다.

소스 위치에 VOD 소스를 추가하려면

1. <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.
2. 탐색 창에서 채널 어셈블리 > 소스 위치를 선택합니다.
3. 소스 위치 창에서 [소스 위치를 생성하려면](#) 절차에서 생성한 소스 위치를 선택합니다.
4. VOD 소스 추가를 선택합니다.
5. VOD 소스 세부 정보에서 VOD 소스의 이름을 입력합니다.
 - 이름: my-example-video와 같은 VOD 소스의 식별자입니다.
6. 패키지 구성 > *source-group-name*에서 패키지 구성에 대한 정보를 입력합니다.

Note

소스의 패키지 구성은 모두 소스 매니페스트에 따라 기간이 동일해야 합니다. 또한 패키지 구성 내의 모든 소스에는 동일한 수의 하위 스트림이 있어야 합니다. 이러한 요구 사항을 충족하려면 자산에 인코딩 템플릿을 사용하는 것이 좋습니다. 최소 세그먼트 길이가 1초 인 인코딩 템플릿을 사용하는 것이 좋습니다. MediaTailor는 제목당 또는 자동 적응형 비트레이트 스트리밍(ABR)을 지원하지 않습니다. 이러한 인코딩 방법은 이러한 요구 사항을 위반하기 때문입니다.

- 소스 그룹: HLS-4k와 같이 패키지 구성을 설명하는 소스 그룹 이름을 입력합니다. 이 이름을 기록해 둡니다. 채널의 출력을 생성할 때 참조합니다. 자세한 내용은 [채널의 출력과 함께 소스 그룹 사용](#) 단원을 참조하십시오.
- 유형: 이 구성의 패키지 형식을 선택합니다. MediaTailor는 HLS 및 DASH를 지원합니다.
- 상대 경로: 소스 위치의 기본 HTTP URL에서 매니페스트까지의 상대 경로입니다. 예: /my/path/index.m3u8.

Note

MediaTailor는 상위 다변량 재생 목록에 포함된 닫힌 캡션과 하위 스트림을 모두 자동으로 가져옵니다. 각 소스 변환(DASH) 또는 변형 스트림(HLS)에 대해 별도의 패키지 구성을 생성할 필요가 없습니다.

패키지 구성에 대한 자세한 내용은 섹션을 참조하세요 [패키지 구성 사용](#).

7. VOD 소스 추가를 선택합니다.

VOD 소스를 더 추가하려면 절차의 4~7단계를 반복합니다.

라이브 소스 작업

라이브 소스는 소스 위치에 추가하는 라이브 축구 게임 또는 뉴스 방송과 같은 단일 라이브 스트림을 나타냅니다. 채널을 생성한 후 소스 위치에 하나 이상의 라이브 소스를 추가한 다음 각 라이브 소스를 프로그램에 연결합니다.

MediaTailor는 다음과 같은 유형의 선형 채널 어셈블리를 지원합니다.

- VOD-to-live 콘텐츠가 포함된 채널의 VOD 소스
- VODlive-to-live 콘텐츠와 VOD-to-live 소스

VOD-to-live 콘텐츠의 예로는 VOD 자산 라이브러리를 라이브 스트림으로 어셈블하는 채널이 있습니다. VODlive-to-live 콘텐츠와 VOD-to-live 콘텐츠의 한 가지 예는 야간 뉴스 이벤트 또는 사전 예약된 라이브 스포츠 이벤트를 제외하고 대부분 VOD 콘텐츠를 표시하는 채널입니다. VODlive-to-live 콘텐츠와 VOD-to-live 콘텐츠의 또 다른 예는 오리지널이 시간대에 따라 달라지는 모든 live-to-live 채널입니다.

라이브 소스를 사용하여 대부분 국가 프로그래밍을 표시하지만 지역 프로그래밍 재정의의 포함하고 VOD 콘텐츠가 혼합된 리전 채널을 설정할 수 있습니다. 이렇게 하려면 국가 콘텐츠에 대해 인코더/패키저 페어 하나를 실행한 다음 해당 리전이 활성화되면 리전 인코더를 실행합니다. 그런 다음 각각 고유한 일정이 있는 리전 채널 어셈블리 채널을 생성합니다. 이렇게 하면 뷰어가 필요에 따라 앞으로 전환할 수 있습니다. 이 설정은 인코딩/패키징 비용을 최소화하는 데 도움이 됩니다.

각 라이브 소스에는 하나 이상의 패키지 구성이 있어야 합니다. 패키지 구성은 라이브 소스의 패키지 형식, 매니페스트 위치 및 소스 그룹을 지정합니다. 채널을 생성할 때 패키지 구성의 소스 그룹을 사용하여 채널에 해당 출력을 생성합니다. 예를 들어 소스가 HLS와 DASH의 두 가지 형식으로 패키징된 경우 DASH용과 HLS용의 두 가지 패키지 구성을 생성합니다. 그런 다음 각 패키지 구성에 대해 하나씩 두 개의 채널 출력을 생성합니다. 각 채널 출력은 재생 요청에 사용되는 엔드포인트를 제공합니다. 이 예제에서 채널은 HLS 재생 요청에 대한 엔드포인트와 DASH 재생 요청에 대한 엔드포인트를 제공합니다.

라이브 소스 사용에 대한 일반 요구 사항

라이브 소스를 사용하는 경우 콘텐츠는 다음 일반 요구 사항에 부합해야 합니다.

- HLS 라이브 소스 - 매니페스트 창의 첫 번째 세그먼트와 모든 불연속성에 #EXT-X-PROGRAM-DATE-TIME 태그를 제공해야 합니다.
- HLS - 광고 마커를 로 구성해야 합니다 DATERANGE.
- 소스 매니페스트 기간 - MediaTailor 채널 어셈블리 채널의 매니페스트 기간보다 기간이 긴 매니페스트 기간을 사용하는 것이 좋습니다. 채널 어셈블리 채널의 매니페스트 기간보다 30초 이상 긴 매니페스트 기간 사용을 고려하는 것이 좋습니다.
- 대상 기간을 기존 소스의 기간과 일치시킵니다.
- 하위 재생 목록 수를 기존 소스의 재생 목록 수와 일치하도록 설정합니다.

구성

라이브 소스 워크플로의 일부로 다른 AWS Elemental 미디어 서비스를 사용하는 경우 MediaPackage 구성을 설정할 때 모범 사례를 따르는 것이 좋습니다. 다음 표에서는 사용하는 스트리밍 표준에 따라 MediaPackage 설정을 구성하는 방법을 설명합니다.

라이브 소스에 대한 MediaPackage 설정

표준	설정	값	필요성	참고
HLS	엔드포인트 유형	Apple HLS	CMAF를 사용하지 않는 한 필수	HLS ts AWS Elemental MediaConvert 작업을 일치시키려면
HLS	엔드포인트 유형	CMAF	Apple HLS를 사용하지 않는 한 필수	HLS mp4 AWS Elemental MediaConvert 작업을 일치시키려면
HLS	ProgramDateIntervalSeconds	1	필수	불연속성이 있을 때 재생 문제를 방지하려면 모든 세그먼트 #EXT-X-PROGRAM-DATE-TIME 에

표준	설정	값	필요성	참고
				를 지정해야 합니다.
HLS	PlaylistWindowSeconds	채널 어셈블리 매니페스트 기간보다 30초 더 길니다.	필수	
HLS	AdMarkers	DATERANGE	광고 마커를 통과할 때 필요합니다.	
HLS	IncludeIframeOnlyStream	비활성화됨	권장	
DASH	ManifestLayout	FULL	권장	
DASH	SegmentTemplateFormat	NUMBER_WITH_TIMELINE 또는 TIME_WITH_TIMELINE	권장	NUMBER_WITH_DURATION 는 지원되지 않습니다.
DASH	ManifestWindowSeconds	채널 어셈블리 매니페스트 기간보다 30초 더 길니다.	필수	
DASH	PeriodTriggers	ADS	광고 마커를 통과할 때 필요합니다.	

소스 위치에 라이브 소스 추가

다음 절차에서는 MediaTailor 콘솔을 사용하여 소스 위치에 라이브 소스를 추가하고 패키지 구성을 설정하는 방법을 설명합니다. MediaTailor API를 사용하여 라이브 소스를 추가하는 방법에 대한 자세한 내용은 API 참조의 [CreateLiveSource](#)를 AWS Elemental MediaTailor 참조하세요.

Important

라이브 소스를 추가하기 전에 패키지 구성 내에서 각 소스의 하위 스트림 수가 동일한지 확인합니다.

소스 위치에 라이브 소스를 추가하려면

1. <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.
2. 탐색 창에서 채널 어셈블리 > 소스 위치를 선택합니다.
3. 소스 위치 창에서 [소스 위치를 생성하려면](#) 절차에서 생성한 소스 위치를 선택합니다.
4. 라이브 소스 탭에서 라이브 소스 추가를 선택합니다.
5. 라이브 소스 세부 정보에서 라이브 소스의 이름을 입력합니다.
 - 이름: my-example-video와 같은 라이브 소스의 식별자입니다.
6. 패키지 구성 > *source-group-name*에서 패키지 구성에 대한 정보를 입력합니다.

Note

패키지 구성 내에서 모든 VOD 소스와 라이브 소스의 하위 스트림 수는 동일해야 합니다. 소스 스트림을 동일한 방식으로 구성하는 것이 좋습니다.

- 소스 그룹: HLS-4k와 같이이 패키지 구성을 설명하는 소스 그룹 이름을 입력합니다. 이 이름을 기록해 둡니다. 채널의 출력을 생성할 때 참조합니다. 자세한 내용은 [채널의 출력과 함께 소스 그룹 사용](#) 단원을 참조하십시오.
- 유형:이 구성의 패키지 형식을 선택합니다. MediaTailor는 HLS 및 DASH를 지원합니다.
- 상대 경로: 소스 위치의 기본 HTTP URL에서 매니페스트까지의 상대 경로입니다. 예: /my/path/index.m3u8.

Note

MediaTailor는 상위 다변량 재생 목록에 포함된 닫힌 캡션과 하위 스트림을 모두 자동으로 가져옵니다. 각 소스 변환(DASH) 또는 변형 스트림(HLS)에 대해 별도의 패키지 구성을 생성할 필요가 없습니다.

패키지 구성에 대한 자세한 내용은 섹션을 참조하세요 [패키지 구성 사용](#).

7. 라이브 소스 추가를 선택합니다.

라이브 소스를 더 추가하려면 절차의 4~6단계를 반복합니다.

패키지 구성 사용

패키지 구성은 다양한 디바이스에서 재생하는 데 필요한 다양한 패키징 특성을 포함하는 소스를 나타냅니다. 예를 들어 DRM을 사용하는 HLS, 세그먼트 타임라인 주소 지정을 사용하는 DASH, CMAF 세그먼트를 사용하는 HLS 등 세 가지 패키지 형식이 있는 소스가 있을 수 있습니다.

채널 어셈블리는 소스를 다시 패키징하지 않습니다. 지정된 소스에 대해 여러 패키지 형식을 포함하려면 각 패키지 형식을 소스 위치에서 사용할 수 있도록 하고 각 패키지 형식의 경로를 지정해야 합니다.

각 패키지 구성 객체에는 다음이 포함되어야 합니다.

- 상대 경로 - 소스 위치를 기준으로 소스의 패키지 형식에 대한 전체 경로입니다. 예: /my/path/index.m3u8.
- 소스 그룹 - 패키지 구성을 채널의 출력과 연결하는 데 사용되는 소스 그룹의 이름입니다.
- 유형 - HLS 또는 DASH.

채널을 생성한 후에는 채널 출력에 사용할 각 소스 그룹도 선언해야 합니다.

매니페스트 캐싱

MediaTailor는 소스 재생 목록을 주기적으로 그리고 기회적으로 캐시하여 채널 어셈블리 성능과 신뢰성을 개선합니다. 캐시된 버전이 소스 위치의 오리진 버전에 비해 오래된 경우가 있습니다.

MediaTailor가 캐시된 버전의 소스를 새로 고치도록 하려면 [UpdateVodSource](#)를 호출합니다. 예를 들어 소스에서 임베디드 경로가 변경될 때 호출을 사용합니다. MediaTailor의 요청이 거의 없더라도 소스 위치에서 사용할 수 있는 소스의 up-to-date 버전을 항상 유지해야 합니다.

채널 작업

채널은 소스 매니페스트를 선형 스트림으로 어셈블합니다. 각 채널에는 패키지 구성에 해당하는 출력이 하나 이상 포함됩니다.

먼저 채널을 생성한 다음 프로그램을 생성하여 VOD 소스와 라이브 소스를 채널 일정에 추가합니다. 각 프로그램은 VOD 소스 또는 라이브 소스와 연결됩니다.

주제

- [MediaTailor 콘솔을 사용하여 채널 생성](#)
- [채널의 출력과 함께 소스 그룹 사용](#)
- [MediaTailor 콘솔을 사용하여 채널 삭제](#)

MediaTailor 콘솔을 사용하여 채널 생성

다음 절차에서는 MediaTailor 콘솔을 사용하여 채널을 생성하는 방법을 설명합니다.

채널 생성

1. <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.
2. 탐색 창에서 채널 어셈블리 > 채널을 선택합니다.
3. 탐색 모음에서 채널 생성을 선택합니다.
4. 채널 세부 정보에서 채널에 대한 세부 정보를 입력합니다.
 - 이름: 채널의 이름을 입력합니다.
 - 티어: 티어는 채널이 지원하는 기능과 채널 실행 비용을 결정합니다. 요금에 대한 자세한 내용은 [채널 어셈블리 요금 페이지](#)를 참조하세요. MediaTailor는 다음 계층을 지원합니다.
 - 기본 - 기본 계층은 선형 및 루프 재생 모드를 모두 지원하며 라이브 소스를 지원하지 않습니다.
 - 표준 - 표준 계층은 라이브 소스를 지원하며 선형 재생 모드가 필요합니다.

채널 세부 정보에서 표준을 선택하면 대상 세부 정보에서 대상을 정의할 수 있습니다. 이러한 대상은 기본 프로그램에 대한 audienceMedia를 생성할 때 programRules에 사용됩니다.

- 추가를 선택합니다.
- 텍스트 상자에 대상 이름을 입력합니다. 1~32자의 영숫자여야 합니다.
- 확인을 선택합니다.

- 다음을 선택합니다.
 - 재생 모드: 재생 모드는 채널의 재생 동작을 설정합니다. MediaTailor는 다음 재생 모드를 지원합니다.
 - 루프 - 일정의 프로그램은 무한 루프에서 back-to-back 재생됩니다. 일정에서 마지막 프로그램이 재생되면 재생은 첫 번째 프로그램으로 다시 루프됩니다. 재생은 채널을 중지할 때까지 계속 반복됩니다.
 - 선형 - 일정의 각 프로그램은 한 번 back-to-back.
5. 필러 슬레이트에서 슬레이트 위치를 참조하는 소스 위치 이름과 슬레이트로 사용할 VOD 소스 이름을 선택합니다. MediaTailor는 슬레이트를 사용하여 일정의 프로그램 간 간격을 채웁니다. 슬레이트 지속 시간이 프로그램 간 간격 지속 시간보다 짧으면 MediaTailor는 슬레이트를 루프합니다. 채널이 선형 재생 모드를 사용하는 경우 필러 슬레이트 필드를 로 구성해야 합니다. MediaTailor는 루프 재생 모드에 대한 필러 슬레이트를 지원하지 않습니다.
6. 다음을 선택합니다.
7. 프로그램 규칙에서 대상 세부 정보를 지정합니다.
8. 채널 세부 정보에서 표준을 선택하면 대상 세부 정보에서 대상을 정의할 수 있습니다. 이러한 대상은 기본 프로그램에 대한 audienceMedia를 생성할 때 programRules에 사용됩니다.
- 추가를 선택한 다음 텍스트 상자에 대상을 추가하고 확인을 선택합니다.

 Note

영숫자 32자를 초과하지 않는 이름을 입력합니다.

- 출력 유형: 채널의 스트리밍 형식을 선택합니다. DASH 및 HLS가 지원됩니다.
 - 소스 그룹:에 설명된 대로 패키지 구성에서 생성한 소스 그룹의 이름을 입력합니다 [소스 위치에 VOD 소스 추가](#).
9. 다음을 선택합니다.
10. 매니페스트 설정에서 매니페스트 설정에 대한 추가 정보를 입력합니다.
- 매니페스트 기간(초): 각 매니페스트에 포함된 초 단위의 기간입니다. 최소값은 30초이고 최대값은 3600초입니다.
 - 광고 마크업 유형(HLS 출력만 해당): VOD 프로그램 광고 시간에 표시되는 광고 태그의 유형입니다. MediaTailor가 EXT-X-DATERANGE 태그를 사용하여 VOD 프로그램에 광고를 삽입하도록 Daterange하려면 선택합니다. MediaTailor가 EXT-X-CUE-OUT 및 EXT-X-CUE-IN 태그를 사용하여 VOD 프로그램에 광고를 삽입하도록 Scte35 Enhanced하려면 선택합니다. 이러한 태

그 유형에 대한 자세한 내용은 섹션을 참조하세요 [광고 시간에 대한 SCTE-35 메시지](#). 라이브 워크플로의 경우 MediaTailor는 항상 DATERANGE 태그를 전달하며 선택한 광고 마크업 유형에 관계없이 항상 Scte35 태그를 전달하지 않습니다.

11. 여러 채널 출력을 구성하려면 출력에서 추가를 선택합니다. 그런 다음이 절차의 6단계와 7단계를 완료하여 출력에 대한 세부 정보를 구성합니다.
12. 다음을 선택합니다.
13. 채널 정책에서 채널의 IAM 정책 설정을 선택합니다.
 - 채널 정책 연결 안 함: 이 계정의 자격 증명에 액세스할 수 있는 사용자로만 재생을 제한합니다.
 - 사용자 지정 정책 연결: 자체 정책을 정의하고 원하는 만큼 액세스를 제한합니다.
 - 퍼블릭 정책 연결: 채널의 출력에 대한 모든 수신 클라이언트 요청을 수락합니다. MediaTailor 광고 삽입을 사용하려면이 옵션을 사용해야 합니다.
14. 다음을 선택합니다.
15. 검토 및 생성 창에서 설정을 검토합니다.
16. Create funnel(퍼널 생성)을 선택합니다.

Note

채널은 중지된 상태로 생성됩니다. MediaTailor 콘솔 또는 MediaTailor StartChannel API 로 채널을 시작할 때까지 채널이 활성화되지 않습니다.

채널의 출력과 함께 소스 그룹 사용

소스 그룹은 패키지 구성을 채널의 출력과 연결합니다. 소스에서 패키지 구성을 생성할 때 소스 그룹의 이름을 식별합니다. 그런 다음 채널에서 출력을 생성할 때 동일한 이름을 입력하여 출력을 패키지 구성과 연결합니다. 채널의 프로그램에 추가된 VOD 소스 및 라이브 소스는 출력에서 식별된 소스 그룹에 속해야 합니다.

예제:

- VOD 소스 1과 2에는 모두 소스 그룹이 있는 HLS, DASH, HLS-4k의 세 가지 패키지 구성이 있습니다.
- VOD 소스 3에는 소스 그룹 HLS와 DASH가 포함된 두 가지 패키지 구성이 있습니다.

채널 A에 소스 그룹 HLS 및 DASH가 있는 두 개의 출력이 있는 경우 채널 출력은 세 개의 VOD 소스를 모두 사용할 수 있습니다. VOD 소스 1, 2, 3에는 모두 소스 그룹 레이블 HLS 및 DASH가 있는 패키지 구성이 있기 때문입니다.

채널 B에 소스 그룹 HLS 및 HLS-4k가 있는 두 개의 출력이 있는 경우 VOD 소스 1 및 2를 사용할 수 있지만 3은 사용할 수 없습니다. 이는 VOD 소스 1과 2 모두 소스 그룹 레이블 HLS 및 HLS-4k가 있는 패키지 구성을 갖기 때문입니다.

채널 C에 소스 그룹 DASH가 있는 단일 출력이 있는 경우 세 개의 VOD 소스를 모두 사용할 수 있습니다. 세 VOD 소스에는 모두 DASH 소스 그룹이 포함된 패키지 구성이 있습니다.

MediaTailor 콘솔을 사용하여 채널 삭제

채널을 삭제하려면 다음 절차를 완료하세요.

채널을 삭제하려면

1. <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.
2. 탐색 창에서 채널 어셈블리 > 채널을 선택합니다.
3. 삭제할 채널을 선택합니다.
4. 채널이 실행 중인 경우 작업 드롭다운 메뉴에서 중지를 선택합니다. 채널을 삭제하려면 먼저 채널을 중지해야 합니다.
5. 채널이 중지되면 작업 드롭다운 메뉴에서 삭제를 선택합니다.

채널 일정에 프로그램 추가

각 프로그램에는 VOD 소스 또는 계정의 소스 위치에 속하는 라이브 소스가 포함되어 있습니다. 채널을 일정에 추가하여 채널 스트림에서 재생되는 순서를 제어합니다.

VOD 소스가 포함된 프로그램은 하나 이상의 광고 시간으로 구성할 수 있습니다. 각 광고 시간에는 소스 위치의 VOD 소스인 슬레이트가 포함되어 있습니다. 광고 시간을 생성하려면 오프셋의 슬레이트를 프로그램에 밀리초 단위로 추가합니다.

주제

- [MediaTailor 콘솔을 사용하여 채널 일정 내에서 프로그램 생성](#)
- [프로그램 규칙을 사용하여 대상 그룹 및 대체 콘텐츠 정의](#)
- [대상별 매니페스트 생성](#)

MediaTailor 콘솔을 사용하여 채널 일정 내에서 프로그램 생성

다음 절차에서는 MediaTailor 콘솔을 사용하여 채널 일정 내에 프로그램을 생성하는 방법을 설명합니다. 또한 선택 사항인 광고 시간을 구성하는 방법도 설명합니다. MediaTailor API를 사용하여 프로그램을 생성하는 방법에 대한 자세한 내용은 API 참조의 [CreateProgram](#)을 AWS Elemental MediaTailor 참조하세요.

프로그램을 추가하려면

1. <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.
2. 탐색 창에서 채널 어셈블리 > 채널을 선택합니다.
3. 채널 창에서 [채널 생성](#) 절차에서 생성한 채널을 선택합니다.
4. 프로그램 세부 정보에서 프로그램에 대한 세부 정보를 입력합니다.
 - 이름: 채널에 추가하는 프로그램의 이름입니다.
 - 소스 유형: 프로그램이 재생하는 소스 비디오 유형을 결정합니다. 이 옵션은 표준 채널에서만 사용할 수 있습니다.
 - VOD - 프로그램은 사전 녹화된 TV 에피소드와 같은 video-on-demand 소스를 재생합니다.
 - 라이브 - 프로그램은 라이브 뉴스 브로드캐스트와 같은 라이브 소스를 재생합니다.
 - 소스 위치 이름: MediaTailor가 프로그램과 연결하는 소스 위치입니다.
 - 기존 소스 위치 선택을 선택한 경우 소스 위치 선택 메뉴에서 소스 위치 이름을 선택합니다. 또는 이름으로 소스 위치를 검색합니다. 이는 소스 위치가 많은 경우에 유용합니다.
 - 소스 위치 이름 입력을 선택한 경우 이름으로 소스 위치를 검색합니다.
 - VOD 소스 이름: MediaTailor가 프로그램과 연결하는 VOD 소스의 이름입니다.
 - 기존 VOD 소스 선택을 선택한 경우 계정과 연결된 VOD 소스 목록에서 VOD 소스 이름을 선택합니다. 또는 이름으로 VOD 소스를 검색합니다. 이는 VOD 소스가 많은 경우에 유용합니다.
 - 이름으로 검색을 선택한 경우 이름으로 라이브 소스를 검색합니다.
 - 라이브 소스 이름: 프로그램과 연결할 라이브 소스의 이름입니다. 이 옵션은 라이브를 소스 유형으로 선택한 경우에만 사용할 수 있습니다.
 - 기존 소스 위치 선택을 선택한 경우 소스 위치 선택 메뉴에서 소스 위치 이름을 선택합니다. 또는 이름으로 소스 위치를 검색합니다. 이는 소스 위치가 많은 경우에 유용합니다.
 - 소스 위치 이름 입력을 선택한 경우 이름으로 소스 위치를 검색합니다.
 - VOD 소스 이름: MediaTailor가 프로그램과 연결하는 VOD 소스의 이름입니다.

- 기존 라이브 소스 선택을 선택한 경우 계정과 연결된 라이브 소스 목록에서 라이브 소스 이름을 선택합니다. 또는 이름으로 라이브 소스를 검색할 수 있습니다. 이는 라이브 소스가 많은 경우에 유용합니다.
 - 이름으로 검색을 선택한 경우 이름으로 라이브 소스를 검색합니다.
5. 다음을 선택하여 일정 구성 탭으로 이동합니다.
 6. 재생 구성에서 채널 일정에서 프로그램이 재생되는 시기를 정의합니다.
 - 기간 밀리초: 프로그램의 기간을 밀리초 단위로 정의합니다. 이 옵션은 라이브 소스를 사용하는 프로그램에만 사용할 수 있습니다.
 - 전환 유형: 일정에서 프로그램 간 전환을 정의합니다.
 - 상대: 프로그램은 일정의 다른 프로그램 앞이나 뒤에 재생됩니다. 이 옵션은 VOD 소스를 사용하는 프로그램에만 사용할 수 있습니다.
 - 절대: 프로그램이 특정 벽시계 시간에 재생됩니다. MediaTailor는 지정한 클럭 시간에 프로그램을 재생하기 위해 최선을 다합니다. MediaTailor는 이전 프로그램 또는 슬레이트 사이의 공통 세그먼트 경계에서 프로그램 재생을 시작합니다. 이 옵션은 선형을 사용하도록 구성된 채널에만 사용할 수 있습니다. 재생 모드: 재생 모드는 채널의 재생 동작을 설정합니다. MediaTailor는 다음 재생 모드를 지원합니다..
 - 프로그램 시작 시간: 절대 전환 유형의 경우 프로그램이 재생되도록 예약된 벽시계 시간입니다. 실행 중인 선형 채널에이 프로그램을 추가하는 경우 현재 시간으로부터 15분 이상의 시작 시간을 입력해야 합니다.
 - 상대 위치: 다른 프로그램을 기준으로 프로그램을 일정에 삽입할 위치를 선택합니다. 프로그램 전 또는 프로그램 후를 선택할 수 있습니다. 채널 일정의 첫 번째 프로그램인 경우에는 이 설정이 적용되지 않습니다.
 - 기존 프로그램 선택을 선택한 경우 기존 프로그램 사용 메뉴의 채널에서 재생하는 다음 100개 프로그램의 사전 정의된 목록에서 프로그램 이름을 선택합니다.
 - 이름으로 프로그램 검색을 선택한 경우 채널에 있는 기존 프로그램의 이름을 입력합니다.

프로그램에 광고 시간을 추가하려면 다음 단계를 계속 진행합니다. 광고 시간은 VOD 소스를 사용하는 프로그램에 대해서만 구성할 수 있습니다. 라이브 소스의 경우 DASH 매니페스트의 광고 중단과 EXT-X-DATERANGE 태그를 사용하는 HLS 매니페스트의 광고 중단은 자동으로 전달됩니다.

7. 다음을 선택하여 광고 시간 추가로 이동합니다.
8. 광고 시간 추가를 선택합니다. 광고 시간에서 광고 시간에 대한 설정을 구성합니다.

- 슬레이트 소스 위치 이름: 기존 소스 위치 선택을 선택하고이 작업의 앞부분에서 생성한 슬레이트가 저장되는 소스 위치를 선택합니다.
- VOD 소스 이름: 기존 VOD 소스 선택을 선택하고이 작업의 앞부분에서 추가한 슬레이트에 사용 중인 VOD 소스를 선택합니다. 슬레이트 지속 시간에 따라 광고 시간 지속 시간이 결정됩니다.
- 밀리초 단위 오프셋:이 값은 광고 시간 시작 시간을 프로그램 시작을 기준으로 한 오프셋으로 밀리초 단위로 결정합니다. VOD 소스 기간보다 짧고 프로그램의 VOD 소스 내 모든 트랙(모든 오디오, 비디오 및 닫힌 캡션 트랙)의 세그먼트 경계와 일치하는 값을 입력합니다. 그렇지 않으면 광고 시간을 건너뛵니다. 예를 들어 0을 입력하면 프로그램이 시작되기 전에 재생되는 롤링 전 광고 시간이 생성됩니다.
- 가용 번호: MediaTailor는 SCTE-35 사양의 섹션 9.7.3.1, [디지털 프로그램 삽입 대기열 메시지에](#) 정의된 `splice_insert.avail_num`대로이 값에 기록합니다. 기본값은 0입니다. 값은 0에서 256 사이여야 합니다.
- 예상 가용성: MediaTailor는 SCTE-35 사양의 섹션 9.7.3.1에 정의된 `splice_insert.avails_expected`대로이 값에 씁니다. 기본값은 0입니다. 값은 0에서 256 사이여야 합니다.
- 스포라이스 이벤트 ID: MediaTailor는 SCTE-35 사양의 섹션 9.7.3.1에 정의된 `splice_insert.splice_event_id`대로이 값에 기록합니다. 기본값은 1입니다.
- 고유 프로그램 ID: MediaTailor는 SCTE-35 사양의 섹션 9.7.3.1에 정의된 `splice_insert.unique_program_id`대로이 값에 씁니다. 기본값은 0입니다. 값은 0에서 256 사이여야 합니다.

9. 표준 선형 채널의 경우 다음을 선택하여 대체 미디어 설정으로 이동합니다.

MediaTailor를 사용하여 대체 미디어를 생성하는 방법에 대한 자세한 내용은 섹션을 참조하세요 [대체 미디어 생성](#).

MediaTailor를 사용하여 광고 시간을 개인화하는 방법에 대한 자세한 내용은 섹션을 참조하세요 [채널 스트림에 개인화된 광고 및 광고 시간 삽입](#).

10. 다음을 선택하여 검토 및 생성으로 이동합니다.

11. 프로그램 추가를 선택합니다.

MediaTailor를 사용하여 광고 시간을 개인화하는 방법에 대한 자세한 내용은 섹션을 참조하세요 [채널 스트림에 개인화된 광고 및 광고 시간 삽입](#).

12.

⚠ Important

채널 반복의 경우 다음 10분 이내에 예약된 프로그램의 프로그램 목록을 수정하면 다음 루프까지 편집이 명확하지 않습니다.

프로그램 세부 정보에서 프로그램에 대한 세부 정보를 입력합니다.

- 이름: 채널에 추가하는 프로그램의 이름입니다.
- 소스 유형: 프로그램이 재생하는 소스 유형을 결정합니다. 이 옵션은 표준 채널에서만 사용할 수 있습니다.
 - VOD - 프로그램은 사전 녹화된 TV 에피소드와 같은 VOD 소스를 재생합니다.
 - 라이브 - 프로그램은 라이브 뉴스 브로드캐스트와 같은 라이브 소스를 재생합니다.
- 소스 위치 이름: 프로그램과 연결할 소스 위치입니다.

기존 소스 위치 선택을 선택한 경우 소스 위치 선택 드롭다운 메뉴에서 소스 위치 이름을 선택합니다. 또는 이름으로 소스 위치를 검색할 수 있습니다. 이는 소스 위치가 많은 경우에 유용합니다.

소스 위치 이름 입력을 선택한 경우 이름으로 소스 위치를 검색합니다.

- VOD 소스 이름: 프로그램과 연결할 VOD 소스의 이름입니다.

기존 VOD 소스 선택을 선택한 경우 계정과 연결된 VOD 소스 목록에서 VOD 소스 이름을 선택합니다. 또는 이름으로 VOD 소스를 검색할 수 있습니다. 이는 VOD 소스가 많은 경우에 유용합니다.

이름으로 검색을 선택한 경우 이름으로 VOD 소스를 검색합니다.

- 라이브 소스 이름: 프로그램과 연결할 라이브 소스의 이름입니다. 이 옵션은 라이브를 소스 유형으로 선택한 경우에만 사용할 수 있습니다.

기존 라이브 소스 선택을 선택한 경우 계정과 연결된 라이브 소스 목록에서 라이브 소스 이름을 선택합니다. 또는 이름으로 라이브 소스를 검색할 수 있습니다. 이는 라이브 소스가 많은 경우에 유용합니다.

이름으로 검색을 선택한 경우 이름으로 라이브 소스를 검색합니다.

13. 재생 구성에서 채널 일정에서 프로그램이 재생되는 시기를 정의합니다.

- **밀리초 단위의 기간:** 프로그램의 기간을 밀리초 단위로 정의합니다. 이 옵션은 라이브 소스를 사용하는 프로그램에만 사용할 수 있습니다.
- **전환 유형:** 일정에서 프로그램에서 프로그램으로의 전환을 정의합니다.
- **상대 - 프로그램은 일정의 다른 프로그램 앞이나 뒤에 재생됩니다.** 이 옵션은 VOD 소스를 사용하는 프로그램에만 사용할 수 있습니다.
- **절대 - 프로그램은 특정 벽시계 시간에 재생됩니다.** MediaTailor는 지정한 클럭 시간에 프로그램을 재생하기 위해 최선을 다합니다. 이전 프로그램 또는 슬레이트 사이의 공통 세그먼트 경계에서 프로그램 재생을 시작합니다. 이 옵션은를 사용하도록 구성된 채널에서만 사용할 수 있습니다[linear playback mode](#).

 Note

절대 전환 유형에 대해 다음 동작에 유의하세요.

- 일정의 이전 프로그램에 벽시계 시간 이상으로 연장된 기간이 있는 경우 MediaTailor는 벽시계 시간과 가장 가까운 공통 세그먼트 경계에서 이전 프로그램을 잘라냅니다.
- 일정에 프로그램 간에 간격이 있는 경우 MediaTailor는를 재생합니다[filler slate](#). 슬레이트 지속 시간이 간격 지속 시간보다 짧으면 MediaTailor는 슬레이트를 루프합니다.

- **프로그램 시작 시간 - 절대 전환 유형의 경우 프로그램이 재생되도록 예약된 벽시계 시간입니다.** 실행 중인 선형 채널에이 프로그램을 추가하는 경우 현재 시간으로부터 15분 이상의 시작 시간을 입력해야 합니다.
- **상대 위치:** 다른 프로그램을 기준으로 프로그램을 일정에 삽입할 위치를 선택합니다. 프로그램 전 또는 프로그램 후를 선택할 수 있습니다. 채널 일정의 첫 번째 프로그램인 경우에는이 설정이 적용되지 않습니다.
- **상대 프로그램:** 이전 또는 이후에 새 프로그램을 삽입하는 데 사용할 프로그램의 이름입니다. 채널 일정의 첫 번째 프로그램인 경우에는이 설정이 적용되지 않습니다.

기존 프로그램 선택을 선택한 경우 기존 프로그램 사용 드롭다운 메뉴의 채널에서 재생하는 다음 100개 프로그램의 사전 정의된 목록에서 프로그램 이름을 선택합니다.

이름으로 프로그램 검색을 선택한 경우 채널에 있는 기존 프로그램의 이름을 입력합니다.

프로그램에 광고 시간을 추가하려면 다음 단계를 계속 진행합니다. 광고 시간은 VOD 소스를 사용하는 프로그램에 대해서만 구성할 수 있습니다. 라이브 소스의 경우 DASH 매니페스트의 광고 중단과 EXT-X-DATERANGE 태그를 사용하는 HLS 매니페스트의 광고 중단은 자동으로 전달됩니다.

14. 광고 시간 추가를 선택합니다. 광고 시간에서 광고 시간에 대한 설정을 구성합니다.

- 슬레이트 소스 위치 이름: 기존 소스 위치 선택을 선택하고이 자습서의 앞부분에서 생성한 슬레이트가 저장되는 소스 위치를 선택합니다.
- VOD 소스 이름: 기존 VOD 소스 선택을 선택하고이 자습서의 앞부분에서 추가한 슬레이트에 사용 중인 VOD 소스를 선택합니다. 슬레이트 지속 시간에 따라 광고 중단 지속 시간이 결정됩니다.
- 밀리초 단위 오프셋의 경우:이 값은 프로그램 시작을 기준으로 오프셋으로 광고 시간 시작 시간을 밀리초 단위로 결정합니다. VOD 소스 기간보다 짧고 프로그램의 VOD 소스 내 모든 트랙(모든 오디오, 비디오 및 닫힌 캡션 트랙)의 세그먼트 경계와 일치하는 값을 입력합니다. 그렇지 않으면 광고 시간을 건너뛰니다. 예를 들어 0을 입력하면 프로그램이 시작되기 전에 재생되는 롤링 전 광고 시간이 생성됩니다.

Note

MediaTailor가 VOD 소스 내에서 지속 시간이 0인 HLSEXT-X-CUE-OUT의 경우 또는, EventStream DASH의 경우와 같은DATERANGE 광고 마커를 감지하는 경우 드롭다운 메뉴에서 해당 광고 마커의 오프셋을 선택하여 광고 시간의 오프셋으로 사용할 수 있습니다. 광고 기회를 감지하려면 VOD 소스 내의 모든 패키지 구성에서 동일한 오프셋에 존재해야 하며 기간은 0이어야 합니다.

- 메시지 유형: SCTE-35 광고 삽입 유형입니다. SPLICE_INSERT 또는 TIME_SIGNAL을 선택합니다.
- SPLICE_INSERT: 스플라이스 삽입 파라미터를 사용하여 광고 시간에 대한 기본 메타데이터를 제공합니다.
- TIME_SIGNAL: 세분화 설명자를 사용하여 고급 메타데이터를 제공합니다. 메시지 유형 간의 차이점에 대한 자세한 내용은 섹션을 참조하세요 [the section called “광고 시간에 대한 SCTE-35 메시지”](#).

SPLICE_INSERT 메시지 유형의 경우:

- 가용 번호의 경우 SCTE-35 사양의 섹션 9.7.3.1에 정의된 splice_insert.avail_num대로 기록됩니다. 기본값은 0입니다. 값은 0 ~ 사이어야 합니다256.

- 예상 가용성의 경우 SCTE-35 사양의 섹션 9.7.3.1에 정의된 `splice_insert.avails_expected`대로에 기록됩니다. 기본값은 0입니다. 값은 0 ~ 사이여야 합니다256.
- 스플라이스 이벤트 ID의 경우 SCTE-35 사양의 섹션 9.7.3.1에 정의된 `splice_insert.splice_event_id`대로에 기록됩니다. 기본값은 1입니다.
- 고유 프로그램 ID의 경우 SCTE-35 사양의 섹션 9.7.3.1에 정의된 `splice_insert.unique_program_id`대로에 기록됩니다. 기본값은 0입니다. 값은 0 ~ 사이여야 합니다256.

TIME_SIGNAL 메시지 유형의 경우:

- 분할 이벤트 ID의 경우 SCTE-35 사양의 섹션 10.3.3.1에 정의된 `segmentation_descriptor.segmentation_event_id`대로에 기록됩니다. 기본값은 1입니다.
- 세분화 유형 ID의 경우 SCTE-35 사양의 섹션 10.3.3.1에 정의된 `segmentation_descriptor.segmentation_type_id`대로에 기록됩니다. 기본값은 48 (0x30)입니다. 값은 0 ~ 사이여야 합니다256.
- 분할 UPID의 경우 SCTE-35 사양의 섹션 10.3.3.1에 정의된 `segmentation_descriptor.segmentation_upid`대로에 기록됩니다. 값은 및 문자가 포함된 16진수 문자열이어야 합니다0-9A-F. 기본값은 빈 문자열입니다.
- 분할 UPID 유형의 경우 SCTE-35 사양의 섹션 10.3.3.1에 정의된 `segmentation_descriptor.segmentation_upid_type`대로에 기록됩니다. 기본값은 14 (0x0E)입니다. 값은 0 ~ 사이여야 합니다256.
- 세그먼트 번호의 경우 SCTE-35 사양의 섹션 10.3.3.1에 정의된 `segmentation_descriptor.segment_num`대로에 기록됩니다. 기본값은 0입니다. 값은 0 ~ 사이여야 합니다256.
- 예상되는 세그먼트의 경우 SCTE-35 사양의 섹션 10.3.3.1에 정의된 `segmentation_descriptor.segments_expected`대로에 기록됩니다. 기본값은 0입니다. 값은 0 ~ 사이여야 합니다256.

15. 프로그램 추가를 선택합니다.

MediaTailor를 사용하여 광고 시간을 개인화하는 방법에 대한 자세한 내용은 섹션을 참조하세요 [채널 스트림에 개인화된 광고 및 광고 시간 삽입](#).

Note

채널에 Enhanced SCTE35 광고 마크업 유형의 출력이 하나 이상 있는 경우 광고 브레이크 메타데이터를 제출할 수 있습니다. MediaTailor는 광고 시간을 위해 제출된 키-값 페어를 EXT-X-ASSET 태그에 기록합니다.

프로그램 규칙을 사용하여 대상 그룹 및 대체 콘텐츠 정의

프로그램 규칙을 사용하면 채널의 대상 그룹을 정의하고 해당 대상에 대해 재생할 대체 미디어를 지정할 수 있습니다. 하나 이상의 대체 콘텐츠 소스를 프로그램의 대상과 연결할 수 있습니다. 프로그램이 종료되면 추가 대체 미디어를 지정하지 않는 한 기본 대상 콘텐츠가 재생됩니다.

프로그램 규칙은 선형 재생 모드를 사용하는 STANDARD 티어 채널에서 사용할 수 있습니다. MediaTailor 채널은 모든 VOD 소스 및 라이브 소스에 대한 대체 미디어를 지원합니다.

사용 예제는 [Using program rules with AWS MediaTailor](#)를 참조하세요.

대상 정의

MediaTailor 채널을 구성할 때 대상을 하나씩 입력하여 채널의 대상을 정의합니다. MediaTailor 콘솔 또는 MediaTailor CreateChannel API를 통해이 작업을 수행할 수 있습니다. 각 대상은 1~32자의 영숫자여야 합니다. 대상에 제공된 값이 유효하지 않으면 요청이 실패합니다.

LINEAR 재생 모드에서는 STANDARD 티어 채널에서만 대상을 정의할 수 있습니다.

대상을 업데이트해야 하는 경우 MediaTailor 콘솔 또는 MediaTailor UpdateChannel API를 사용하여 업데이트할 수 있습니다.

ProgramRules 기능을 사용하는 경우 CreateProgram 또는 UpdateProgram 요청에 AudienceMedia 정의된 채널에 정의된 기존 대상이 포함되어 있는지 확인합니다.

대체 미디어 생성

다음 작업에서는 MediaTailor 콘솔을 사용하여 대체 미디어를 정의하는 방법을 설명합니다. MediaTailor API를 사용하여 대체 미디어를 정의하는 방법에 대한 자세한 내용은 API 참조의 섹션을 AWS Elemental MediaTailor 참조[CreateProgram](#)하세요.

새 프로그램에서 대체 미디어를 정의하려면:

1. <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.
2. 탐색 창에서 채널 어셈블리 > 채널을 선택합니다.
3. 대체 미디어를 추가할 채널 이름을 선택합니다.
4. 프로그램을 생성합니다. 자세한 내용은 [MediaTailor 콘솔을 사용하여 채널 일정 내에서 프로그램 생성](#) 단원을 참조하십시오.
5. 대체 미디어 구성:
 - 대상 상자에서 추가를 선택하여 대체 미디어를 정의할 대상을 선택합니다.
 - 대상 메뉴에서 채널에 정의된 대상을 선택합니다.
 - 대체 미디어 추가를 선택하여 프로그램에 대한 대체 미디어 정의를 시작합니다.
 - MediaTailor는 대체 미디어 1 상자를 생성합니다. MediaTailor가 프로그램에서 대체 미디어로 재생하는 첫 번째 콘텐츠입니다.
 - 대체 미디어 1 상자 내에서:
 - 소스 위치를 선택합니다.
 - VOD 또는 라이브 소스 유형을 선택합니다.

VOD의 경우

- 소스 유형으로 VOD를 선택합니다.
- (선택 사항) 클립 범위를 지정합니다. 대체 미디어 VOD 소스를 포함한 VOD 소스를 사용하면 재생할 VOD 소스의 일부를 지정하여 소스의 시작 및/또는 끝에서 클리핑할 수 있습니다. 지정 시작 및 종료 오프셋은 밀리초 단위입니다.
- (선택 사항) a)Add Ad Breaks. 이는 프로그램을 생성할 때와 동일한 방식으로 수행됩니다. 자세한 내용은 [MediaTailor 콘솔을 사용하여 채널 일정 내에서 프로그램 생성](#) 단원을 참조하십시오.

라이브의 경우

- 소스 유형으로 라이브를 선택합니다.
- 라이브 소스를 선택합니다.
- 이 라이브 소스가 시작해야 하는 wall-clock 시간의 시작 시간을 밀리초 단위로 입력합니다. 라이브 소스는 정의 중인 기본 프로그램의 기간 내에만 재생됩니다. 시작 시간이 기본 프로그램 시작 이전인 경우 기본 프로그램이 시작할 때까지 시작되지 않습니다. 시작 시간이 기본 프로그램이 종료된 후인 경우 MediaTailor는 라이브 소스를 재생하지 않습니다.
- 기간을 밀리초 단위로 입력합니다. 기간은 10분 이상이어야 합니다.

- 대체 미디어 추가를 다시 선택하여 대상에 대해 프로그램에 추가 대체 미디어를 추가할 수 있습니다. 그러면 대체 미디어 2라는 레이블이 지정된 다른 상자가 생성됩니다. 대상별로 프로그램당 최대 5개의 대체 미디어 소스를 지정할 수 있습니다.
- 원하는 모든 대상에 대한 대체 미디어 정의를 마치면 다음을 선택하고 프로그램 생성을 계속합니다.

자세한 내용은 [MediaTailor 콘솔을 사용하여 채널 일정 내에서 프로그램 생성](#) 단원을 참조하십시오.

Note

대체 미디어는 정의된 프로그램의 기간 동안만 재생됩니다. 모든 대체 콘텐츠가 기본 콘텐츠를 초과하면 MediaTailor는 이를 잘라냅니다. MediaTailor는 정의된 순서대로 대체 미디어를 재생합니다. 라이브 대체 미디어 시작 시간은 항상 우선하며 이전에 예약된 VOD 소스 또는 라이브 소스를 잘라냅니다. 대상에 대한 대체 미디어로 채워지지 않은 모든 채널 정의 필러 슬레이트로 채워집니다.

- 다른 대상에 대한 대상 미디어를 정의하려면 대상 옆의 추가를 다시 선택합니다. 새로 생성된 대상을 선택하고 대상 ID를 설정한 다음 위에 설명된 대로 대체 미디어를 추가합니다. 최대 5명의 대상이 하나의 프로그램에 대체 미디어를 사용할 수 있습니다.

대상별 매니페스트 생성

특정 대상에 대한 매니페스트를 검색하려면 `aws.mediatailor.channel.audienceId` 쿼리 파라미터를 사용합니다. 이 쿼리 파라미터는 CDN에서 동적으로 추가되거나 콘텐츠 또는 고객 관리 시스템에 대한 호출을 통해 추가될 수 있습니다. 지정된 재생 세션과 MediaTailor audienceId 외부의 연결을 유지해야 합니다. 그러면 기본 콘텐츠 대신 해당 대상에 대해 정의된 대체 미디어가 있는 대상별 매니페스트가 검색됩니다. 특정 대상에 대해 매니페스트가 요청되면 플레이어는 항상 동일한 대상 ID로 매니페스트를 요청해야 합니다. 그렇지 않으면 재생 오류가 발생할 수 있습니다.

채널에 없는 대상에 대해 요청이 이루어진 경우 MediaTailor는 404 오류를 반환합니다.

Example 대상에 대한 매니페스트 가져오기

```
https:// prefix>.channel-assembly.mediatailor.us-west-2.amazonaws.com/v1/channel/ExampleChannel/index_dash.mpd?aws.mediatailor.channel.audienceId=Seattle
```

채널 스트림에 개인화된 광고 및 광고 시간 삽입

MediaTailor를 사용하면 콘텐츠를 SCTE-35 마커로 조정하지 않고도 프로그램에 광고 시간을 삽입하여 채널 어셈블리 선형 스트림을 수익화할 수 있습니다. MediaTailor 광고 삽입 서비스 또는 서버 측 광고 삽입(SSAI)과 함께 채널 어셈블리를 사용할 수 있습니다.

다음 주제에서는 채널의 선형 스트림에 개인화된 광고 및 광고 시간을 삽입하는 방법을 보여줍니다.

주제

- [MediaTailor를 사용하여 광고 삽입 설정](#)
- [광고 시간에 대한 SCTE-35 메시지](#)

MediaTailor를 사용하여 광고 삽입 설정

채널의 스트림에 개인화된 광고를 삽입하려면 채널의 엔드포인트 URL이 콘텐츠 소스입니다 AWS Elemental MediaTailor. 이 가이드에서는 광고 삽입을 위해 MediaTailor를 설정하는 방법을 보여줍니다.

사전 조건

시작하기 전에 다음 요구 사항을 충족하는지 확인하세요.

- MediaTailor 광고 삽입을 위한 HLS 및 DASH 스트림을 준비합니다.
 - 콘텐츠를 스트림을 아직 준비하지 않은 경우 MediaTailor 시작하기 광고 삽입 주제 [2단계: 스트림 준비](#)의 섹션을 참조하세요.
- 광고 결정 서버(ADS)가 있어야 합니다.
- 프로그램에서 광고 중단 설정을 구성합니다. 자세한 내용은 [Configuring ad breaks for your program](#) 절차를 참조하세요.

채널 어셈블리와 MediaTailor 광고 삽입 사이에 콘텐츠 전송 네트워크(CDN)를 사용하는 것이 가장 좋습니다. MediaTailor 광고 삽입 서비스는 추가 오리진 요청을 생성할 수 있습니다. 따라서 채널 어셈블리에서 매니페스트를 프록시하도록 CDN을 구성한 다음 콘텐츠 소스 URLs에서 CDN 접두사 URL을 사용하는 것이 좋습니다.

광고 삽입을 위한 MediaTailor 구성

다음은 채널의 스트림에 개인화된 광고를 삽입할 수 있도록 MediaTailor 콘솔 설정을 구성하는 방법을 보여줍니다.

광고 삽입을 위해 MediaTailor를 구성하려면

1. <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.
2. 탐색 창에서 구성을 선택합니다.
3. 필수 설정에서 구성에 대한 기본 필수 정보를 입력합니다.
 - 이름: 구성의 이름입니다.
 - 콘텐츠 소스: 채널 출력의 재생 URL에서 파일 이름과 확장명을 뺀 값을 입력합니다. MediaTailor 구성에 대한 자세한 내용은 섹션을 참조하세요 [필수 설정](#).
 - 광고 결정 서버: ADS의 URL을 입력합니다.
4. 선택적으로 구성 별칭, 개인화 세부 정보 및 고급 설정을 구성할 수 있습니다. 이러한 설정에 대한 자세한 내용은 섹션을 참조하세요 [선택적 구성 설정](#).
5. 탐색 모음에서 구성 생성을 선택합니다.

광고 삽입을 위해 MediaTailor를 설정했으므로 이제 광고 시간을 설정할 수도 있습니다. 자세한 지침은 [MediaTailor 광고 삽입 시작하기](#) 섹션을 참조하세요.

광고 시간에 대한 SCTE-35 메시지

MediaTailor를 사용하면 소스 위치 및 VOD 소스 리소스를 기반으로 콘텐츠 채널을 생성할 수 있습니다. 그런 다음 채널 일정에 따라 각 프로그램에 대해 하나 이상의 광고 시간을 설정할 수 있습니다. SCTE-35 사양에 기반한 메시지를 사용하여 광고 중단에 대한 콘텐츠를 조절합니다. 예를 들어 SCTE-35 메시지를 사용하여 광고 시간에 대한 메타데이터를 제공할 수 있습니다. SCTE-35 사양에 대한 자세한 내용은 [디지털 프로그램 삽입 대기열 메시지를 참조하세요](#).

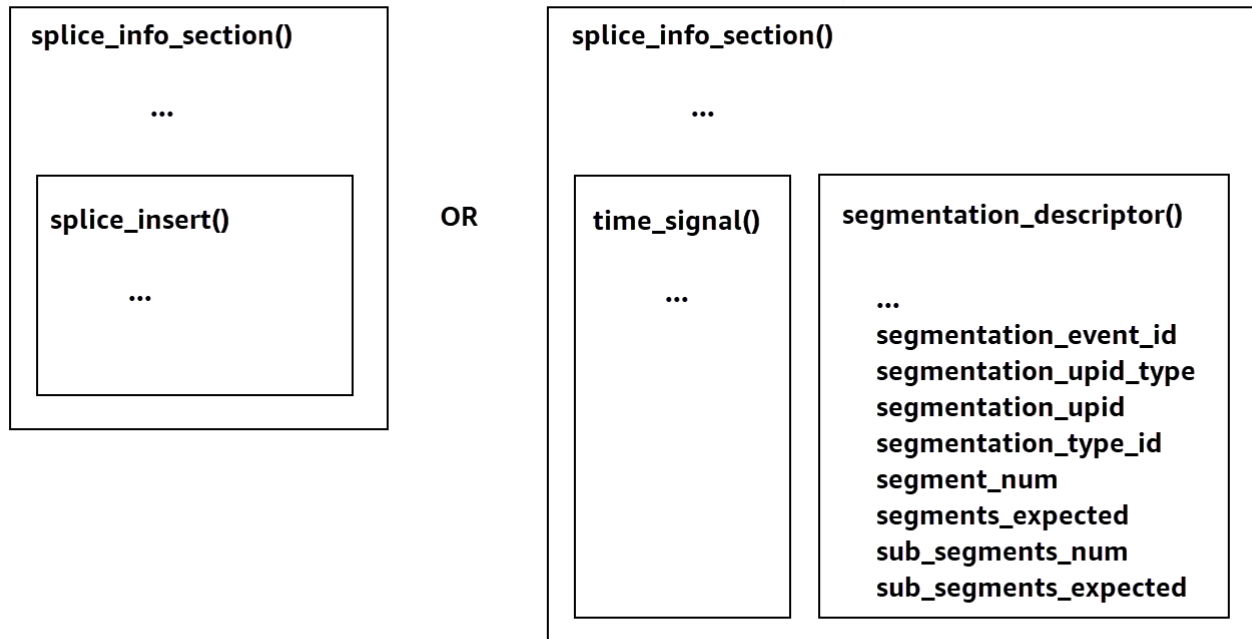
다음 두 가지 방법 중 하나로 광고 시간을 설정합니다.

- 메시지와 함께 time_signal SCTE-35 segmentation_descriptor 메시지 연결. 이 segmentation_descriptor 메시지에는 광고 시간에 대한 자세한 정보를 전달하는 콘텐츠 식별자와 같은 고급 메타데이터 필드가 포함되어 있습니다. MediaTailor는 (HLS) 또는 EXT-X-DATERANGE (EventStreamDASH) 광고 마커의 SCTE-35 데이터의 일부로 출력 매니페스트에 광고 메타데이터를 씁니다.

- 광고 시간에 대한 기본 메타데이터를 제공하는 `splice_insert` SCTE-35 메시지 연결.
- HLS:
 - 광고 마크업 유형이 인 경우 Daterange MediaTailor는 매니페스트에서 광고 시간을 `EXT-X-DATERANGE` 태그로 지정합니다.
 - 광고 마크업 유형이 인 경우 Scte35 Enhanced MediaTailor는 다음 태그를 사용하여 광고 시간을 지정합니다.
 - MediaTailor는 광고 슬레이트의 첫 번째 세그먼트 `EXT-X-CUE-OUT`에를 배치하여 콘텐츠에서 광고 시간까지의 컷을 나타냅니다. 여기에는와 같은 예상 광고 중단 기간이 포함됩니다 `EXT-X-CUE-OUT:Duration=30`.
 - `>EXT-X-ASSET`:이 태그는와 동일한 세그먼트에 나타나 `EXT-X-CUE-OUT`며 프로그램이 생성되거나 업데이트될 때 AdBreak에 제공된 광고 중단 메타데이터를 포함합니다. 항상를 포함합니다 CAID.
 - `EXT-0ATCLS-SCTE35`:이 태그는와 동일한 세그먼트에 나타나 `EXT-X-CUE-OUT`며 SCTE-35 메시지의 base64 인코딩 바이트를 포함합니다.
 - `EXT-X-CUE-OUT-CONT`:이 태그는 광고 슬레이트 내의 각 후속 세그먼트에 표시되며 기간 및 경과 시간 정보를 포함합니다. 또한 base64로 인코딩된 SCTE-35 메시지와 도 포함되어 있습니다 CAID.
 - `EXT-X-CUE-IN`:이 태그는 광고 시간이 끝난 후 콘텐츠의 첫 번째 세그먼트에 표시되며, 광고 시간에서 콘텐츠로의 컷백을 나타냅니다.

다음 그림은 SCTE-35 메시지를 사용하여 채널에서 광고 시간을 설정하는 두 가지 방법을 보여줍니다.

- `splice_insert()` 메시지를 사용하여 기본 메타데이터로 광고 시간을 설정합니다.
- `time_signal()` 메시지와 함께 메시지를 사용하여 더 자세한 메타데이터로 광고 시간을 `segmentation_descriptor()` 설정합니다.



사용에 대한 자세한 내용은 2022 SCTE-35 사양의 섹션 9.7.4, 디지털 프로그램 삽입 대기열 메시지를 `time_signal` 참조하세요. <https://webstore.ansi.org/Standards/SCTE/ANSISCTE352022>

광고 중단 정보는 출력 `splice_info_section` SCTE-35 데이터에 표시됩니다. MediaTailor를 사용하면 단일 `segmentation_descriptor` 메시지와 단일 `time_signal` 메시지를 함께 페어링할 수 있습니다.

Note

`segmentation_descriptor` 메시지를 보내는 경우 `time_signal` 메시지 유형의 일부로 보내야 합니다. `time_signal` 메시지에는 MediaTailor가 지정된 타임스탬프를 사용하여 구성하는 `splice_time` 필드만 포함됩니다.

다음 표에서는 MediaTailor가 각 `segmentation_descriptor` 메시지에 필요한 필드를 설명합니다. 자세한 내용은 [ANSI 웹 스토어 웹 사이트에서](#) 구매할 수 있는 2022 SCTE-35 사양의 섹션 10.3.3.1을 참조하세요.

segmentation_descriptor 메시지의 필수 필드

Field	유형	기본값	설명
segmentation_event_id	정수	1	이름에 기록됩니다segmentation_descriptor.segmentation_event_id .
segmentation_upid_type	정수	14(0x0E)	이름에 기록됩니다segmentation_descriptor.segmentation_upid_type . 값은 0에서 256 사이여야 합니다.
segmentation_upid	문자열	"" (빈 문자열)	이름에 기록됩니다segmentation_descriptor.segmentation_upid . 값은 문자 0-9 및를 포함하는 16진수 문자열 이어야 합니다A-F.
segmentation_type_id	정수	48(0x30)	이름에 기록됩니다segmentation_descriptor.segmentation_type_id . 값은 0에서 256 사이여야 합니다.

Field	유형	기본값	설명
segment_num	정수	0	이논에 기록됩니다segmentat ion_descr iptor.segment_num . 값은 0에서 256 사이여야 합니다.
segments_expected	정수	0	이논에 기록됩니다segmentat ion_descr iptor.segments_expected . 값은 0에서 256 사이여야 합니다.
sub_segment_num	정수	null	이논에 기록됩니다segmentat ion_descr iptor.sub _segment_num . 값은 0에서 256 사이여야 합니다.
sub_segments_expected	정수	null	이논에 기록됩니다segmentat ion_descr iptor.sub _segments _expected . 값은 0에서 256 사이여야 합니다.

다음 표에는 MediaTailor가 일부 segmentation_descriptor 메시지 필드에 대해 자동으로 설정하는 값이 나와 있습니다.

segmentation_descriptor 메시지 필드에 대해 MediaTailor에서 설정한 값

Field	유형	값
segmentation_event_cancel_indicator	부울	True
program_segmentation_flag	부울	True
delivery_not_restricted_flag	부울	True

MediaTailor는 항상 segmentation_duration_flag로 설정합니다 True. MediaTailor는 segmentation_duration 필드를 상태 콘텐츠의 기간으로 틱 단위로 채웁니다.

 Note

MediaTailor는 time_signal 메시지를 전송할 때 splice_info_section 메시지의 splice_command_type 필드를 6(0x06)으로 설정합니다.

HLS 출력에서 time_signal 메시지가 AdBreak 있는의 경우 출력 EXT-X-DATERANGE 태그에는 splice_info_section 메시지의 직렬화된 버전으로 설정된 SCTE-35 필드가 포함됩니다. 예를 들어 다음 EXT-X-DATERANGE 태그는 splice_info_section 메시지의 직렬화된 버전을 보여줍니다.

```
#EXT-X-DATERANGE:ID="\1",START-DATE="\2020-09-25T02:13:20Z",DURATION=300.0,SCTE35-OUT=0xFC002C0000000000000000FF00506800000000001602144355454900000000100E000019BFCC00E00300000000
```

DASH 출력에서 time_signal 메시지가 AdBreak 있는의 경우 출력 EventStream 요소에 scte35:TimeSignal는 및 scte35:SpliceInfoSection 요소가 하위 요소인 scte35:SegmentationDescriptor 요소가 포함됩니다. scte35:TimeSignal 요소에는 하위 scte35:SpliceTime 요소가 있고 scte35:SegmentationDescriptor 요소에는 하위 scte35:SegmentationUpid 요소가 있습니다. 예를 들어 다음 DASH 출력은 EventStream 요소의 하위 요소 구조를 보여줍니다.

```
<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="270000000" segmentationEventCancelIndicator="false"
segmentationEventId="1" segmentationTypeId="48" segmentsExpected="0">
    <scte35:SegmentationUpid segmentationUpidFormat="hexBinary"
segmentationUpidType="14">012345</scte35:SegmentationUpid>
  </scte35:SegmentationDescriptor>
</scte35:SpliceInfoSection>
</Event>
</EventStream>

```

SCTE-35 메시지를 사용하여 채널 어셈블리의 광고 시간, 해당 메시지의 구조 및 필수 필드, SCTE-35 메시지가 포함된 샘플 HLS 및 DASH 출력을 설정하는 방법을 배웠습니다.

채널 재생 시간 전환

MediaTailor 채널 어셈블리를 사용하면 채널의 재생 시간을 변경할 수 있습니다. 시간 이동을 통해 최종 사용자는 원래 브로드캐스트 시간이 아닌 시간에 채널의 콘텐츠를 볼 수 있습니다. 예를 들어, 시청자는 프로그램을 시작한 후 브로드캐스트에 참여하더라도 처음부터 프로그램 시청을 시작할 수 있습니다. 또는 최종 사용자는 프로그램을 일시 중지하고 나중에 다시 볼 수 있습니다.

채널의 재생을 시간 이동하려면 채널의 재생 URL에 `start` 파라미터를 추가합니다. `start` 파라미터는 현재 시간을 기준으로 재생이 시작되어야 하는 시간을 지정합니다. 예를 들어 현재 시간이 오후 2시이고 오후 1시에 재생을 시작하려면 `start` 파라미터를 초당 -1시간-3600인 로 설정합니다.

`start` 파라미터는 양수 값과 음수 값을 모두 초 단위로 허용합니다.

- 음수 값은 현재 시간을 기준으로 과거의 시간을 나타냅니다. 예를 들어는 "1시간 전 재생 시작"을 `start=-3600` 의미합니다.
- 양수 값은 현재 시간을 기준으로 미래 시간을 나타냅니다. 예를 들어는 "향후 1시간부터 재생 시작"을 `start=3600` 의미합니다.

다음 예제에서는 채널의 재생 URL에 `start` 파라미터를 추가하는 방법을 보여줍니다.

```

https://a1b2c3d4e5f6.mediapackage.us-west-2.amazonaws.com/out/v1/examplechannel/
index.m3u8?start=-3600

```

이 예제에서는 재생이 1시간 전부터 시작됩니다.

`start` 파라미터에는 다음과 같은 제한 사항이 적용됩니다.

- 최대 시간 이동 기간은 과거 또는 미래 24시간입니다.
- 시간 이동은 선형 재생 모드를 사용하는 채널에서만 사용할 수 있습니다.
- 루프 재생 모드를 사용하는 채널에는 시간 전환을 사용할 수 없습니다.

사용 가능한 시간 전환 기간을 벗어나는 `start` 파라미터를 지정하면 MediaTailor가 오류를 반환합니다.

CDN을 사용하여 MediaTailor 광고 개인화 및 콘텐츠 전송 최적화

AWS Elemental MediaTailor 는 독립 실행형 서비스로 효과적으로 작동하지만 Amazon CloudFront 또는 기타 타사 CDN과 같은 콘텐츠 전송 네트워크(CDNs)와 통합하면 스트리밍 워크플로를 크게 개선할 수 있습니다. CDN 통합은 지리적으로 분산된 대규모 대상에게 콘텐츠를 제공해야 하거나 다양한 AWS 리전에서 일관된 광고 전송을 보장하려는 경우에 특히 유용합니다.

CDN이 없으면 뷰어는 맞춤형 매니페스트 및 광고 세그먼트를 위해 MediaTailor에 직접 연결되므로 특히 MediaTailor 구성이 배포된 AWS 리전에서 멀리 떨어진 뷰어의 경우 지연 시간이 늘어날 수 있습니다. 또한 트래픽이 많은 이벤트 중에 MediaTailor에 직접 연결하면 로드가 증가하여 성능에 영향을 미칠 수 있습니다.

MediaTailor 개념 및 워크플로에 대한 자세한 내용은 섹션을 참조하세요 [AWS Elemental MediaTailor란 무엇인가요?](#).

CDN을 MediaTailor와 통합할 때는 웹 기반 플레이어에서 재생 실패를 일으킬 수 있는 문제를 방지하기 위해 적절한 CORS(교차 오리진 리소스 공유) 처리를 구성하는 것이 중요합니다. 적절한 CORS 구성은 광고 세그먼트와 콘텐츠 세그먼트 모두에 필수적입니다. 광고 세그먼트는 CORS 문제에 더 취약하지만 모든 세그먼트 유형에 일관된 CORS 처리를 적용하면 가장 안정적인 재생 환경을 보장할 수 있습니다. 적절한 CORS 처리로 CDN 라우팅 동작을 구성하는 방법에 대한 자세한 지침은 섹션을 참조하세요 [MediaTailor에 대한 프로덕션 지원 CloudFront 구성](#).

CDN 통합을 통해 고급 파라미터 전달 및 동적 라우팅 기능도 사용할 수 있습니다. 권한 부여 및 라우팅을 위해 CDN의 [MediaTailor 매니페스트 쿼리 파라미터](#). 구성 별칭을 사용한 동적 광고 서버 및 오리진 라우팅은 섹션을 참조하세요 [ADS 요청에 대한 MediaTailor 동적 광고 변수](#).

최종 사용자와 MediaTailor 사이에 CDN을 배치하면 다음과 같은 이점이 있습니다.

- 엣지 로케이션의 콘텐츠를 최종 사용자에게 더 가깝게 제공하여 지연 시간 단축
- CDN의 글로벌 인프라에 로드를 분산하여 확장성 향상
- 중복 전송 경로를 통해 안정성 향상
- 오리진 트래픽을 줄여 비용 최적화
- 스트리밍 품질 향상을 위해 Media Quality-Aware Routing(MQAR)과 같은 고급 기능 구현

주제

- [필요에 적합한 CDN 선택](#)
- [에 대한 CDN 통합 계획 AWS Elemental MediaTailor](#)
- [MediaTailor와 CDN 통합 설정](#)
- [맞춤형 비디오 광고를 위해 CDN으로 SSAI 설정](#)
- [채널 어셈블리 및 CDN을 사용하여 MediaTailor 선형 채널 구축](#)
- [MediaTailor를 MediaPackage 및 CDN과 통합](#)
- [Amazon CloudFront AWS Elemental MediaTailor 와 통합](#)
- [MediaTailor 광고 전송을 위한 타사 CDN 설정](#)
- [CDN 및 MediaTailor 통합을 위한 성능 최적화 가이드](#)
- [MediaTailor CDN 작업 및 성능 모니터링](#)
- [CDN 및 MediaTailor 통합 테스트 및 검증](#)
- [MediaTailor 및 CDN 통합 관련 문제 해결](#)
- [MediaTailor에 대한 CDN 통합 로그 분석 및 오류 코드 참조](#)
- [를 사용하여 MediaTailor 및 CDN 자동화 CloudFormation](#)
- [MediaTailor에 대한 프로덕션 지원 CloudFront 구성](#)
- [CDN 및 MediaTailor 통합에 대한 지원 및 문제 해결 도움말 받기](#)

필요에 적합한 CDN 선택

올바른 콘텐츠 전송 네트워크(CDN) 공급자를 선택하는 것은 콘텐츠 전송 성능, 비용 및 최종 사용자 경험에 영향을 미칠 수 있는 중요한 결정입니다 AWS Elemental MediaTailor. MediaTailor 구현을 위한 CDN을 선택할 때 다음 요소를 고려하세요.

지리적 적용 범위

대상이 위치한 리전에 강력한 입지를 갖춘 CDN을 선택합니다. CDN 공급자마다 지리적 리전마다 강점이 다릅니다.

AWS 서비스와의 통합

Amazon CloudFront는 MediaTailor 및 기타 AWS 서비스와 가장 긴밀하게 통합되어 설정 및 관리를 간소화할 수 있습니다. 타사 CDN은 특수 비디오 전송 기능 또는 특정 리전에서의 강력한 존재와 같은 다른 이점을 제공할 수 있습니다.

비디오별 기능

적응형 비트 전송률 최적화, 비디오 압축, 최종 사용자 경험에 초점을 맞춘 분석과 같이 비디오 전송을 위해 특별히 설계된 기능을 제공하는 CDN을 찾습니다.

비용 구조

트래픽 볼륨, 지리적 분포 및 기능 요구 사항과 같은 요소를 고려하여 공급자 간에 요금 모델을 비교합니다. 일부 CDN 사용량 패턴과 일치할 수 있는 대량 할인 또는 약정된 사용 할인을 제공합니다.

고급 기능 지원

선택한 CDN이 토큰 인증, 지리적 제한, 요청 축소, 적절한 헤더 전달과 같은 필요한 기능을 지원하는지 확인합니다.

특정 CDN 공급자 및 MediaTailor와의 통합에 대한 자세한 내용은 다음 리소스를 참조하세요.

- [CloudFront 통합](#) Amazon CloudFront용
- [타사 CDN 설정](#) 타사 CDN 공급자용

다음 주제에서는 최적의 성능을 위해 MediaTailor를 CDN으로 구성하는 방법에 대한 포괄적인 지침을 제공합니다.

에 대한 CDN 통합 계획 AWS Elemental MediaTailor

용 CDN 통합을 통해 최종 사용자 환경을 개선하고 지연 시간을 줄일 수 있습니다 AWS Elemental MediaTailor. 콘텐츠 전송 네트워크(CDN)를 구현하면 최종 사용자에게 더 가까운 위치에서 콘텐츠를 제공할 수 있습니다. 이렇게 하면 여러 지리적 리전에서 로드 시간 단축, 확장성 향상, 일관된 광고 전송이 보장됩니다.

를 사용하여 CDN을 구현하기 전에 적절한 계획이 필요합니다 AWS Elemental MediaTailor. 이 섹션에서는 주요 계획 영역을 안내합니다. 실제 구성을 시작하기 전에 이러한 영역을 해결합니다. 이 단계는 대상을 위한 최적의 시청 환경을 만드는 데 도움이 됩니다.

계획을 완료한 후 step-by-step 구현 지침은 [MediaTailor와 CDN 통합](#)을 참조하세요.

CDN 계획에 영향을 미칠 수 있는 MediaTailor 할당량에 대한 자세한 내용은 섹션을 참조하세요 [의 할당량 AWS Elemental MediaTailor](#). CloudFront 할당량에 대한 자세한 내용은 CloudFront 개발자 안내서의 [의 할당량을](#) 참조하세요.

계획을 시작하기 전에 MediaTailor가 CDN과 상호 작용하는 방법을 이해합니다.

1. 최종 사용자는 MediaTailor에서 직접 요청하는 대신 CDN을 통해 콘텐츠를 요청합니다.
2. CDN은 개인화를 위해 매니페스트 요청을 MediaTailor에 전달합니다.
3. CDN은 엣지 로케이션에서 콘텐츠 세그먼트와 광고 세그먼트를 캐싱하고 제공합니다.

이 아키텍처는 MediaTailor에 대한 로드를 줄이는 동시에 최종 사용자가 지연 시간을 최소화하면서 개인화된 광고를 수신할 수 있도록 합니다.

매니페스트 용어를 이해하면 CDN을 올바르게 구성하는 데 도움이 됩니다. 다양한 스트리밍 프로토콜은 캐싱 및 라우팅 설정 방식에 영향을 미치는 특정 매니페스트 구조를 사용합니다.

- HLS 매니페스트 - HLS 스트림으로 작업할 때 다음을 처리합니다.
 - 다변량 재생 목록: 개인화를 위해 이러한 최상위 매니페스트를 MediaTailor로 라우팅하도록 CDN을 구성합니다.
 - 미디어 재생 목록: 콘텐츠 세그먼트에 대한 링크가 포함된 이러한 매니페스트에 대해 적절한 캐싱 규칙을 설정합니다.
- DASH 매니페스트 - DASH 스트림으로 작업할 때 다음을 처리합니다.
 - MPD(미디어 프레젠테이션 설명): 개인화 요구 사항에 따라 이러한 매니페스트를 처리하도록 CDN을 구성합니다.

CDN 계획 프로세스에는 다음과 같은 주요 단계가 포함되며, 각 단계는 특정 작업에 중점을 둡니다.

- [CDN 및 MediaTailor 통합에 대한 트래픽 요구 사항 추정](#): 예상 최종 사용자 동시성과 대역폭 요구 사항을 계산합니다.
- [CDN 및 MediaTailor 통합을 위한 최적화 전략 구성](#): 최적의 콘텐츠 전송 및 광고 개인화를 위해 CDN을 구성합니다.
- [CDN 및 MediaTailor 통합 계획 사용자 지정](#): 특정 MediaTailor 워크플로를 기반으로 CDN 전략을 조정합니다.
- [CDN 및 MediaTailor 통합에 대한 모니터링 및 조정 설정](#): 안정적인 성능을 위한 모니터링 및 조정 전략을 구현합니다.
- [CDN 및 MediaTailor 통합 비용 최적화](#): 성능과 비용 효율성의 균형을 맞춥니다.
- [CDN 및 MediaTailor 통합에 대한 구현 테스트](#): 프로덕션 배포 전에 CDN 통합을 철저히 테스트합니다.

CDN 및 MediaTailor 통합에 대한 트래픽 요구 사항 추정

와 콘텐츠 전송 네트워크(CDN) 통합의 크기를 정확하게 조정하려면 AWS Elemental MediaTailor:

1. 기록 데이터 또는 유사한 이벤트를 사용하여 예상 최종 사용자 동시성을 계산합니다. 예상치 못한 스파이크를 처리하기 위해 기존 이상의 추가 용량을 계획합니다. 현재 조정 권장 사항은 AWS 계정 팀에 문의하세요. 그 밖에 [의 할당량 AWS Elemental MediaTailor](#)을 확인할 수도 있습니다.
2. 콘텐츠 일정에서 피크 트래픽 패턴과 잠재적 스파이크를 식별합니다. 다음과 같은 요소를 고려하세요.
 - 라이브 스포츠 이벤트 또는 시즌 특별 이벤트
 - 마케팅 캠페인 또는 프로모션 이벤트
 - 전 세계 시청자의 시간대 차이
 - 공휴일 또는 계절 보기 패턴
3. 최종 사용자 수에 스트림 비트레이트를 곱하여 대역폭 요구 사항을 결정합니다. CDN 공급자와 협력하여 적절한 용량을 계산합니다. 이 계산은 특정 콘텐츠 비트레이트와 예상 대상 크기를 기준으로 합니다. 공급자가 권장하는 대로 광고 세그먼트 및 매니페스트 요청에 대한 오버헤드를 추가합니다.
4. CDN 공급자와 협력하여 대상 리전에 충분한 엣지 용량을 확보합니다.

다음과 같은 특정 작업을 수행하여 광고 삽입 용량이 최종 사용자 수요를 충족하는지 확인합니다.

1. [Service Quotas 콘솔](#)에서 현재 광고 삽입 요청 할당량을 확인합니다. 현재 서비스 제한을 검토하여 구성에서 지원할 수 있는 동시 뷰어 수를 파악합니다.
2. 트래픽이 많은 이벤트의 경우 [Service Quotas 콘솔](#)을 통해 증가된 할당량을 요청합니다.
3. 동시 시청자가 500,000명 이상이 될 것으로 예상되는 경우 이벤트 최소 2주 전에 [AWS Support](#)에 문의하세요. 이를 통해 AWS는 광고 개인화 요구 사항에 충분한 용량을 확보할 수 있습니다.

워크플로에서 용량 계획을 구현하는 방법에 대한 자세한 내용은 [미리 가져오기 일정 관리를 사용하여](#) 트래픽이 많은 이벤트에 대한 광고 전송 최적화를 참조하세요.

CDN 및 MediaTailor 통합을 위한 최적화 전략 구성

트래픽 추정을 완료하면 콘텐츠 전송 네트워크(CDN)를 구성하여 콘텐츠 전송 및 광고 개인화를 최적화합니다 AWS Elemental MediaTailor. 이러한 최적화는 대상 광고를 유지하면서 원활한 재생을 보장하는 데 도움이 됩니다.

다음과 같은 특정 CDN 최적화를 구현합니다.

1. CDN에서 오리진 실드 기능을 구성하여 MediaTailor의 부하를 줄이고 캐싱 효율성을 개선합니다. 오리진 실드는 다음과 같은 중간 캐싱 계층 역할을 합니다.

- 여러 최종 사용자 요청을 단일 오리진 요청으로 통합
- MediaTailor에 대한 중복 요청 수 감소
- 캐시된 콘텐츠의 응답 시간 개선

CloudFront를 사용하여 오리진 실드를 설정하는 방법에 대한 구현 세부 정보는 CloudFront 개발자 안내서의 [Origin Shield 사용](#)을 참조하세요.

2. 다양한 콘텐츠 유형에 적절한 TTL(Time To Live) 값을 설정합니다. TTL은 CDN이 콘텐츠를 캐싱하는 기간을 결정합니다. 이 시간이 지나면 CDN은 오리진에서 새 사본을 요청합니다.

- 매니페스트:
 - 광고 삽입의 경우 0초
 - 채널 어셈블리의 경우 5~10초

광고 삽입에서 MediaTailor는 최종 사용자에게 개인화된 광고를 매니페스트에 제공합니다. 재생 목록 또는 MPD가 캐시되어 잘못된 재생 디바이스에 제공되는 경우 디바이스에 재생 또는 추적 문제가 발생할 수 있습니다.

- 콘텐츠 세그먼트: 24시간 이상(이는 거의 변경되지 않으며 오리진 로드를 줄이기 위해 적극적으로 캐싱할 수 있음)
- 광고 세그먼트: 24시간 이상(광고 콘텐츠는 일반적으로 최종 사용자 간에 재사용되며 장기간 캐시할 수 있음)

모든 MediaTailor 워크플로에 대한 포괄적인 TTL 권장 사항 및 캐싱 최적화 전략은 섹션을 참조하세요 [CDN 및 MediaTailor 통합을 위한 캐싱 최적화](#).

CloudFront에서 캐시 동작을 구성하는 방법에 대한 자세한 지침은 CloudFront 개발자 안내서의 [캐시 동작 설정](#)을 참조하세요.

3. 최종 사용자 모집단 근처에 CDN 엣지 노드를 배포합니다. CDN 공급자와 협력하여 다음을 수행합니다.

- 최종 사용자 인구 통계를 기반으로 최적의 엣지 노드 위치 식별
- 각 리전에서 충분한 용량 보장
- 엣지 성능을 모니터링하고 필요에 따라 조정

구현 지침은 [CloudFront 엣지 로케이션](#)을 참조하여 대상 리전에서 사용 가능한 엣지 로케이션을 식별하세요.

4. 글로벌 대상의 경우 다중 CDN 전략을 구현하는 것이 좋습니다. 이 접근 방식은 다음과 같습니다.

- 여러 CDN 공급자를 사용하여 신뢰성 향상
- 최종 사용자를 해당 위치에 가장 적합한 CDN으로 라우팅합니다.
- CDN 중단 시 장애 조치 옵션 제공
- 다양한 요금 모델을 활용하여 비용을 최적화할 수 있습니다.

구현 세부 정보는 AWS 네트워킹 및 콘텐츠 전송 블로그의 [다중 CDN 전략](#)을 참조하세요.

CDN 및 MediaTailor 통합 계획 사용자 지정

AWS Elemental MediaTailor 워크플로마다 콘텐츠 전송 네트워크(CDN) 계획에 영향을 미치는 고유한 요구 사항이 있습니다. 워크플로별 지침은 [구성 작업을](#) 참조하세요.

특정 MediaTailor 워크플로에 따라 용량 계획을 조정합니다.

MediaTailor 광고 삽입 워크플로의 경우

1. 캐싱 없이 개인화된 매니페스트를 처리하도록 CDN을 구성합니다. 이렇게 하면 각 시청자가 고유한 대상 광고를 수신할 수 있습니다. 광고는 프로필과 보기 컨텍스트를 기반으로 합니다.
2. 광고 결정 서버(ADS)의 크기를 조정하여 최대 요청 볼륨을 처리합니다. ADS 구성에 대한 지침은 [MediaTailor를 사용한 광고 삽입](#)을 참조하세요. 다음을 고려하세요.
 - 사용 사례에 대한 응답 시간 요구 사항
 - 예상 동시 뷰어 용량
 - 중복 및 장애 조치 요구 사항
 - 지리적 배포 요구 사항
3. CDN 수준에서 요청 축소를 구현하여 동기화된 광고 중단 요청을 처리합니다. 요청 축소는 여러 개의 동일한 요청을 단일 오리진 요청으로 결합합니다. 이는 다음과 같은 경우에 매우 중요합니다.
 - 많은 시청자가 동시에 광고 시간에 도달하는 라이브 스포츠 이벤트
 - 동기화된 상업적 휴식이 포함된 인기 TV 프로그램 프리미어
 - 갑작스러운 시청자 급증을 유발하는 뉴스 이벤트

구현 세부 정보는 [오리진 장애 조치](#)를 참조하여 피크 로드 중 요청 처리를 구성합니다.

MediaTailor 채널 어셈블리 워크플로의 경우

1. 채널 수와 비트레이트를 기준으로 용량 요구 사항을 계산합니다. 채널 어셈블리 용량 계획에 대한 지침은 [MediaTailor의 채널 어셈블리를](#) 참조하세요. 다음을 고려하세요.
 - 총 채널 수
 - 채널당 비트 전송률 요구 사항
 - 예상 동시 최종 사용자 로드
 - 지리적 배포 요구 사항
2. 게시된 일정에 따라 예측 가능한 트래픽 패턴을 처리하도록 CDN을 구성합니다. 채널 어셈블리는 일반적으로 광고 삽입보다 예측 가능한 패턴을 갖습니다. 그 이유는 다음과 같습니다.
 - 프로그래밍 일정을 미리 알고 있음
 - 뷰어 동작이 설정된 패턴을 따릅니다.
 - 최종 사용자별로 콘텐츠가 동적으로 변경되지 않음
3. 오리진에 일관된 채널 출력을 유지하기에 충분한 대역폭이 있는지 확인합니다. 구현:
 - 고가용성 채널을 위한 중복 오리진 서버
 - 기본 오리진과 백업 오리진 간의 자동 장애 조치
 - 오리진 성능 문제를 감지하기 위한 모니터링

구현 지침은 복원력이 뛰어난 [오리진 인프라를 생성하기 위한 오리진 중복 설정을](#) 참조하세요.

결합된 MediaTailor 워크플로의 경우

1. 인프라 크기를 조정하여 두 서비스의 결합된 트래픽 패턴을 처리합니다. 결합된 워크플로에 대한 지침은 섹션을 참조하세요 [AWS Elemental MediaTailor 를 사용하여 광고 삽입](#). 다음을 고려하세요.
 - 채널 어셈블리 기준 요구 사항
 - 광고 삽입 오버헤드 요구 사항
 - 피크 트래픽 패턴
 - 중복 요구 사항
2. 선형 콘텐츠 전송 및 동적 광고 삽입을 위해 별도의 CDN 동작을 구성합니다. 이 분리를 통해 다음을 수행할 수 있습니다.
 - 각 콘텐츠 유형에 대한 캐싱 정책을 독립적으로 최적화
 - 콘텐츠 유형에 따라 요청을 적절한 오리진으로 라우팅

3. 최적의 성능을 유지하기 위해 엣지 CDN과 오CDNs 간에 적절한 라우팅을 설정합니다. 다음을 사용하는 것이 좋습니다.

- 콘텐츠 세그먼트(/content/*)와 광고 세그먼트(/ads/*)의 서로 다른 오리진 경로
- 매니페스트와 세그먼트에 대한 별도의 캐시 동작
- 여러 리전의 지연 시간을 최적화하기 위한 지리적 라우팅

구현 세부 정보는 경로 기반 라우팅 및 캐싱 규칙을 설정하도록 [캐시 동작 구성](#)을 참조하세요.

CDN 및 MediaTailor 통합에 대한 모니터링 및 조정 설정

AWS Elemental MediaTailor 콘텐츠 전송 네트워크(CDN) 통합을 통해 최적의 성능과 최종 사용자 경험을 유지하려면 효과적인 모니터링 및 규모 조정 전략이 중요합니다. 이러한 접근 방식을 구현하여 CDN 통합이 대규모로 안정적으로 수행되도록 합니다.

다음과 같은 모니터링 및 조정 전략을 구현합니다.

1. 이러한 주요 지표에 대한 모니터링을 구성합니다. 적절한 대상 값에 대한 지침은 [Amazon CloudWatch를 사용하여 MediaTailor 모니터링](#)을 참조하고 CDN 공급자의 모범 사례를 참조하세요.

- CDN 캐시 적중률(콘텐츠 유형 및 전송 패턴을 기반으로 기존 지표 및 대상 설정)
- 오리진 요청 볼륨(정상 작업 중 패턴을 모니터링하여 이상 탐지 기준을 설정)
- 오류 유형별 오류 발생률(서비스 수준 목표 및 MediaTailor 모범 사례를 기반으로 임계값 정의)
- 응답 시간(최종 사용자 경험 요구 사항 및 지리적 분포에 따라 적절한 지연 시간 대상 설정)

자세한 구현 지침은 MediaTailor 및 CDN 지표를 함께 시각화하는 [CloudWatch 대시보드 생성](#)을 참조하세요.

2. 예상치 못한 트래픽 패턴 또는 성능 저하에 대한 알림을 설정합니다. 기존 지표 및 서비스 수준 목표에 따라 임계값을 구성합니다. 알림 설정에 대한 지침은 [Amazon CloudWatch 경보 생성](#)을 참조하세요. 모니터링을 고려하세요.

- 기존 캐시 적중률과의 상당한 편차(일반적으로 85~90% 미만일 때 알림)
- 오리진 요청 볼륨의 갑작스러운 증가(기준 대비 30% 이상 증가 시 경고)
- 정의된 임계값을 초과하는 오류 발생률 급증(일반적으로 4xx 오류의 경우 1~2%, 5xx 오류의 경우 0.5%)
- 허용 수준을 초과하는 응답 시간 저하(일반적으로 매니페스트의 경우 >500ms, 세그먼트의 경우 >200ms)

구현 예제는 효과적인 모니터링 대시보드 및 알림 생성을 위한 [CloudWatch 개념](#)을 참조하세요.

3. 예측 가능한 트래픽이 많은 이벤트에 대한 조정 계획을 생성합니다. 계획에는 다음과 같은 주요 요소가 포함되어야 합니다.

- 이벤트 전 용량 증가(이벤트 시작 24~48시간 전)
- 점진적 최종 사용자 증가 일정(일반적으로 5분 간격당 예상 대상의 10~20%)
- 대상에 따른 리전 용량 분포(예상 리전 시청자에 비례하여 용량 할당)
- 이벤트 후 조정 절차(이벤트 종료 후 30~60분 동안 최대 용량 유지)

트래픽이 많은 이벤트의 규모 조정에 대한 구현 지침은 AWS 미디어 블로그의 [복원력이 뛰어난 end-to-end 라이브 워크플로 설정](#)을 참조하세요.

4. 다음을 포함하여 중요한 스트림에 대한 장애 조치 및 중복 조치를 구현합니다.

- 다중 리전 CDN 배포(중요 콘텐츠의 경우 최소 2개의 리전)
- 백업 오리진 서버(30초마다 자동 상태 확인으로 구성됨)
- 상태 확인을 기반으로 하는 자동 장애 조치 트리거(일반적으로 2~3회의 실패한 검사 후)
- 다양한 장애 시나리오에 대한 복구 절차(특정 응답 시간 대상으로 문서화됨)

자세한 구현 단계는 [CloudFront 오리진 장애 조치를 통한 고가용성 최적화](#)를 참조하세요.

CDN 및 MediaTailor 통합 비용 최적화

콘텐츠 전송 네트워크(CDN) 비용은 트래픽 패턴, 지리적 배포 및의 기능 사용량에 따라 크게 달라질 수 있습니다 AWS Elemental MediaTailor. 현재 요금 정보는 [CloudFront 요금](#)을 참조하세요. CDN 공급자의 설명서를 참조할 수도 있습니다.

다음 전략을 사용하여 성능과 비용 효율성의 균형을 맞춥니다.

1. CDN 트래픽 패턴을 분석하여 가장 비용 효율적인 요금 티어를 선택합니다. CDN 공급자와 함께 다음 요소를 검토합니다.

- 리전 및 기간별 데이터 전송 볼륨
- 매니페스트 및 세그먼트에 대한 요청 패턴
- 지리적 배포 요구 사항
- 피크 및 평균 사용 패턴

비용 분석에 도움이 필요하면 [AWS 요금 계산기](#)를 사용하여 특정 사용 패턴을 기반으로 CloudFront 비용을 추정합니다.

2. 예측 가능한 워크로드의 경우 CDN 공급자와의 예약 용량 계약을 평가합니다. 이러한 계약은 다음과 같은 이점을 제공할 수 있습니다.

- 약정 사용량에 대한 할인 요금
- 예산 책정을 위한 예측 가능한 월별 비용
- 우선 순위 지원 및 용량 할당

CDN 공급자에게 문의하여 예약 용량이 사용 패턴에 적합한지 확인하세요. CloudFront의 경우 [CloudFront 프리미엄 기능을](#) 참조하세요. 예약 용량 옵션에 대한 정보를 제공합니다.

3. MediaTailor와 CDN 공급자 간의 트래픽을 밸런싱하여 송신 비용을 최적화합니다. 전략에는 다음이 포함됩니다.

- 캐시 적중률을 극대화하여 오리진 요청 감소
- 오리진 실드를 사용하여 요청 통합
- 데이터 전송 볼륨을 줄이기 위한 압축 구현
- 리전 요금에 맞는 CDN AWS 리전 선택

압축에 대한 구현 지침은 CloudFront 개발자 안내서의 [압축 파일 제공을 참조하세요](#).

4. 다양한 콘텐츠 유형에 적합한 캐싱 전략을 구현하여 오리진 요청을 줄입니다. 캐시 최적화에 대한 지침은 [캐시 적중률 개선을](#) 참조하세요. 중점 사항:

- 콘텐츠 세그먼트(장기 캐시 가능)
- 광고 세그먼트(일반적으로 최종 사용자 간에 재사용됨)
- 플레이어 파일 및 이미지와 같은 정적 자산

캐시 적중률이 개선되면 오리진 비용이 크게 절감됩니다. CDN 공급자와 협력하여 특정 콘텐츠 패턴에 맞게 캐시 구성을 최적화합니다. 자세한 구현 단계는 [캐시 동작 구성을](#) 참조하세요.

CDN 및 MediaTailor 통합에 대한 구현 테스트

프로덕션 배포 전에 AWS Elemental MediaTailor 콘텐츠 전송 네트워크(CDN) 통합을 철저히 테스트하여 신뢰할 수 있는 최종 사용자 경험을 보장합니다. 적절한 테스트는 대상에 영향을 미치기 전에 문제를 식별하고 해결하는 데 도움이 됩니다. 테스트 방법에 대한 지침은 [CloudFront 배포 테스트를 참조하세요](#). CDN 공급자의 테스트 설명서를 참조할 수도 있습니다.

CDN 통합을 검증하는 다음 테스트 단계를 따르세요.

1. 프로덕션 구성을 미러링하는 테스트 환경을 생성합니다. 포함:

구현 테스트

- 동일한 CDN 설정 및 캐시 동작
- 다양한 비트 전송률 및 형식의 대표 콘텐츠 포함
- 샘플 광고 응답으로 테스트 광고 결정 서버 구성
- 모니터링 및 알림 구성 설정

step-by-step 구현 지침은 CloudFront 개발자 안내서의 [스테이징 배포 생성](#)을 참조하세요.

2. 로드 테스트를 수행하여 용량 추정치를 확인합니다. 로드 테스트에 대한 지침은 [MediaPackage 모니터링](#)을 참조하세요. 테스트 시나리오에는 다음이 포함되어야 합니다.

- 점진적 최종 사용자 수 증가(일반적으로 5분마다 예상 피크의 10~20%)
- 예상 피크 부하에 따라 갑작스러운 트래픽 급증(60초 이내에 피크 대상의 50%를 조인 시뮬레이션)
- 최대 부하 기간 연장(최소 30~60분 동안 최대 부하 유지)
- 대상과 일치하는 지리적 배포(예상 시청자 위치에 따라 테스트 트래픽 배포)

응답 시간이 목표 임계값 미만으로 유지되는지 확인합니다. 일반적으로 매니페스트의 경우 500ms 미만, 세그먼트의 경우 200ms 미만입니다. 오류 발생률은 1% 미만으로 유지되어야 합니다. 로드 테스트 도구 및 방법론에 대한 구현 세부 정보는 AWS 네트워킹 및 콘텐츠 전송 블로그의 [CloudFront를 사용한 로드 테스트](#)를 참조하세요.

3. 장애 조치 시나리오를 테스트하여 신뢰성을 보장합니다. 시뮬레이션:

- 오리진 서버 장애(완전 중단 및 부분 성능 저하 시나리오)
- CDN 엣지 로케이션 중단(백업 위치로 트래픽 라우팅을 사용하여 테스트)
- 광고 결정 서버를 사용할 수 없음(5~10초 제한 시간으로 테스트)
- 네트워크 연결 문제(패킷 손실 및 지연 시간 증가 시뮬레이션)

CDN 공급자와 협력하여 사용 사례에 적합한 장애 조치 응답 시간 목표를 설정합니다. 일반적으로 이는 장애 조치 완료까지 3초 미만을 의미합니다. 장애 조치 테스트에 대한 구현 지침은 CloudFront 개발자 안내서의 [오리진 장애 조치를](#) 참조하세요.

4. 주요 이벤트의 경우 [AWS 로드 테스트 지침](#)에 따라 점진적인 램프 업 전략을 구현합니다.

- 서로 다른 대상 세그먼트의 시작 시간에 시차를 둡니다. 예를 들어 대상 그룹 간에 15분 간격으로 프리미엄 구독자를 먼저 허용합니다.
- 사전 워밍 기법을 사용하여 부하를 점진적으로 늘립니다. 사전 워밍에는 다음이 포함됩니다.
 - 이벤트 2~4시간 전에 인기 콘텐츠로 CDN 캐시 채우기
 - 합성 트래픽을 예상 피크의 20~30%까지 점진적으로 늘려 시스템을 워밍업합니다.

- 실제 콘텐츠를 사용하여 실제 로드 조건에서 모든 구성 요소 테스트
- 램프 업 기간 동안 시스템 성능을 모니터링하여 다음을 추적합니다.
 - 캐시 적중률 및 응답 시간(목표 >90% 적중률, <500ms 응답)
 - 오류율 및 오리진 로드(오류율 <1%, 오리진 CPU <70% 유지)
 - 광고 개인 맞춤 성공률(>98%의 성공적인 개인 맞춤 목표)
 - 뷰어 경험 지표(대상 <2초 시작 시간, <0.5% 재버퍼링)
- 예기치 않은 트래픽 급증에 대한 비상 계획을 수립합니다. 계획에는 다음과 같은 필수 구성 요소가 포함되어야 합니다.
 - 15분 내에 용량을 50~100% 늘리는 문서화된 단계가 포함된 비상 용량 조정 절차
 - 트래픽의 20~50%를 보조 CDN으로 전환할 수 있는 백업 CDN 활성화 프로토콜
 - 대상 지정 파라미터를 10개 이상에서 3~5개로 줄이기 위해 광고 삽입 폴백 간소화
 - 사전 정의된 알림 템플릿 및 연락처 목록이 있는 이해관계자를 위한 커뮤니케이션 계획

테스트를 완료한 후 프로덕션 배포 단계를 위한 [CDN 통합 구현으로](#) 이동합니다.

CDN 및 MediaTailor 통합의 일반적인 문제 해결

최종 사용자에게 영향을 미치지 AWS Elemental MediaTailor 전의 일반적인 콘텐츠 전송 네트워크(CDN) 통합 문제를 해결합니다. 이 섹션에서는 CDN과 통합하는 동안 발생하는 일반적인 문제를 식별하고 해결하는 데 도움이 됩니다 AWS Elemental MediaTailor.

포괄적인 문제 해결 지침은 [MediaTailor 문제 해결](#) 및 [CloudFront 배포 문제 해결을 참조하세요](#).

MediaTailor 매니페스트 전송 문제 해결

최종 사용자에게 재생 문제가 발생하거나 잘못된 광고가 표시되는 경우 다음과 같은 일반적인 매니페스트 관련 문제를 확인하세요.

- 잘못된 캐싱 설정: CDN이 개인화된 매니페스트를 캐싱하는 경우 최종 사용자는 다른 사용자를 위한 광고를 볼 수 있습니다.

해결 방법: MediaTailor에 대한 매니페스트 요청에 대해 캐시 TTL이 0인 CDN을 구성합니다.

- 오리진 요청 실패: CDN이 MediaTailor에 연결할 수 없는 경우 매니페스트 요청이 실패합니다.

해결 방법: CDN과 MediaTailor 간의 네트워크 연결을 확인합니다. CDN이 올바른 헤더를 올바르게 전달하는지 확인합니다.

- 세션 파라미터 문제: 세션 파라미터가 누락되거나 잘못되면 개인화에 실패할 수 있습니다.

해결 방법: 플레이어는 매니페스트 요청에 필요한 모든 세션 파라미터를 올바르게 추가하고 있는지 확인합니다.

MediaTailor 세그먼트 전송 문제 해결

콘텐츠 또는 광고 세그먼트가 제대로 로드되지 않는 경우 다음과 같은 일반적인 문제를 조사합니다.

- 세그먼트 경로 재작성: 잘못된 CDN 구성은 세그먼트 URLs 잘못 재작성할 수 있습니다.

해결 방법: CDN이 세그먼트 URLs를 올바르게 처리하는지 확인합니다. 참조를 끊는 방식으로 경로를 수정하지 않는지 확인합니다.

- CORS 구성: CORS 헤더가 누락되거나 올바르지 않으면 브라우저가 세그먼트를 로드하지 못할 수 있습니다.

해결 방법: 세그먼트 요청에 적합한 CORS 헤더를 전달하도록 CDN을 구성합니다.

- 캐시 누락 폭풍: 트래픽이 많은 이벤트 중에 여러 캐시 누락이 오리진 서버를 압도할 수 있습니다.

솔루션: 요청 축소 및 오리진 실드 기능을 구현하여 트래픽 급증 시 오리진 부하를 줄입니다.

MediaTailor CDN 성능 문제 해결

뷰어에게 버퍼링 또는 느린 로드가 발생하는 경우 다음과 같은 성능 관련 문제를 확인하세요.

- 낮은 캐시 적중률: CDN이 오리진에서 콘텐츠를 자주 요청하면 성능이 저하됩니다.

해결 방법: 콘텐츠 유형별로 캐시 적중률을 분석하고 TTL 설정을 조정하여 캐싱 효율성을 개선합니다.

- 지리적 분포: CDN 엣지 로케이션에서 멀리 떨어진 최종 사용자는 지연 시간 증가를 경험할 수 있습니다.

해결 방법: CDN 엣지 로케이션 배포를 검토합니다. 시청자 집중도가 높은 리전에 용량을 추가합니다.

- 오리진 용량 제한: 오리진 서버에 과부하가 걸리면 응답 시간이 증가합니다.

해결 방법: 오리진 요청 제한을 구현합니다. 오리진 용량을 늘리거나 캐싱을 개선하여 오리진 로드를 줄일 수도 있습니다.

추가 문제 해결 지원은 [MediaTailor 문제 해결을 참조](#)하거나 AWS Support에 문의하세요. 일반적인 CDN 문제 해결에 대한 구현 지침은 AWS 미디어 블로그의 [콘텐츠 전송 네트워크 디버깅을 참조하세요](#).

MediaTailor와 CDN 통합 설정

이 섹션에서는 콘텐츠 전송 네트워크(CDN) AWS Elemental MediaTailor와의 통합에 대한 지침을 제공합니다.

MediaTailor와의 효과적인 CDN 통합은 대규모 맞춤형 광고를 통해 고품질 스트리밍 경험을 제공하는 데 필수적입니다. 이 가이드에서는 CDN 통합을 설정, 구성 및 최적화하는 전체 프로세스를 안내합니다.

자세한 내용은 다음 링크를 참조하세요.

- 권한 부여 및 광고 대상 지정을 위해 CDN의 [MediaTailor 매니페스트 쿼리 파라미터](#).
- 동적 변수 및 구성 별칭을 사용한 고급 라우팅은 섹션을 참조하세요 [ADS 요청에 대한 MediaTailor 동적 광고 변수](#).
- MediaTailor 구성 생성에 대한 자세한 내용은 섹션을 참조하세요 [AWS Elemental MediaTailor를 사용하여 광고 삽입](#).
- CloudFront 배포 생성에 대한 자세한 내용은 CloudFront 개발자 안내서의 [배포 생성](#)을 참조하세요.

이 안내서의 명확성을 보장하기 위해 다음 용어를 사용하여 다양한 유형의 매니페스트를 설명합니다.

- HLS 매니페스트:
 - 다변량 재생 목록: 미디어 재생 목록 링크가 포함된 최상위 매니페스트
 - 미디어 재생 목록: 콘텐츠 세그먼트 링크가 있는 2단계 매니페스트
- DASH 매니페스트:
 - MPD(미디어 프레젠테이션 설명): DASH 매니페스트의 표준 용어

MediaTailor를 CDN과 통합하면 다음과 같은 이점이 있습니다.

- 최종 사용자의 지연 시간 단축
- 트래픽이 많은 이벤트에 대한 확장성 향상
- 중복 전송 경로를 통한 안정성 향상
- 오리진 트래픽을 줄여 비용 최적화

- DDoS 공격에 대한 보호 개선

CDN 통합 구성 요소 및 요구 사항

MediaTailor와 CDN을 성공적으로 통합하려면 다음 주요 구성 요소를 구성해야 합니다.

CDN 라우팅 동작

다양한 유형의 요청(매니페스트, 콘텐츠 세그먼트, 광고 세그먼트)이 CDN을 통해 라우팅되는 방식을 결정하는 규칙입니다.

MediaTailor의 CDN 매핑

매니페스트가 오리진 서버를 직접 참조하는 대신 CDN 도메인을 참조하도록 하는 MediaTailor의 구성입니다.

보안 설정

전송 보안, 액세스 제어 및 모니터링을 포함하여 콘텐츠와 인프라를 보호하는 구성입니다.

테스트 및 검증

프로덕션에 배포하기 전에 CDN 통합이 올바르게 작동하는지 확인하는 절차입니다.

CDN 통합을 위한 사전 조건

CDN 통합을 구성하기 전에 다음이 있는지 확인합니다.

1. 다음 설정을 사용하는 MediaTailor 구성:

- 콘텐츠 소스로서의 콘텐츠 오리진
- 광고 결정 서버로서의 ADS

CDN 통합 단계에서도 오리진 및 ADS URLs이 필요합니다.

2. CDN의 구성 인터페이스에 대한 액세스

3. 동작, 규칙 및 캐시 설정에 대한 CDN의 특정 용어 이해

4. 세그먼트에 사용되는 파일 확장명(예: .ts, .mp4 또는 .m4s)을 포함한 콘텐츠 구조에 대한 지식

CDN 통합 설정 단계

MediaTailor를 CDN과 통합하는 프로세스는 다음과 같은 상위 단계를 따릅니다.

1. CDN 라우팅 동작 구성 - 다양한 유형의 요청을 적절하게 라우팅하도록 CDN을 설정합니다.
2. MediaTailor에서 CDN 매핑 구성 - CDN 도메인 이름을 사용하도록 MediaTailor 구성을 업데이트합니다.
3. 보안 모범 사례 구현 - 콘텐츠와 인프라를 보호하도록 보안 설정을 구성합니다.
4. 통합 테스트 - CDN 통합이 올바르게 작동하는지 확인합니다.

MediaTailor CDN 통합에 필요한 헤더

MediaTailor가 CDN과 올바르게 작동하려면 특정 HTTP 헤더를 전달하도록 CDN을 구성해야 합니다. 이러한 헤더는 압축, 디바이스 감지, 광고 개인화 및 지리 타겟팅을 비롯한 적절한 기능에 필수적입니다.

다음 헤더를 MediaTailor에 전달하도록 CDN을 구성합니다.

Accept-Encoding

용도: 압축 기능에 필요

세부 정보: 이 헤더는 클라이언트가 지원하는 압축 방법을 MediaTailor에 알려줍니다. MediaTailor는 이 정보를 사용하여 가능하면 매니페스트를 압축하여 대역폭 사용량을 줄이고 성능을 개선합니다. 압축을 지원하지 않는 레거시 디바이스는 이 헤더를 전송하지 않으며 MediaTailor는 압축되지 않은 매니페스트를 반환합니다.

User-Agent

용도: 디바이스 감지 및 광고 개인화에 필요

세부 정보: MediaTailor는 User-Agent 헤더를 분석하여 클라이언트 디바이스 유형, 브라우저 및 기능을 식별합니다. 이 정보는 광고 타겟팅, 디바이스별 최적화 및 다양한 재생 클라이언트와의 호환성을 보장하는 데 사용됩니다.

Host

용도: 적절한 요청 라우팅에 필요

세부 정보: Host 헤더는 요청이 올바른 MediaTailor 엔드포인트로 라우팅되도록 합니다. 이는 다중 테넌트 환경과 사용자 지정 도메인 구성을 사용할 때 특히 중요합니다.

Amazon CloudFront를 포함한 많은 CDN들 기본적으로 Host 헤더를 전달하지 않습니다.

CloudFront 사용자의 경우: 구성 지침은 섹션을 참조 [캐시 동작 구성](#)하세요.

X-Forwarded-For

용도: 클라이언트 IP 감지 및 지리 대상 지정에 필요합니다.

세부 정보: 이 헤더는 요청이 CDN을 통과할 때 원래 클라이언트 IP 주소를 보존합니다. MediaTailor는 지리적 광고 타겟팅, 분석 및 리전 콘텐츠 제한 준수에 이 정보를 사용합니다.

Important

전체 MediaTailor 기능에는 네 개의 헤더가 모두 필요합니다. 이러한 헤더가 누락되면 다음과 같은 기능이 저하될 수 있습니다.

- 매니페스트를 압축할 수 없음(수락 인코딩 누락)
- 광고 타겟팅 불량 및 디바이스 호환성 문제(사용자-에이전트 누락)
- 요청 라우팅 실패(호스트 누락)
- 부정확한 지리적 대상 지정 및 분석(X-Forwarded-For 누락)

CDN별 구성 지침은 이 헤더 목록을 참조하는 라우팅 동작 및 캐싱 섹션을 참조하세요.

다음 주제에서는 CDN 통합의 각 측면에 대한 자세한 지침을 제공합니다.

주제

- [MediaTailor에 대한 CDN 라우팅 동작 설정](#)
- [MediaTailor에서 CDN 매핑 설정](#)
- [MediaTailor에 대한 CDN 통합 보안 모범 사례](#)

MediaTailor에 대한 CDN 라우팅 동작 설정

이 섹션에서는 다양한 유형의 요청을 AWS Elemental MediaTailor 통합에 적합하게 라우팅하도록 콘텐츠 전송 네트워크(CDN)를 설정하는 방법을 설명합니다. 적절한 라우팅 구성을 통해 매니페스트 요청, 콘텐츠 세그먼트 및 광고 세그먼트를 올바르게 처리할 수 있습니다.

CDN 라우팅 동작을 구성하는 것은 효율적인 콘텐츠 전송 파이프라인을 생성하는 데 중요한 단계입니다. 다양한 콘텐츠 유형에 대해 특정 라우팅 규칙을 설정하면 캐싱을 최적화하고, 전송 성능을 개선하고, 맞춤형 광고 삽입이 올바르게 작동하는지 확인할 수 있습니다.

동적 변수 및 구성 별칭을 사용하는 고급 라우팅 시나리오는 섹션을 참조하세요 [ADS 요청에 대한 MediaTailor 동적 광고 변수](#). CDN 라우팅 전반의 쿼리 파라미터 보존에 대한 자세한 내용은 섹션을 참조하세요 [MediaTailor 매니페스트 쿼리 파라미터](#).

Important

캐시된 객체 또는 최종 사용자에게 대한 CDN 응답에 CORS 헤더를 포함하지 않으면 재생에 실패할 수 있습니다.

CDN 라우팅 동작 구성

다양한 유형의 요청을 적절하게 라우팅하도록 CDN을 설정합니다.

콘텐츠 세그먼트 라우팅

콘텐츠 세그먼트 라우팅은 실제 콘텐츠 세그먼트에 대한 요청을 오리진 서버로 보냅니다. 광고 세그먼트 라우팅과 마찬가지로 콘텐츠 세그먼트 라우팅에는 웹 기반 플레이어에서 원활한 재생을 보장하기 위한 적절한 CORS 구성도 필요합니다.

자세한 구성 지침은의 CloudFront 예제를 참조하세요 [우선순위 4: 콘텐츠 오리진 경로 동작](#). 이 예제에서는 CloudFront에 대해 따르거나 다른 CDN에 맞게 조정해야 하는 특정 설정을 제공합니다.

콘텐츠 세그먼트 라우팅의 주요 구성 요구 사항은 다음과 같습니다.

- 콘텐츠 세그먼트 파일 확장명(예: *.ts, *.mp4 또는 *.m4s)과 일치하는 경로 패턴 사용
- 콘텐츠 오리진(예: Amazon S3 버킷 또는 MediaPackage 엔드포인트)으로 요청 라우팅
- 최적의 캐시 적중률을 위해 오리진이 캐시 키의 응답을 수정하고 오리진 요청을 전달하도록 하는 쿼리 문자열 파라미터만 포함합니다.
- 24시간 TTL 값을 초과하는 적절한 캐시 정책 적용
- 최종 사용자에게 CORS 응답 헤더 포함

광고 세그먼트 라우팅

광고 세그먼트 라우팅은 시청자에게 개인 맞춤형 광고를 제공하는 데 매우 중요합니다. 광고 세그먼트 라우팅을 구성할 때는 웹 기반 플레이어에서 재생 실패를 일으킬 수 있는 문제를 방지하기 위해 적절한 CORS(교차 오리진 리소스 공유) 처리를 구현해야 합니다.

자세한 구성 지침은의 CloudFront 예제를 참조하세요 [우선순위 0: 광고 세그먼트 경로 동작](#). 이 예제에서는 CloudFront에 대해 따르거나 다른 CDN에 맞게 조정해야 하는 특정 설정을 제공합니다.

광고 세그먼트 라우팅의 주요 구성 요구 사항은 다음과 같습니다.

- MediaTailor 광고 세그먼트 /tm/* 전용 경로 패턴 사용
- 로 요청 라우팅 `segments.mediatailor.region.amazonaws.com`
- 최적의 캐시 적중률을 위해 캐시 키 또는 오리진 요청에 최종 사용자 요청 헤더, 쿠키 또는 쿼리 문자열 파라미터를 포함하지 마세요.
- 24시간 TTL 값을 초과하는 적절한 캐시 정책 적용
- 최종 사용자에게 CORS 응답 헤더 포함

매니페스트 요청 라우팅

다변량 재생 목록, 미디어 재생 목록 및 MPD 요청을 MediaTailor로 라우팅하려면 다음 일반 설정을 사용합니다. CloudFront 구성은 섹션을 참조하세요 [매니페스트 캐시 동작 구성](#).

1. CDN 구성 인터페이스에서 다양한 매니페스트 유형에 대한 동작을 생성합니다.
2. 다변량 재생 목록 및 미디어 재생 목록 파일 확장명(*.m3u8HLS의 경우) 및 MPD 파일 확장명(*.mpdDASH의 경우)과 일치하도록 경로 패턴을 설정합니다.
3. MediaTailor 구성 엔드포인트를 가리키도록 CDN에서 오리진 설정을 구성합니다.
4. 광고 삽입의 경우 개인화된 다변량 재생 목록, 미디어 재생 목록 및 MPDs. 광고 삽입은 개인화된 매니페스트를 제공하므로 CDN이 이를 캐싱해서는 안 됩니다. 의도한 재생 디바이스와 다른 재생 디바이스가 캐시된 재생 목록 또는 MPD를 수신하면 재생 또는 추적에 문제가 발생할 수 있습니다. 모든 콘텐츠 유형에 대한 TTL 권장 사항을 포함한 포괄적인 캐싱 지침은 섹션을 참조하세요 [CDN 및 MediaTailor 통합을 위한 캐싱 최적화](#).
5. 모든 헤더에 대한 헤더 전달을 구성합니다. 최소 요구 사항은 섹션을 참조하세요 [MediaTailor CDN 통합에 필요한 헤더](#).
6. 쿼리 문자열 전달을 활성화하여 광고 대상 지정 파라미터를 전달합니다.

HLS multivariant playlist

HLS 다변량 재생 목록 요청은 다음 형식을 따릅니다.

```
https://<playback-endpoint>/v1/index/<hashed-account-id>/<origin-id>/<index>.m3u8
```

예제:

```
https://777788889999.mediatailor.us-east-1.amazonaws.com/v1/master/
a1bc06b59e9a570b3b6b886a763d15814a86f0bb/Demo/assetId.m3u8
```

HLS media playlist

HLS 미디어 재생 목록 요청은 다음 형식을 따릅니다.

```
https://<playback-endpoint>/v1/manifest/<hashed-account-id>/<session-id>/
<manifestNumber>.m3u8
```

에 대한 플레이어 요청은 요청의 키워드를 `https://mediatailor.us-west-2.amazonaws.com/v1/session/configuration/endpoint` 기반으로 MediaTailor 경로 `*.m3u8`로 라우팅 `https://CDN_Hostname/some/path/asset.m3u8`됩니다.

예제:

```
https://777788889999.mediatailor.us-east-1.amazonaws.com/v1/manifest/
a1bc06b59e9a570b3b6b886a763d15814a86f0bb/c240ea66-9b07-4770-8ef9-7d16d916b407/0.m3u8
```

DASH MPD

DASH MPD 요청은 다음 형식을 따릅니다.

```
https://<playback-endpoint>/v1/dash/<hashed-account-id>/<origin-id>/<assetName>.mpd
```

에 대한 플레이어 요청은 요청의 키워드를 `https://mediatailor.us-west-2.amazonaws.com/v1/dash/configuration/endpoint` 기반으로 MediaTailor 경로 `*.mpd`로 라우팅 `https://CDN_Hostname/some/path/asset.mpd`됩니다.

예제:

```
https://777788889999.mediatailor.us-east-1.amazonaws.com/v1/dash/
a1bc06b59e9a570b3b6b886a763d15814a86f0bb/Demo/0.mpd
```

CDN 라우팅 모범 사례

CDN 라우팅 동작을 구성할 때 최적의 성능과 신뢰성을 보장하려면 다음 모범 사례를 따르세요.

특정 경로 패턴 사용

콘텐츠 구조와 정확하게 일치하는 특정 경로 패턴을 생성하여 적절한 라우팅을 보장합니다.

동작 순서 우선 순위 지정

대부분의 CDN에서는 동작이 순서대로 평가됩니다. 보다 일반적인 동작 앞에 보다 구체적인 동작을 배치합니다.

동작 패턴 테스트

프로덕션에 배포하기 전에 경로 패턴이 예상 요청과 올바르게 일치하는지 확인합니다.

구성 문서화

CDN 라우팅 동작에 대한 문서를 유지 관리하여 문제 해결 및 향후 업데이트를 용이하게 합니다.

다음 단계

CDN 라우팅 동작을 구성한 후 다음 단계는 MediaTailor에서 CDN 매핑을 구성하는 것입니다. 자세한 내용은 [MediaTailor에서 CDN 매핑 설정](#) 섹션을 참조하세요.

MediaTailor에서 CDN 매핑 설정

이 섹션에서는 콘텐츠 전송 네트워크(CDN) 도메인 이름을 AWS Elemental MediaTailor 사용하도록 구성하는 방법을 설명합니다. CDN 라우팅 동작을 설정한 후에는 매니페스트가 오리진 서버를 직접 참조하는 대신 CDN 도메인을 참조하도록 MediaTailor 구성을 업데이트해야 합니다.

MediaTailor에서 CDN 매핑을 구성하면 매니페스트의 모든 콘텐츠 및 광고 세그먼트 URLs이 오리진 서버를 직접 가리키지 않고 CDN을 가리킵니다. 이 단계는 완전한 CDN 전송 체인을 생성하고 CDN 통합의 이점을 극대화하는 데 필수적입니다.

MediaTailor의 CDN 매핑 구성

CDN 라우팅 동작을 설정한 후 CDN 도메인 이름을 사용하도록 MediaTailor를 구성합니다.

1. [MediaTailor 콘솔](#)을 엽니다.
2. 업데이트할 구성을 선택합니다.
3. CDN 구성 섹션의 고급 설정에서 CDN 콘텐츠 세그먼트 접두사 필드에 CDN 도메인 이름을 입력합니다.

4. 광고 세그먼트에 별도의 CDN 도메인을 사용하는 경우 CDN 광고 세그먼트 접두사 필드에 해당 도메인을 입력합니다.
5. 변경 내용을 저장합니다.

이 구성을 사용하면 MediaTailor가 오리진 또는 광고 세그먼트 스토리지를 직접 가리키는 대신 CDN을 가리키는 URLs을 사용하여 매니페스트를 생성할 수 있습니다.

기본 URL 동작 이해

MediaTailor는 CDN 접두사 구성을 기반으로 매니페스트의 기본 URLs을 결정합니다.

- CDN 광고 세그먼트 접두사 구성됨: 광고 세그먼트는 CDN 접두사를 기본 URL로 사용합니다.
- CDN 광고 세그먼트 접두사가 구성되지 않음: 광고 세그먼트는 직접 MediaTailor 위치를 기본 URL로 사용합니다.
- CDN 콘텐츠 세그먼트 접두사 구성됨: 콘텐츠 세그먼트는 CDN 접두사를 기본 URL로 사용합니다.
- CDN 콘텐츠 세그먼트 접두사가 구성되지 않음: 콘텐츠 세그먼트는 원래 콘텐츠 오리진을 참조합니다.

DASH BaseURL 처리

DASH 매니페스트의 경우 MediaTailor는 콘텐츠 및 광고 세그먼트에 대한 BaseURL 설정을 다르게 관리합니다.

콘텐츠 세그먼트:

- CDN 콘텐츠 세그먼트 접두사 사용: MediaTailor는 지정된 접두사를 사용하여 BaseURL MPD 수준에서 정확히 하나를 설정합니다.
- CDN 콘텐츠 세그먼트 접두사 사용 안 함: MediaTailor는 오리진 매니페스트에서 기존 설정을 보존하거나 오리진 MPD URL에 따라 기존 BaseURL 설정을 추가합니다.

광고 세그먼트:

- CDN 광고 세그먼트 접두사 사용: 각 광고 기간은 구성된 접두사를 BaseURL 사용하여 정확히 하나씩 가져옵니다.
- CDN 광고 세그먼트 접두사 사용 안 함: 각 광고 기간은 MediaTailor 광고 콘텐츠 서버를 정확히 한번 BaseURL 가리킵니다.

Example CDN 매핑 예제

콘텐츠 오리진이 `http://origin.com/contentpath/` 이고 CDN 콘텐츠 세그먼트 접두사가 인 `https://cdn.example.com/` 경우 일반적으로 로 참조되는 콘텐츠 세그먼트 `http://origin.com/contentpath/subdir/content.ts`는 매니페스트에 로 표시됩니다 `https://cdn.example.com/subdir/content.ts`.

중요 고려 사항

MediaTailor에서 CDN 매핑을 구성할 때는 다음 중요 고려 사항을 염두에 두세요.

CDN 접두사에 HTTPS 사용

항상 CDN 접두사에 HTTPS URLs을 사용하여 안전한 콘텐츠 전송을 보장합니다.

CDN 동작 경로 일치

MediaTailor에서 구성하는 CDN 접두사가 CDN 동작에서 구성한 경로 패턴과 일치하는지 확인합니다.

리전 CDNs 고려

리전마다 다른 CDN 도메인을 사용하는 경우 각 리전에 대해 별도의 MediaTailor 구성을 생성해야 합니다.

도메인 소유권 확인

MediaTailor에서 구성하는 CDN 도메인을 제어할 수 있는지 확인합니다.

CDN 매핑 구성 확인

MediaTailor에서 CDN 매핑을 구성한 후 구성이 올바르게 작동하는지 확인합니다.

1. CDN을 통해 매니페스트를 요청합니다.
2. 매니페스트 콘텐츠를 검사하여 세그먼트 URLs이 CDN 도메인을 참조하는지 확인합니다.
3. 매니페스트의 콘텐츠 세그먼트 URLs CDN 도메인을 가리키는지 확인합니다.
4. 매니페스트의 광고 세그먼트 URLs CDN 도메인을 가리키는지 확인합니다.

포괄적인 테스트 및 검증 절차는 섹션을 참조하세요 [CDN 및 MediaTailor 통합 테스트 및 검증](#).

다음 단계

MediaTailor에서 CDN 매핑을 구성한 후 다음 단계는 CDN 통합에 대한 보안 모범 사례를 구현하는 것입니다. 자세한 내용은 [MediaTailor에 대한 CDN 통합 보안 모범 사례](#) 섹션을 참조하세요.

MediaTailor에 대한 CDN 통합 보안 모범 사례

이 섹션에서는 AWS Elemental MediaTailor 콘텐츠 전송 네트워크(CDN) 통합에 대한 보안 모범 사례를 구현하는 방법을 설명합니다. 적절한 보안 구성은 콘텐츠와 인프라를 무단 액세스 및 잠재적 위협으로부터 보호하는 데 도움이 됩니다.

CDN 통합을 보호하는 것은 콘텐츠를 보호하고, 무단 액세스를 방지하고, 보안 요구 사항을 준수하는 데 매우 중요합니다. 이러한 모범 사례를 구현하면 스트리밍 워크플로에 대한 강력한 보안 태세를 구축하는 데 도움이 됩니다.

CDN 보안 구성

CDN 라우팅 및 캐시 설정을 구성하기 전에 다음 보안 모범 사례를 구현하여 콘텐츠와 인프라를 보호합니다.

전송 보안

MediaTailor는 모든 통신에 HTTPS만 사용하며 HTTP 연결을 허용하지 않습니다. 스트리밍 워크플로의 모든 구성 요소 간에 보안 통신을 보장하려면 다음 단계를 완료하세요.

1. CDN과 MediaTailor 엔드포인트 간의 모든 통신을 위해 HTTPS를 구성합니다. MediaTailor에는 HTTPS가 필요하며 HTTP 연결을 허용하지 않습니다.
2. 모든 HTTPS 연결에 TLS 1.2 이상을 사용합니다.
3. 최종 사용자의 HTTPS 전용 연결을 적용하도록 CDN을 구성합니다.

액세스 관리

콘텐츠와 인프라를 무단 액세스로부터 보호하려면 다음 액세스 제어를 구현합니다.

1. 콘텐츠 액세스를 특정 리전으로 제한해야 하는 경우 지리적 제한 설정을 구성합니다.
2. 최종 사용자 인증이 필요한 콘텐츠에 대해 서명된 URLs 또는 서명된 쿠키를 구현합니다.
3. CDN 구성에 대한 관리 액세스를 위한 IP 기반 허용 목록을 설정합니다.

보안 모니터링

보안 이벤트를 효과적으로 탐지하고 대응하려면 다음 모니터링 사례를 구현합니다.

1. CDN 배포에 대한 액세스 로깅을 활성화합니다.
2. 비정상적인 트래픽 패턴 또는 액세스 시도에 대한 알림을 설정합니다.
3. 보안 구성을 정기적으로 검토하고 필요에 따라 업데이트합니다.

고급 보안 기능

다음 고급 보안 기능은 CDN 통합을 위한 향상된 보호를 제공합니다.

웹 애플리케이션 방화벽(WAF)

CDN이 WAF 기능을 지원하는 경우 일반적인 웹 취약성 및 공격으로부터 보호하도록 구성합니다.

DDoS 보호

CDN에서 제공하는 DDoS 보호 기능을 활성화하여 분산 denial-of-service 공격을 완화합니다.

오리진 액세스 제어

직접 액세스 시도를 방지하기 위해 오리진 서버가 CDN의 요청만 수락하도록 구성합니다.

콘텐츠 암호화

매우 민감한 콘텐츠의 경우 전송 보안 이외의 추가 암호화 메커니즘을 구현하는 것이 좋습니다.

다음 단계

보안 모범 사례를 구현한 후 다음 단계는 CDN 통합을 테스트하고 문제를 해결하는 것입니다. 포괄적인 테스트 및 문제 해결 지침은 섹션을 참조 [CDN 통합 문제 해결](#)하세요.

맞춤형 비디오 광고를 위해 CDN으로 SSAI 설정

이 섹션에서는 AWS Elemental MediaTailor 서버 측 광고 삽입(SSAI)을 콘텐츠 전송 네트워크(CDN)와 통합하기 위한 포괄적인 지침을 제공합니다. 다음 단계에 따라 SSAI CDN 통합을 설정, 구성 및 최적화합니다.

서버 측 광고 삽입(SSAI)은 클라이언트 수준이 아닌 서버 수준에서 비디오 스트림에 개인화된 광고를 원활하게 삽입하는 기술입니다. CDN과 함께 사용하면 지연 시간을 최소화하면서 전 세계 고객에게 맞춤형 광고를 제공할 수 있는 강력하고 확장 가능한 솔루션을 만들 수 있습니다.

이 주제에서는 매니페스트라는 용어를 사용하여 다변량 재생 목록, 미디어 재생 목록 및 MPDs.

필요한 사항

CDN을 사용하여 MediaTailor 광고 삽입을 설정하기 전에 다음과 같은 필수 리소스를 수집합니다.

AWS 계정 및 권한

MediaTailor 리소스를 생성하고 관리할 수 있는 적절한 권한이 있는 AWS 계정

MediaTailor, CloudFront(사용하는 경우) 및 관련 서비스에 대한 IAM 권한

자세한 권한 요구 사항은 섹션을 참조하세요 [의 보안 AWS Elemental MediaTailor](#).

필수 서비스

AWS Elemental MediaTailor 구성 및 실행

콘텐츠 전송 네트워크(CDN) 계정(Amazon CloudFront 또는 타사 CDN)

콘텐츠의 오리진 서버(HLS 또는 DASH)

VAST 또는 VMAP를 지원하는 광고 결정 서버(ADS)

콘텐츠 요구 사항

HLS 또는 DASH 형식으로 적절하게 인코딩되고 패키징된 콘텐츠

콘텐츠의 광고 중단 마커(VOD용) 또는 SCTE-35 마커(라이브용)

시작하기 전 준비 사항

CDN을 사용하여 MediaTailor 광고 삽입을 구현하기 전에 다음 설정 작업을 완료합니다.

1. CDN, MediaTailor 및 오리진 서버 간의 네트워크 연결 구성
2. 보안 콘텐츠 전송을 위한 HTTPS 설정
3. CDN 도메인에 대한 DNS 설정 구성
4. 기본 MediaTailor 설정의 경우 [설정](#) 및의 단계를 완료합니다 [MediaTailor 광고 삽입 시작하기](#).

지식 사전 조건

이 솔루션을 성공적으로 구현하려면 다음이 필요합니다.

- 스트리밍 프로토콜 이해(HLS/DASH)
- CDN 구성 원칙에 대한 기본 지식
- 광고 삽입 개념에 대한 지식

CDN 통합의 이점

SSAI를 CDN과 통합하면 다음과 같은 주요 이점이 있습니다.

시청자 경험 개선

적절하게 구성된 CDN은 광고 전환 중에 버퍼링, 시작 시간 및 재생 오류를 줄입니다. 이로 인해 시청자의 참여와 만족도가 높아집니다.

비용 절감

효율적인 캐싱 전략은 오리진 요청을 최소화합니다. 이렇게 하면 데이터 전송 비용과 오리진 서버로드가 줄어들며, 특히 대용량 광고 지원 콘텐츠에 중요합니다.

확장성

최적화된 CDN 구성은 성능 저하 없이 인기 있는 이벤트 중에 트래픽 급증을 처리하여 최대 시청 시간 중에도 맞춤형 광고가 전달되도록 합니다.

글로벌 도달 범위

적절하게 구성된 CDN은 위치에 관계없이 전 세계 시청자에게 짧은 지연 시간으로 콘텐츠를 제공하여 잠재적 시청자를 확장합니다.

원활한 광고 전환

최적화된 CDN 구성은 콘텐츠와 광고 간의 원활한 전환을 보장하여 브로드캐스트 품질의 시청 환경을 만듭니다.

다음 주제에서는 최적의 성능을 위해 MediaTailor를 CDN으로 구성하는 방법에 대한 포괄적인 지침을 제공합니다.

주제

- [CDN 및 MediaTailor 통합을 위한 광고 삽입 아키텍처 이해](#)
- [최적의 광고 전송을 위해 CDN을 사용하여 기본 MediaTailor SSAI 설정](#)
- [MediaTailor SSAI를 수익화된 선형 채널의 채널 어셈블리와 통합](#)

- [CDN 및 MediaTailor 통합을 위한 성능 최적화](#)
- [CDN 및 MediaTailor 통합에 대한 작업 모니터링](#)
- [중단 없는 광고 전송을 위해 CDNs 사용한 MediaTailor SSAI 문제 해결](#)

CDN 및 MediaTailor 통합을 위한 광고 삽입 아키텍처 이해

이 섹션에서는 콘텐츠 전송 네트워크(CDNs)를 사용하는 서버 측 광고 삽입(SSAI)의 개념과 아키텍처를 설명합니다 AWS Elemental MediaTailor. 동적 광고 삽입 및 매니페스트 조작이 함께 작동하여 효과적인 비디오 수익화를 가능하게 하는 방법을 알아봅니다.

MediaTailor를 사용한 서버 측 광고 삽입(SSAI)을 통해 다음을 수행할 수 있습니다.

- 정의된 광고 중단점에서 비디오 스트림에 개인 맞춤형 광고 삽입
- 최종 사용자 데이터를 기반으로 광고를 정확하게 대상으로 지정
- 클라이언트 측 광고 삽입 기술의 필요성 제거

CDN과 결합하면 성능 및 확장성이 향상된 시청자에게 이러한 개인화된 스트림을 제공하여 비디오 수익화 전략을 개선할 수 있습니다.

CDN을 사용한 광고 삽입을 위한 권장 아키텍처는 최종 사용자와 광고 삽입 간에 CDN을 배치하고 광고 삽입은 오리진에서 직접 콘텐츠에 액세스합니다. 이 아키텍처는 콘텐츠 전송과 비디오 수익화 모두에 대해 다음과 같은 이점을 제공합니다.

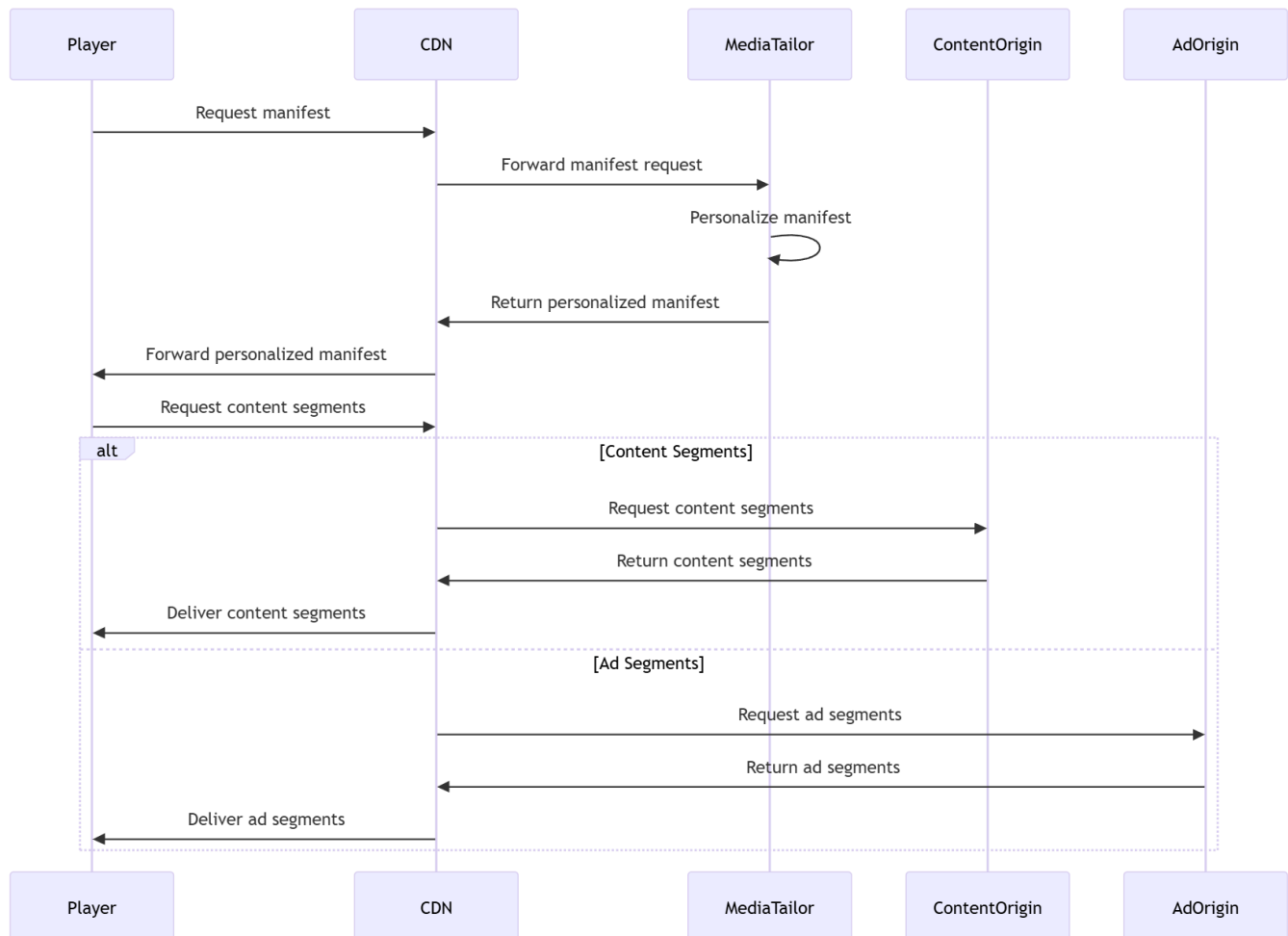
- 콘텐츠 및 광고 세그먼트의 효과적인 캐싱
- MediaTailor에서 요청 로드 감소
- 최종 사용자에게 대한 전송 속도 개선
- 간소화된 URL 관리
- 디바이스 간 맞춤형 광고의 일관된 제공

이 권장 아키텍처에서:

1. CDN의 최종 사용자 요청 매니페스트
2. CDN은 요청을 광고 삽입에 전달합니다.
3. 광고 삽입은 오리진에서 콘텐츠 매니페스트를 요청합니다.
4. 광고 삽입은 광고 결정 서버(ADS)에서 광고를 요청합니다.

5. 광고 삽입은 광고 마커(오리진 매니페스트)를 특정 뷰어(ADS)의 대상 광고 세그먼트를 가리키는 URL로 대체하여 매니페스트를 개인화합니다.
6. 광고 삽입은 광고 세그먼트 URL이 포함된 개인화된 매니페스트를 CDN에 반환하여 최종 사용자에게 전달합니다.
7. 최종 사용자는 CDN을 통해 세그먼트를 요청합니다.
8. CDN은 세그먼트 유형에 따라 세그먼트 요청을 라우팅합니다.
 - 콘텐츠 세그먼트 요청은 콘텐츠 오리진으로 이동합니다.
 - 광고 세그먼트 요청은 MediaTailor로 이동합니다.

이 아키텍처는 CDN 사용의 보안 및 유연성 이점을 유지하면서 최적의 성능을 보장합니다.



Note

이 흐름은 VOD 콘텐츠와 라이브 콘텐츠 간에 약간 다릅니다. VOD의 경우 매니페스트를 더 오래 캐시할 수 있지만 라이브 콘텐츠를 사용하려면 스트림 연속성을 유지하기 위해 매니페스트 업데이트가 더 자주 필요합니다.

VOD와 라이브 콘텐츠 캐싱의 주요 차이점:

VOD 콘텐츠

매니페스트는 자주 변경되지 않으므로 더 긴 TTL 값(분~시간)을 설정합니다.

라이브 콘텐츠

최종 사용자가 최신 스트림 세그먼트를 수신하도록 매니페스트에 더 짧은 TTL 값(초) 설정

콘텐츠 오리진과 사이에 CDN을 배치하지 않는 것이 좋습니다 AWS Elemental MediaTailor. 이렇게 하면 다음과 같은 몇 가지 기술적 문제가 발생할 수 있습니다.

캐시 키 충돌

쿼리 파라미터를 올바르게 처리하도록 CDN을 구성합니다. 이렇게 하면 쿼리 파라미터가 다른 동일한 매니페스트를 요청할 때 MediaTailor가 잘못된 매니페스트를 수신하지 못합니다.

Gzip 압축 문제

매니페스트 구문 분석 오류가 발생하는 경우 CDN이 올바른 형식의 매니페스트를 MediaTailor에 전달해야 합니다. 일부 CDN은 구문 분석 실패를 일으킬 수 있는 손상된 gzip 페이로드를 제공할 수 있습니다. 이 경우 워크플로의 다른 곳에서 비용 절감을 위해 압축을 유지하면서 CDN과 MediaTailor 간의 압축을 비활성화해야 할 수 있습니다.

매니페스트 신선도

라이브 스트림의 경우 현재 매니페스트를 MediaTailor에 전달하도록 CDN을 구성합니다. 이렇게 하면 콘텐츠와 광고 간의 동기화 문제가 방지됩니다.

성능 최적화

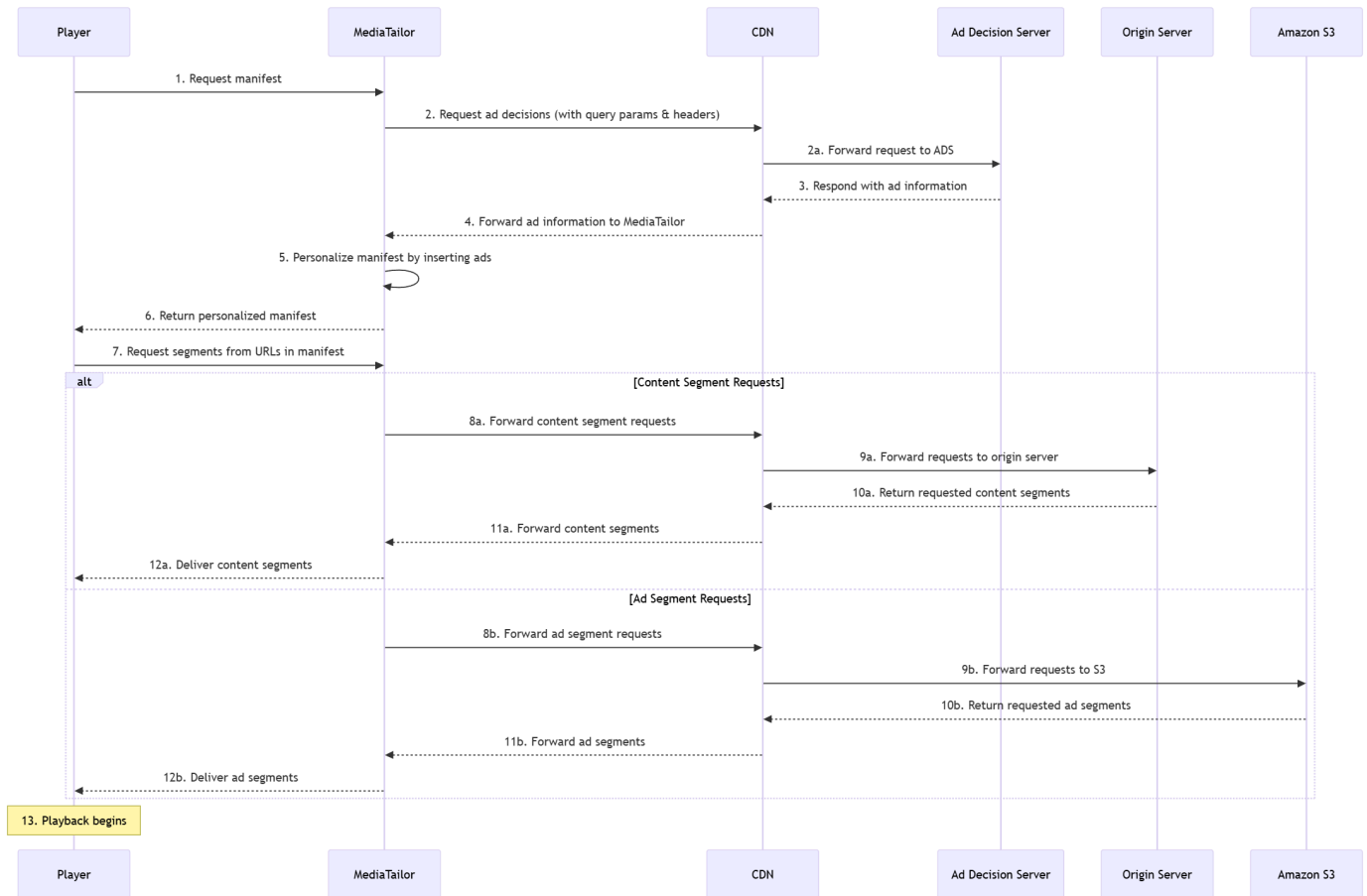
네트워크 홉과 잠재적인 캐시 누락을 최소화하여 재생 시작 시간을 줄입니다.

캐시 관리

매니페스트가 자주 업데이트되는 라이브 콘텐츠에 대해 간소화된 캐시 무효화 전략을 구현합니다.

이 최적화되지 않은 아키텍처에서:

1. 최종 사용자는에서 직접 다변량 재생 목록, 미디어 재생 목록 또는 MPDs 요청합니다 AWS Elemental MediaTailor.
2. MediaTailor는 CDN을 통해 콘텐츠 매니페스트(다변량 재생 목록, 미디어 재생 목록 또는 MPDs)를 요청합니다.
3. CDN은 오리진 서버에 요청을 전달합니다.
4. 오리진 서버는 다변량 재생 목록, 미디어 재생 목록 또는 MPDs에 반환합니다.
5. CDN은 다변량 재생 목록, 미디어 재생 목록 또는 MPDs를 MediaTailor에 전달합니다.
6. MediaTailor는 광고 결정 서버(ADS)에서 광고를 요청합니다.
7. MediaTailor는 다변량 재생 목록, 미디어 재생 목록 또는 MPDs에 광고를 삽입하여 매니페스트를 개
인화하고 최종 사용자에게 직접 전달합니다.
8. 이 아키텍처는 추가 지연 시간, 잠재적 캐싱 문제를 발생시키고 문제 해결을 복잡하게 만듭니다.



요청 및 응답 흐름

CDN을 사용하여 동적 광고 삽입을 구현할 때 요청 및 응답 흐름을 지원하도록 시스템을 구성합니다.

1. MediaTailor를 매니페스트 오리진으로 사용하여 CDN에서 다변량 재생 목록(HLS) 또는 MPDs(DASH)를 요청하도록 플레이어 구성합니다.
2. 모든 쿼리 파라미터 및 헤더를 포함하여 모든 다변량 재생 목록, 미디어 재생 목록 및 MPD 요청을 MediaTailor에 전달하도록 CDN을 설정합니다.
3. MediaTailor가 광고 결정 서버(ADS)와 통신하여 쿼리 파라미터 및 헤더를 전달할 수 있는지 확인합니다.
4. 쿼리 파라미터를 사용하여 삽입할 광고를 결정하도록 ADS를 구성합니다.

5. MediaTailor 재생 구성에서 CDN 접두사를 설정하여 MediaTailor가 콘텐츠 및 광고 세그먼트 URL 접두사로 CDN 도메인 이름을 대체할 수 있도록 합니다.
6. MediaTailor에서 요청 플레이어로 개인화된 다변량 재생 목록, 미디어 재생 목록 및 MPDs를 전달하도록 CDN을 구성합니다.
7. 세그먼트 URLs 번역하도록 CDN을 설정하고 콘텐츠 세그먼트 요청을 오리진 서버로 전달하고 광고 요청을 MediaTailor가 트랜스코딩된 광고를 저장하는 Amazon S3 버킷으로 전달합니다.

광고 삽입에 대한 CDN 용어

이러한 주요 용어를 이해하면 광고 삽입 CDN 통합을 구현하고 문제를 해결하는 데 도움이 됩니다.

오리진 CDN 및 엣지 CDN

오리진 CDN: MediaTailor와 콘텐츠 오리진 사이에 위치한 CDN입니다. 오리진 서버의 부하를 줄이기 위해 콘텐츠 세그먼트를 캐싱합니다. 다중 CDN 아키텍처에서이 계층은 오리진과 직접 인터페이스하는 첫 번째 CDN 계층입니다.

엣지 CDN: 최종 사용자와 MediaTailor 사이에 위치한 CDN입니다. 시청자에게 개인화된 매니페스트와 콘텐츠를 제공합니다. 다중 CDN 아키텍처에서이 계층은 최종 사용자와 직접 인터페이스하는 가장 바깥쪽의 CDN 계층입니다.

CDN 구성 용어

캐시 동작: CDN이 다양한 유형의 요청을 처리하는 방법을 결정하는 규칙입니다. 이러한 규칙은 다음과 같습니다.

- 캐싱 기간 설정
- 오리진 라우팅 구성
- 요청 처리 파라미터

TTL(Time To Live): 오리진에서 새로 고쳐야 하기 전에 CDN 캐시에서 콘텐츠가 유효한 상태로 유지되는 기간입니다.

캐시 키: CDN이 캐시된 콘텐츠를 저장하고 검색하는 데 사용하는 고유 식별자입니다. 여기에는 일반적으로 다음이 포함됩니다.

- URL 경로
- 쿼리 파라미터
- 선택한 헤더

오리진 실드: CDN 엣지 로케이션과 오리진 서버 간의 중간 캐싱 계층입니다. 오리진에 대한 요청 수를 줄입니다.

요청 축소: 동일한 콘텐츠에 대한 여러 동시 요청을 단일 오리진 요청으로 결합하는 CDN 기능입니다.

MediaTailor별 CDN 용어

CDN 콘텐츠 세그먼트 접두사: 매니페스트의 콘텐츠 세그먼트에 대한 URLs 생성할 때 AWS Elemental MediaTailor 에서 사용하는 CDN 도메인 이름입니다.

CDN 광고 세그먼트 접두사: MediaTailor가 매니페스트의 광고 세그먼트에 대한 URLs을 생성할 때 사용하는 CDN 도메인 이름입니다.

MediaTailor를 사용한 CDN 구성에 대한 자세한 내용은 섹션을 참조하세요 [CDN 통합 설정](#).

Note

이러한 용어는 채널 어셈블리 설명서에 사용된 용어와 일치합니다. 채널 어셈블리 용어는 섹션을 참조하세요 [채널 어셈블리에 대한 CDN 용어](#).

최적의 광고 전송을 위해 CDN을 사용하여 기본 MediaTailor SSAI 설정

이 섹션에서는 비디오 수익화 워크플로를 최적화하기 위해 콘텐츠 전송 네트워크(CDN)를 사용하여 AWS Elemental MediaTailor 동적 광고 삽입을 구성하는 step-by-step 지침을 제공합니다.

동적 변수를 사용한 고급 광고 서버 구성은 섹션을 참조하세요 [ADS 요청에 대한 MediaTailor 동적 광고 변수](#). 광고 대상 지정에 대해 CDN을 [MediaTailor 매니페스트 쿼리 파라미터](#).

CDN을 [CDN 및 MediaTailor 통합을 위한 광고 삽입 아키텍처 이해](#).

사전 조건

CDN을 사용하여 광고 삽입을 설정하기 전에 다음을 확인해야 합니다.

- 활성 AWS Elemental MediaTailor 구성
- 적절한 광고 마커가 있는 HLS 또는 DASH 콘텐츠를 제공하는 콘텐츠 오리진 서버

광고 마커에 대한 자세한 내용은 섹션을 참조하세요 [광고 삽입 동작 이해](#).

- 광고 타겟팅을 위해 VAST 또는 VMAP를 지원하는 광고 결정 서버(ADS)
- CDN 계정(예: Amazon CloudFront 또는 다른 CDN 공급자)
- 매니페스트 조작 및 동적 광고 삽입 개념에 대한 기본 지식

1단계: 최적의 광고 전송을 위해 CDN 캐싱 구성

비디오 수익화 워크플로의 성능을 최적화하려면 적절한 CDN 캐싱 구성이 중요합니다. 캐싱 요구 사항은 서버 측 광고 삽입(SSAI)과 서버 안내 광고 삽입(SGAI) 간에 다릅니다. 다음 권장 설정을 사용하여 콘텐츠와 개인 맞춤형 광고를 효율적으로 제공할 수 있습니다.

SSAI CDN 캐싱 설정

서버 측 광고 삽입 워크플로의 경우 최적의 성능을 위해서는 적절한 캐싱 구성이 중요합니다. SSAI에는 세그먼트가 효율적으로 캐싱되는 동안 개인화된 매니페스트가 캐싱되지 않도록 특정 TTL 값과 캐시 키 설정이 필요합니다.

TTL 값, 경로 패턴 및 캐시 키 구성을 포함한 자세한 SSAI 캐싱 설정은 CDN 최적화 가이드 [서버 측 광고 삽입\(SSAI\) 캐싱](#)의 섹션을 참조하세요.

SSAI의 주요 캐싱 원칙:

- 매니페스트: 맞춤형 콘텐츠의 캐싱을 방지하기 위해 TTL을 0초로 설정
- 세그먼트: 오리진 로드를 줄이기 위해 공격적으로 캐싱(24시간 이상)
- 캐시 키: 매니페스트의 모든 쿼리 파라미터 포함, 세그먼트의 URL 경로만 포함

SGAI CDN 캐싱 설정

서버 가이드 광고 삽입 워크플로의 경우 SGAI 매니페스트를 단기간 캐시하면서 맞춤형 광고 경험을 제공할 수 있으므로 캐싱 요구 사항은 SSAI와 다릅니다.

VOD 및 라이브 TTL 값을 포함한 포괄적인 SGAI 캐싱 설정은 최적화 가이드의 캐싱 테이블을 참조하세요. SGAI를 사용하면 광고 개인화 기능을 유지하면서 SSAI보다 캐시 효율성을 높일 수 있습니다.

주요 SGAI 캐싱 차이점:

- 매니페스트: 단기 캐시 가능(VOD의 경우 5~30분, 라이브의 경우 2~10초)
- 세그먼트: SSAI처럼 공격적으로 캐시(대부분의 콘텐츠에서 24시간 이상)

- 성능 이점: 캐시 가능한 매니페스트로 인해 SSAI보다 캐시 적중률이 우수함

Amazon CloudFront의 경우 다양한 TTL 값과 캐시 키 정책이 있는 캐시 동작을 사용하여 이러한 설정을 구현할 수 있습니다. 다른 CDN스 경우 유사한 캐싱 규칙을 구현하기 위한 특정 설명서를 참조하세요.

2단계: 하이브리드 접근 방식 구현(필요한 경우)

아키텍처에 콘텐츠 오리진과 MediaTailor 간에 별도의 CDN 또는 캐싱 계층이 있는 하이브리드 접근 방식이 필요한 경우:

1. CDN 구성에서 문제를 명확하게 분리합니다.
2. 이전 단원에서 설명한 기술적 문제를 방지하려면 특정 CDN 설정을 구성합니다.
3. 구성을 철저히 테스트하여 매니페스트 개인화 기능이 올바르게 작동하는지 확인합니다.
4. 성능 지표를 모니터링하여 다변량 재생 목록, 미디어 재생 목록, MPDs 및 세그먼트를 최적으로 제공할 수 있습니다.

하이브리드 접근 방식을 구현할 때는 다음과 같은 특정 구성을 고려하세요.

- 콘텐츠 오리진과 MediaTailor 간의 CDN의 경우:
 - 원본에서 원래 압축 상태를 보존하도록 매니페스트 파일에 대한 압축 패스스루 구성
 - 캐시 키에 모든 쿼리 파라미터 포함
 - 라이브 콘텐츠 매니페스트에 대한 짧은 TTL 값 설정
- MediaTailor와 뷰어 간의 CDN의 경우:
 - 광고 세그먼트에 대해 더 긴 캐시 시간 구성
 - 개인화된 매니페스트s 설정
 - 콘텐츠 및 광고 세그먼트에 대한 적절한 오리진 라우팅 구현

3단계: CDN 설정 완료

아키텍처를 선택하고 요청 흐름을 이해한 후의 세부 구성 단계에 따라 설정을 완료합니다 [CDN 통합 설정](#).

특정 CDN 공급자의 경우 다음 추가 리소스를 참조하세요.

- Amazon CloudFront: CloudFront별 구성 단계는 섹션을 참조 [CloudFront 통합](#)하세요.

- 기타 CDNs: 이 가이드에 설명된 일반 원칙을 적용하여 특정 CDN의 구성 옵션에 맞게 조정

4단계: 구성 확인

CDN 설정을 완료한 후 동적 광고 삽입 워크플로가 올바르게 작동하는지 확인합니다.

1. 샘플 플레이어를 사용하여 CDN을 통해 재생 테스트
2. 맞춤형 광고가 지정된 광고 중단점에 올바르게 삽입되었는지 확인
3. CDN 로그를 확인하여 적절한 요청 라우팅 확인
4. 캐시 적중률을 모니터링하여 콘텐츠 및 광고 세그먼트 모두에서 최적의 성능을 보장합니다.
5. 광고 대상 지정 파라미터가 워크플로를 통해 제대로 전달되고 있는지 확인

포괄적인 테스트 및 검증 절차는 섹션을 참조하세요 [CDN 및 MediaTailor 통합 테스트 및 검증](#). SSAI 구현 모니터링에 대한 자세한 내용은 섹션을 참조하세요 [CDN 및 MediaTailor 통합에 대한 작업 모니터링](#). 성능을 최적화하려면 섹션을 참조하세요 [CDN 및 MediaTailor 통합을 위한 성능 최적화](#).

MediaTailor SSAI를 수익화된 선형 채널의 채널 어셈블리와 통합

이 주제에서는 AWS Elemental MediaTailor 서버 측 광고 삽입을 채널 어셈블리 및 콘텐츠 전송 네트워크(CDN) 통합과 결합하는 방법을 설명합니다. 이 통합을 통해 다음을 수행할 수 있습니다.

- 맞춤형 광고를 통해 수익화된 선형 채널 생성
- 동일한 콘텐츠를 시청하는 다른 시청자에게 대상 광고를 제공합니다.
- 방송 품질 시청 경험 유지

SSAI를 채널 어셈블리와 결합할 때의 이점

SSAI를 채널 어셈블리와 통합하면 다음과 같은 몇 가지 주요 이점이 있습니다.

선형 채널의 수익화

맞춤형 광고를 선형 채널에 삽입하여 콘텐츠 라이브러리에서 수익을 창출합니다. 단일 선형 스트림 내에서 라이브 콘텐츠와 VOD 콘텐츠를 모두 수익화할 수 있습니다.

맞춤형 광고

동일한 채널 콘텐츠를 시청하는 다양한 시청자에게 다양한 광고를 제공합니다. 이 목표 접근 방식은 기존 방송 광고에 비해 광고 관련성과 잠재적 수익을 높입니다.

간소화된 광고 시간 관리

SCTE-35 마커로 콘텐츠를 조건화할 필요 없이 채널 어셈블리 프로그램에서 광고 중단점을 정의합니다. 이렇게 하면 콘텐츠의 자연스러운 중단점에 광고를 더 쉽게 삽입할 수 있습니다.

방송 품질 경험

MediaTailor는 콘텐츠와 광고 간의 원활한 전환을 통해 고품질 시청 환경을 유지합니다. 서버 측 광고 삽입은 다음과 같은 많은 일반적인 문제를 제거합니다.

- 광고 전환 중 버퍼링
- 수익화를 방지하는 광고 차단기
- 일관되지 않은 재생 품질

확장 가능한 전송

CDN과 결합하면 성능 또는 개인화 기능이 저하되지 않고 수백만 명의 동시 뷰어로 확장할 수 있습니다.

아키텍처 개요

SSAI를 채널 어셈블리와 결합하기 위한 아키텍처에는 일반적으로 다음과 같은 구성 요소가 포함됩니다.

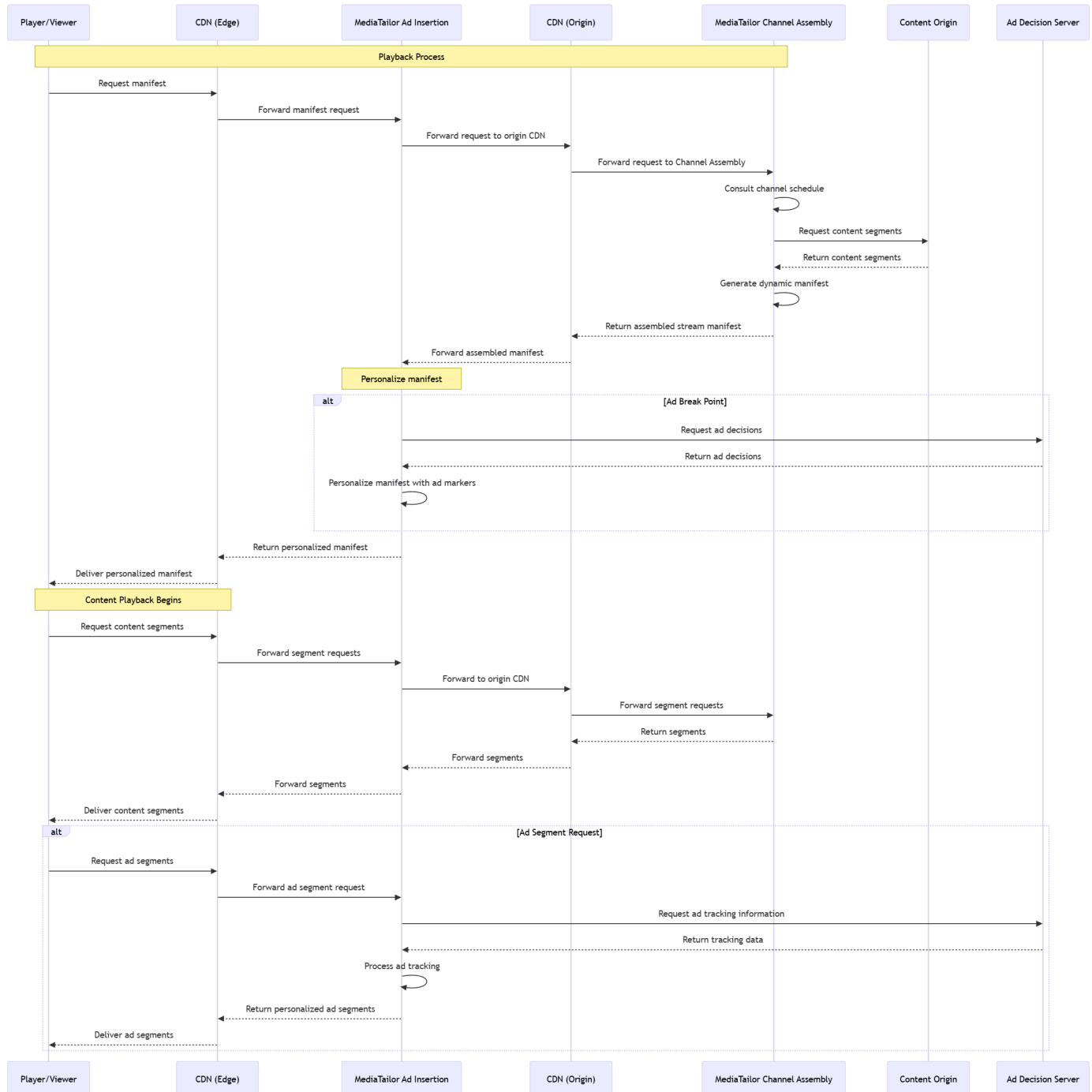
- 채널 어셈블리: VOD 및 라이브 콘텐츠에서 선형 채널을 생성하고 생성된 매니페스트에 광고 마커를 생성하는 슬레이트 콘텐츠를 삽입합니다.
- 광고 삽입: 광고 중단점을 인식하고 매니페스트에서 개인화된 광고 세그먼트를 가리키URLs를 삽입합니다.
- 광고 결정 서버(ADS): 각 뷰어에 삽입할 광고를 결정합니다.
- 콘텐츠 전송 네트워크(CDN): 수집된 콘텐츠와 광고 세그먼트를 최종 사용자에게 전달합니다.
- 오리진 서버: VOD 및 라이브 콘텐츠 세그먼트를 저장합니다.

이 아키텍처에서,

1. 채널 어셈블리는 VOD 및 라이브 콘텐츠에서 선형 채널을 생성하고 생성된 매니페스트에 광고 마커를 생성하는 슬레이트 콘텐츠를 삽입합니다.
2. 최종 사용자가 채널을 요청하면 광고 삽입은 선형 채널에 삽입된 광고 시간을 인식합니다.
3. 광고 삽입은 ADS를 호출하여 광고 목록을 수신하고, 트랜스코딩하고, 트랜스코딩된 광고 세그먼트를 가리키URLs를 개인화된 매니페스트에 삽입합니다.

4. CDN은 최종 사용자에게 개인화된 스트림을 전달합니다.

다음 다이어그램은 이 워크플로를 보여줍니다.



통합 설정

채널 어셈블리를 사용하여 SSAI를 설정하려면:

1. 최종 사용자의 매니페스트 요청을 수락하고 AWS Elemental MediaTailor 광고 삽입에 전달하도록 엣지 CDN을 구성합니다.
2. MediaTailor 광고 삽입을 설정하여 오리진 CDN에 요청을 전달합니다.
3. MediaTailor 채널 어셈블리에 요청을 전달하도록 오리진 CDN을 구성합니다.
4. 현재 일정에 따라 동적 매니페스트를 생성하도록 MediaTailor 채널 어셈블리를 설정합니다.
5. 구성된 매니페스트를 MediaTailor 광고 삽입에 전달하도록 오리진 CDN을 구성합니다.
6. MediaTailor 광고 삽입을 설정하여 광고 중단 시점에 광고 결정 서버에 광고 결정을 요청합니다.
7. 광고 마커(채널 어셈블리)를 대상 광고 세그먼트(ADS)를 가리키는 URLs로 대체하여 매니페스트를 개인화하도록 MediaTailor 광고 삽입을 구성합니다.
8. 최종 사용자에게 개인화된 매니페스트를 제공하도록 엣지 CDN을 설정합니다.
9. 콘텐츠 및 광고 세그먼트 요청을 효율적으로 처리하도록 CDN 아키텍처를 구성합니다.

채널 어셈블리의 광고 시간 정의

채널 어셈블리에서 프로그램을 생성할 때 다음과 같은 여러 가지 방법으로 광고 시간을 정의할 수 있습니다.

프로그램 전환

채널 일정의 프로그램 간에 광고를 삽입합니다. 이는 가장 간단한 접근 방식이며 광고가 프로그램 콘텐츠를 중단하지 않도록 합니다.

SCTE-35 마커

VOD 콘텐츠에 SCTE-35 마커가 포함된 경우 채널 어셈블리는 이러한 마커를 보존할 수 있으며 광고 삽입은 이를 광고 중단점으로 사용할 수 있습니다.

시간 기반 삽입

프로그램 내 특정 시점에 광고 시간을 정의합니다. 이렇게 하면 콘텐츠의 자연스러운 중단점에 광고를 삽입할 수 있습니다.

광고 시간을 사용하여 프로그램을 생성하는 방법에 대한 자세한 내용은 [프로그램 작업을](#) 참조하세요.

CDN 캐싱 고려 사항

채널 어셈블리와 SSAI를 CDN과 결합할 때 최적의 성능을 얻으려면:

- 채널 어셈블리와 SSAI 요청을 구분하는 캐시 동작 구성
- 의 권장 사항에 따라 매니페스트 및 세그먼트에 적절한 TTL 값을 설정합니다. [1단계: 최적의 광고 전송을 위해 CDN 캐싱 구성](#)
- 채널 어셈블리, 광고 삽입 및 CDN 오리진 간에 적절한 라우팅 보장
- 채널 어셈블리 및 광고 삽입 구성 요소 모두에 대한 성능 지표 모니터링

통합 구현을 위한 권장 캐싱 설정

콘텐츠 유형	TTL	캐시 키 요소
채널 어셈블리 매니페스트	0초	URL 경로 + 쿼리 파라미터
SSAI 개인화된 매니페스트	0초	URL 경로 + 모든 쿼리 파라미터
콘텐츠 세그먼트	24시간 이상	URL 경로만
광고 세그먼트	24시간 이상	URL 경로만

통합 솔루션 모니터링

통합 솔루션이 최적으로 작동하는지 확인하려면 다음 주요 지표를 모니터링하세요.

채널 어셈블리 지표

매니페스트 생성 시간, 프로그램 전환 및 채널 어셈블리 프로세스의 오류를 모니터링합니다.

광고 삽입 지표

광고 채우기 속도, 광고 결정 서버 응답 시간 및 광고 삽입 오류를 추적합니다.

CDN 지표

콘텐츠 및 광고 세그먼트 모두에 대한 캐시 적중률, 오리진 요청 볼륨 및 응답 지연 시간을 모니터링합니다.

최종 사용자 경험 지표

특히 광고 전환 중에 재버퍼링 이벤트, 시작 시간 및 최종 사용자 참여를 추적합니다.

모니터링에 대한 자세한 내용은 [CDN 및 MediaTailor 통합에 대한 작업 모니터링](#) 및 단원을 참조하십시오. [MediaTailor 채널 어셈블리 CDN 작업 모니터링](#).

일반적인 문제 해결

통합 솔루션 관련 문제를 해결할 때는 다음과 같은 일반적인 문제를 고려하십시오.

광고 중단 동기화 문제

광고가 예상 중단점에 나타나지 않는 경우 채널 어셈블리 프로그램의 광고 중단 정의가 올바르게 구성되었는지, 광고 삽입이 이러한 중단점을 올바르게 식별하는지 확인합니다.

매니페스트 전송 오류

최종 사용자에게 재생 문제가 발생하는 경우 CDN이 채널 어셈블리와 광고 삽입 간에 매니페스트 요청을 올바르게 전달하고 캐시 설정이 이러한 매니페스트의 동적 특성에 적합한지 확인합니다.

세그먼트 라우팅 문제

콘텐츠 또는 광고 세그먼트가 로드되지 않는 경우 CDN이 세그먼트 요청을 적절한 오리진으로 올바르게 라우팅하고 매니페스트의 세그먼트 URLs의 형식이 올바른지 확인합니다.

성능 저하

최종 사용자가 버퍼링 또는 지연 시간이 긴 경우 CDN 캐시 적중률 및 오리진 요청 볼륨을 확인하여 전송 파이프라인의 잠재적 병목 현상을 식별합니다.

문제 해결 지침은 [중단 없는 광고 전송을 위해 CDNs 사용한 MediaTailor SSAI 문제 해결](#) 섹션을 참조하세요.

모범 사례

SSAI를 채널 어셈블리와 성공적으로 통합하려면 다음 모범 사례를 따르세요.

- 철저한 테스트: 프로덕션에 배포하기 전에 다양한 콘텐츠 유형, 광고 시나리오 및 최종 사용자 조건으로 통합 솔루션을 테스트합니다.
- 지속적인 모니터링: 발생하는 문제를 신속하게 식별하고 해결할 수 있도록 포괄적인 모니터링 및 알림을 설정합니다.

- 캐싱 최적화: 실제 사용 패턴 및 성능 지표를 기반으로 CDN 캐싱 설정을 정기적으로 검토하고 조정합니다.
- 규모 조정 계획: 특히 인기 있는 채널 또는 이벤트의 경우 피크 트래픽 로드를 처리하도록 아키텍처를 설계합니다.
- 중복성 고려: 중요한 구성 요소에 중복성을 구현하여 선형 채널의 고가용성을 보장합니다.
- 광고 전환 최적화: 일관된 인코딩 프로파일과 세그먼트 기간을 사용하여 콘텐츠와 광고 간의 원활한 전환을 보장합니다.

관련 정보

SSAI를 채널 어셈블리와 통합하는 방법에 대한 자세한 내용은 다음을 참조하세요.

채널 어셈블리 설명서

[AWS Elemental MediaTailor 를 사용하여 선형 어셈블 스트림 생성](#) - 채널 어셈블리 개념에 대해 알아보기

[CDN을 사용한 채널 어셈블리](#) - CDN을 사용하여 채널 어셈블리 설정

SSAI 설명서

[CDN을 사용한 광고 삽입](#) - CDN을 사용하여 광고 삽입 설정

[CDN 및 MediaTailor 통합을 위한 광고 삽입 아키텍처 이해](#) - 광고 삽입 CDN 아키텍처 이해

CDN 구성

[CDN 통합 설정](#) - 일반 CDN 구성 지침

[CloudFront 통합](#) - CloudFront별 구성

CDN 및 MediaTailor 통합을 위한 성능 최적화

콘텐츠 전송 네트워크(CDN) 구성을 최적화하여 AWS Elemental MediaTailor 광고 삽입 구현의 성능을 극대화합니다. 이러한 설정은 효율적인 콘텐츠 전송과 최적의 시청자 경험을 보장합니다.

SSAI 구현과 관련된 자세한 캐싱 및 라우팅 최적화 지침은 섹션을 참조하세요 [CDN 성능 최적화](#). 통합 최적화 가이드는 모든 MediaTailor CDN 통합에 적용되는 포괄적인 캐싱 설정, 요청 라우팅 구성 및 성능 벤치마크를 제공합니다.

일반적인 성능 문제

CDNs을 사용한 SSAI 구현은 다음과 같은 몇 가지 성능 문제에 직면할 수 있습니다.

매니페스트 조작 오버헤드

MediaTailor는 실시간 매니페스트 조작을 수행하므로 제대로 최적화되지 않으면 지연 시간이 발생할 수 있습니다. 다음은 지연 시간을 초래할 수 있습니다.

- 광고 결정 서버(ADS) 요청 처리 시간
- 광고 세그먼트 참조를 사용하여 매니페스트를 수정하는 데 필요한 시간
- 개인화를 위한 추가 처리

캐시 효율성 문제

개인화된 매니페스트는 다음과 같은 이유로 CDN 캐시 효율성을 줄일 수 있습니다.

- 각 최종 사용자는 고유한 매니페스트를 수신할 수 있습니다.
- 세션 파라미터는 캐시를 조각화할 수 있습니다.
- 동적 콘텐츠에는 신중한 캐시 구성이 필요합니다.

오리진 로드 스파이크

캐싱이 잘못되면 다음 중에 오리진 로드가 급증할 수 있습니다.

- 트래픽이 많은 이벤트
- 캐시 새로 고침
- CDN 구성 변경 사항

광고 관련 재생 문제

광고 삽입으로 인해 다음과 같은 재생 중단이 발생할 수 있습니다.

- 광고 전환 중 버퍼링
- 콘텐츠와 광고의 품질 차이
- 광고를 검색할 수 없는 경우 재생 실패

캐싱 전략, 요청 라우팅, 성능 벤치마크 및 고급 최적화 기술을 포함한 포괄적인 성능 최적화 지침은 섹션을 참조하세요 [CDN 성능 최적화](#). 통합 최적화 가이드는 SSAI 구현을 포함한 모든 MediaTailor CDN 통합에 적용되는 자세한 설정 및 벤치마크를 제공합니다.

모범 사례 요약

CDNs.

아키텍처 모범 사례

- 규모 및 요구 사항에 적합한 아키텍처 패턴 선택
- 가까운 곳에 서비스를 배포하여 지연 시간 최소화
- 중요한 구성 요소에 대한 중복성 구현

캐싱 모범 사례

- 다양한 콘텐츠 유형에 대해 다양한 캐싱 전략 사용
- 캐시 키를 최적화하여 개인화와 효율성의 균형을 맞춥니다.
- 콘텐츠 유형 및 업데이트 빈도에 따라 적절한 TTLs 설정합니다. 자세한 TTL 권장 사항은 섹션을 참조하세요 [CDN 및 MediaTailor 통합을 위한 캐싱 최적화](#).

광고 전송 모범 사례

- 제한 시간 및 폴백을 사용하여 ADS 상호 작용 최적화
- 콘텐츠 사양과 일치하도록 광고 준비
- 효율적인 광고 세그먼트 전달 구현

모니터링 모범 사례

- SSAI 구현의 모든 구성 요소 모니터링
- 성능 저하에 대한 알림 설정
- 구성을 정기적으로 검토하고 최적화합니다.

전체 최적화 지침

자세한 캐싱 전략, 요청 라우팅 구성, 성능 벤치마크 및 고급 최적화 기술을 포함한 포괄적인 CDN 최적화 지침은 섹션을 참조하세요 [CDN 성능 최적화](#). 통합 최적화 가이드는 SSAI 구현을 포함한 모든 MediaTailor CDN 통합에 적용되는 전체 설정 및 벤치마크를 제공합니다.

CDN 및 MediaTailor 통합에 대한 작업 모니터링

AWS Elemental MediaTailor 는 콘텐츠 전송 네트워크(CDN) 지표와 결합할 때 SSAI 구현에 대한 포괄적인 인사이트를 제공하는 강력한 분석 기능을 제공합니다. 이 주제에서는 다음 내용을 다룹니다.

모든 MediaTailor 구현에 적용되는 필수 지표, 모니터링 도구 설정, 알림 구성 및 문제 해결 전략을 포함한 포괄적인 CDN 모니터링 지침은 섹션을 참조하세요 [CDN 모니터링](#). 이 주제에서는 SSAI별 모니터링 요구 사항 및 광고 삽입 분석에 중점을 둡니다.

- SSAI 및 CDN 통합을 위한 모니터링 전략
- 분석 도구 및 데이터 수집 방법
- 데이터 기반 최적화 기법

CDNs 있는 SSAI의 주요 지표

CDNs을 사용하여 SSAI 구현을 효과적으로 모니터링하려면 다음 필수 지표를 추적합니다.

광고 삽입 지표

광고 채우기 속도: 광고로 성공적으로 채워진 광고 기회의 백분율입니다.

광고 오류율: 오류가 발생한 광고 요청의 백분율입니다.

광고 응답 시간: 광고 결정 서버가 광고 요청에 응답하는 데 걸리는 시간입니다.

광고 기간 정확도: 삽입된 광고의 실제 기간이 예상 기간과 얼마나 일치하는지.

최종 사용자 경험 지표

재버퍼링 비율: 버퍼링에 소요된 보기 시간의 백분율입니다.

시작 시간: 비디오 재생을 시작하는 데 걸리는 시간입니다.

광고 전환 평활도: 플레이어가 콘텐츠와 광고 간에 얼마나 원활하게 전환되는지.

세션 기간: 스트림을 중단하기 전에 시청하는 시간입니다.

분석 도구 및 통합

이러한 도구를 결합하여 SSAI 구현을 위한 포괄적인 분석 솔루션을 생성합니다.

AWS Elemental MediaTailor 서버 측 지표

MediaTailor는 Amazon CloudWatch를 통해 광고 요청, 응답 및 오류를 추적하는 기본 제공 지표를 제공합니다. 이러한 지표는 CloudWatch 콘솔에서 보거나 사용자 지정 대시보드에 통합할 수 있습니다.

주요 MediaTailor 지표는 다음과 같습니다.

- `AdDecisionServer.Ads`: 광고 결정 서버에서 반환한 광고 수입입니다.
- `AdDecisionServer.Duration`: 광고 결정 서버에서 반환한 총 광고 기간입니다.
- `AdDecisionServer.Errors`: 광고 결정 서버에서 반환한 오류 수입입니다.
- `AdDecisionServer.Latency`: 광고 결정 서버의 응답 시간입니다.

MediaTailor 지표의 전체 목록은 [Amazon CloudWatch를 사용하여 MediaTailor 모니터링을 참조](#) 하세요.

CDN 분석

CDN 공급자는 콘텐츠 전송 성능에 대한 자세한 분석을 제공합니다. Amazon CloudFront의 경우 CloudWatch 지표 및 Amazon CloudFront 액세스 로그를 사용하여 전송 패턴을 분석합니다.

모니터링할 중요 CDN 지표:

- 콘텐츠 유형별 요청 수(매니페스트 대 세그먼트)
- 다양한 콘텐츠 유형에 대한 캐시 적중률
- 최종 사용자의 지리적 분포
- 오류 코드별 오류 발생률

클라이언트 측 추적

클라이언트 측 추적을 구현하여 서버 측에서 사용할 수 없는 최종 사용자 경험 지표를 수집합니다.

- 플레이어 이벤트(재생, 일시 중지, 탐색, 버퍼)
- 광고 보기 완료율
- 서비스 품질 지표(해상도 변경 사항, 비트 전송률)
- 최종 사용자 참여 패턴

MediaTailor 클라이언트 측 추적을 사용하여 이러한 지표를 수집하고 보고하는 것이 좋습니다.

통합 대시보드

여러 소스의 지표를 결합하는 포괄적인 대시보드를 생성합니다.

- CloudWatch 대시보드를 사용하여 MediaTailor 및 CloudFront 지표 결합
- 고급 시각화를 위해 타사 분석 플랫폼 고려
- 교차 서비스 상관관계를 설정하여 지표 간 관계 식별

모니터링 전략 구현

다음 단계에 따라 CDN 구현을 통해 SSAI에 대한 포괄적인 모니터링 전략을 구현합니다.

1. 기본 모니터링 설정

- MediaTailor에 대한 CloudWatch 지표 활성화
- CDN 로깅 및 지표 수집 구성
- 비디오 플레이어에서 클라이언트 측 추적 구현

2. 사용자 지정 대시보드 생성

- 주요 지표를 결합하는 CloudWatch 대시보드 구축
- 광고 채우기 속도, CDN 성능 및 최종 사용자 경험에 대한 시각화 포함
- 중요한 이벤트에 대한 주석 추가(구성 변경, 주요 브로드캐스트)

3. 알림 구성

- 중요 지표에 대한 CloudWatch 경보 설정
- 여러 관련 조건에서 트리거되는 복합 경보 생성
- 알림 채널 구성(이메일, SMS, Amazon SNS)

4. 자동 응답 구현

- CloudWatch Events를 사용하여 일반적인 문제에 대한 자동 응답 트리거
- 필요한 경우 수동 개입을 위한 실행서 생성
- 다양한 알림 시나리오에 대한 문제 해결 절차 문서화

Example포괄적인 SSAI 모니터링 대시보드 생성

이 예제에서는 MediaTailor와 CloudFront 지표를 결합하는 CloudWatch 대시보드를 생성하는 방법을 보여줍니다. CloudFront

```
{
  "widgets": [
    {
      "type": "metric",
      "properties": {
        "metrics": [
          [ "AWS/MediaTailor", "AdDecisionServer.Ads", "Configuration", "your-config-
name" ],
          [ ".", "AdDecisionServer.Errors", ".", "." ]
        ],

```

```

        "period": 300,
        "stat": "Sum",
        "region": "us-west-2",
        "title": "Ad Decision Server Performance"
    }
},
{
    "type": "metric",
    "properties": {
        "metrics": [
            [ "AWS/CloudFront", "Requests", "DistributionId", "your-distribution-id" ],
            [ ".", "4xxErrorRate", ".", "." ],
            [ ".", "5xxErrorRate", ".", "." ]
        ],
        "period": 300,
        "stat": "Average",
        "region": "us-east-1",
        "title": "CDN Performance"
    }
}
]
}

```

데이터 기반 최적화

수집한 분석 데이터를 사용하여 SSAI 구현을 최적화합니다.

CDN 캐시 최적화

캐시 적중률을 분석하여 개선 기회를 식별합니다.

- 콘텐츠 유형 및 업데이트 빈도에 따라 TTL 설정을 조정합니다. 자세한 TTL 권장 사항은 섹션을 참조하세요 [1단계: 최적의 광고 전송을 위해 CDN 캐싱 구성](#).
- 캐시 키 설정을 최적화하여 캐시 효율성 향상
- 다중 계층 캐싱을 위한 오리진 실드 구현 고려

광고 전송 최적화

광고 성능 지표를 사용하여 광고 전송을 개선합니다.

- 일반적인 광고 삽입 오류 식별 및 해결
- 광고 결정 서버 응답 시간 최적화
- 채우기 속도 분석을 기반으로 광고 대상 지정 파라미터 조정

최종 사용자 경험 최적화

클라이언트 측 지표를 기반으로 최종 사용자 경험을 개선합니다.

- 광고 시간 중 드롭오프 패턴 분석
- 원활한 재생을 위해 광고 전환 지점 최적화
- 최종 사용자 참여 데이터를 기반으로 광고 빈도 및 기간 조정

비용 최적화

성능 및 비용 고려 사항의 균형을 맞춥니다.

- 대역폭 사용 패턴을 분석하여 CDN 비용 최적화
- CloudFront 배포에 대한 가격 등급 조정 고려
- 다양한 캐싱 전략의 비용 이점 평가

모범 사례

효과적인 SSAI 모니터링 및 분석을 위해 다음 모범 사례를 따르세요.

- 기준 설정: 정상 작동 중에 지표를 수집하여 문제 해결 중에 비교에 사용할 수 있는 성능 기준을 설정합니다.
- 다중 수준 모니터링 구현: 아키텍처의 다양한 수준(오리진, CDN, 플레이어)에서 모니터링하여 성능을 전체적으로 파악할 수 있습니다.
- 서비스 간 지표 상관관계 파악: 문제의 근본 원인을 식별하기 위해 다양한 서비스의 지표 간 관계를 찾습니다.
- 이상 탐지 사용: CloudWatch 이상 탐지를 구현하여 지표에서 비정상적인 패턴을 자동으로 식별합니다.
- 정기 검토 및 개선: 모니터링 전략에 대한 정기 검토를 예약하고 변화하는 요구 사항과 새로운 인사이트에 따라 조정합니다.
- 조사 결과 및 작업 문서화: 최적화 작업과 그 결과에 대한 기록을 유지하여 제도적 지식을 구축합니다.

관련 정보

CDNs.

- [CDN 및 MediaTailor 통합을 위한 성능 최적화](#) 성능 최적화 기술을 위한

- [중단 없는 광고 전송을 위해 CDNs 사용한 MediaTailor SSAI 문제 해결](#) 일반적인 문제 해결을 위한
- [MediaTailor 지표에 대한 자세한 내용은 Amazon CloudWatch를 사용하여 MediaTailor 모니터링](#) MediaTailor
- [CloudFront 지표에 대한 자세한 내용은 CloudFront 및 엣지 함수 지표 보기](#) CloudFront

중단 없는 광고 전송을 위해 CDNs 사용한 MediaTailor SSAI 문제 해결

이 섹션에서는 콘텐츠 전송 네트워크(CDN)에서 AWS Elemental MediaTailor 동적 광고 삽입을 사용할 때 발생하는 일반적인 문제에 대한 솔루션을 제공합니다. 이러한 솔루션은 맞춤형 광고를 통해 비디오 수익화 문제를 해결하는 데 도움이 됩니다.

캐시 성능 문제, HTTP 오류 해결, 테스트 절차 및 모든 MediaTailor 구현에 적용되는 진단 기술을 포함한 포괄적인 CDN 문제 해결 지침은 섹션을 참조하세요 [CDN 통합 문제 해결](#). 이 섹션에서는 SSAI 관련 문제 해결 요구 사항 및 광고 삽입 문제에 중점을 둡니다.

CDN 및 SSAI 설정에서 문제가 발생하면 다음과 같은 일반적인 문제를 확인하세요.

스트림에 표시되지 않는 개인 맞춤형 광고

ADS가 올바르게 응답하고 AWS Elemental MediaTailor 통신하는지 확인합니다. 다음과 같은 잠재적 문제를 확인합니다.

- CDN을 통해 제대로 전달되지 않은 광고 대상 지정 쿼리 파라미터
- 콘텐츠에 잘못 정의된 광고 중단점
- ADS 연결 또는 응답 문제

광고 중단 시점의 재생 오류

콘텐츠 비트레이트 및 해상도에 맞게 광고 세그먼트가 올바르게 트랜스코딩되었는지 확인합니다. 다음과 같은 일반적인 문제를 확인합니다.

- 광고 세그먼트에 대한 요청을 잘못 라우팅하는 CDN
- 전환 시점의 매니페스트 조작 오류
- 콘텐츠와 광고 간에 일치하지 않는 인코딩 프로필

오래된 매니페스트

라이브 콘텐츠의 경우 CDN 캐시 TTL 설정이 적절한지 확인합니다. 개인화된 매니페스트의 경우 0초의 TTL을 사용합니다. 빠르게 변화하는 매니페스트에 캐시 무효화를 구현하는 것이 좋습니다. 포괄적인 TTL 지침은 섹션을 참조하세요 [CDN 및 MediaTailor 통합을 위한 캐싱 최적화](#).

높은 지연 시간

CDN 구성에서 최적의 라우팅을 확인합니다. 최상의 성능을 위해 CDN에 최종 사용자 근처에 엣지 로케이션이 있는지 확인합니다.

고급 문제 해결

더 복잡한 문제를 해결하려면 다음과 같은 고급 문제 해결 기법을 시도해 보세요.

일관성 없는 광고 타겟팅 동작

플레이어 요청과 ADS 요청 간에 쿼리 파라미터 불일치가 있는지 확인합니다. 필요한 모든 대상 지정 파라미터가 CDN을 통해 제대로 전달되었는지 확인합니다.

CDN 캐시 불일치

캐시 키 구성을 확인하여 콘텐츠가 적절하게 구분되도록 합니다. 중요한 매니페스트 업데이트를 위한 캐시 제거를 구현하는 것이 좋습니다.

광고 추적 실패

비컨 URLs 제대로 전달되고 CDN에 의해 차단되지 않았는지 확인합니다. 클라이언트 플레이어에서 추적 엔드포인트에 도달할 수 있는지 확인합니다.

성능 최적화

동적 광고 삽입 및 비디오 수익화 워크플로의 성능을 최적화하려면:

- 콘텐츠 유형 및 뷰어 패턴을 기반으로 TTL 설정을 미세 조정합니다. 자세한 TTL 권장 사항은 섹션을 참조하세요 [CDN 및 MediaTailor 통합을 위한 캐싱 최적화](#).
- 지리적 라우팅을 구현하여 글로벌 대상의 지연 시간 최소화
- 중복성과 최적의 성능을 위해 여러 CDN을 사용하는 것이 좋습니다.
- 캐시 적중률 모니터링 및 그에 따라 구성 조정
- 매니페스트 조작 프로세스를 최적화하여 광고 중단 시점의 처리 시간 단축
- 원활한 맞춤형 광고 삽입을 위해 공통 콘텐츠 프로필과 일치하는 사전 트랜스코딩 광고

성능 벤치마크

AWS Elemental MediaTailor 광고 삽입 CDN 통합을 최적화할 때는 다음 성능 벤치마크를 목표로 하세요.

캐시 적중률 대상

콘텐츠 세그먼트: 캐시 적중률 95% 초과

광고 세그먼트: 캐시 적중률 90% 초과

매니페스트: 해당 사항 없음(개인화된 광고 삽입을 위해 캐시해서는 안 됨)

지연 시간 벤치마크

매니페스트 요청 지연 시간: 100ms 미만(P95)

콘텐츠 세그먼트 전송: 50ms 미만(P95)

광고 세그먼트 전송: 75ms 미만(P95)

End-to-end 시작 시간: 2초 미만

오리진 로드 지표

최종 사용자당 오리진 요청: 최종 사용자당 분당 요청 0.1개 미만

최종 사용자당 오리진 대역폭: 총 최종 사용자 대역폭의 5% 미만

오류율 대상

매니페스트 오류: 0.1% 미만

세그먼트 오류: 0.01% 미만

플레이어 보고 재버퍼링: 1% 미만

확장성 벤치마크

성능 저하 없이 피크 이벤트 중 정상 트래픽의 10배 지원

채널당 초당 1,000개 이상의 요청을 처리할 수 있는 기능

Amazon CloudWatch 지표를 사용하여 이러한 성능 지표를 추적합니다. 자세한 모니터링 지침은 섹션을 참조하세요 [Amazon CloudWatch 지표 AWS Elemental MediaTailor 를 사용한 모니터링](#).

관련 정보

CDNs.

광고 삽입 설명서

[MediaTailor 광고 삽입 시작하기](#) - 광고 삽입 개념에 대해 알아보기

[설정](#) - 광고 삽입 시작하기

CDN 통합

[CDN 통합 설정](#) - 일반 CDN 구성 지침

[CloudFront 통합](#) - CloudFront별 구성

채널 어셈블리 통합

[CDN을 사용한 채널 어셈블리](#) - CDN을 사용한 채널 어셈블리에 대해 알아보기

[광고 삽입 구현](#) - 채널 어셈블리를 사용하여 광고 삽입 구현

모니터링 및 최적화

[CDN 및 MediaTailor 통합에 대한 작업 모니터링](#) - 포괄적인 모니터링 및 분석

[CDN 및 MediaTailor 통합을 위한 성능 최적화](#) - 성능 최적화 가이드

[Amazon CloudWatch 지표 AWS Elemental MediaTailor 를 사용한 모니터링](#) - MediaTailor에 대한 CloudWatch 지표

채널 어셈블리 및 CDN을 사용하여 MediaTailor 선형 채널 구축

이 섹션에서는 AWS Elemental MediaTailor 채널 어셈블리를 콘텐츠 전송 네트워크(CDN)와 통합하기 위한 포괄적인 지침을 제공합니다. 다음 단계에 따라 채널 어셈블리 CDN 통합을 설정, 구성 및 최적화합니다.

채널 어셈블리를 서버 측 광고 삽입(SSAI)과 결합하여 맞춤형 광고를 통해 수익화된 선형 채널을 생성할 수도 있습니다. 이 강력한 통합을 통해 동일한 채널 콘텐츠를 시청하는 다양한 시청자에게 대상 광고를 제공할 수 있으므로 브로드캐스트 품질 시청 환경을 유지하면서 수익 기회를 늘릴 수 있습니다. CDN을 [CDN을 사용한 광고 삽입](#).

이 주제에서는 매니페스트라는 용어를 사용하여 다변량 재생 목록, 미디어 재생 목록 및 MPDs.

MediaTailor 채널 어셈블리에 대한 자세한 내용은 섹션을 참조하세요 [AWS Elemental MediaTailor 를 사용하여 선형 어셈블 스트림 생성](#).

필요한 사항

CDN을 사용하여 MediaTailor 채널 어셈블리를 설정하기 전에 다음과 같은 필수 리소스를 수집합니다.

AWS 계정 및 권한

MediaTailor 리소스를 생성하고 관리할 수 있는 적절한 권한이 있는 AWS 계정

MediaTailor, CloudFront(사용하는 경우) 및 관련 서비스에 대한 IAM 권한

자세한 권한 요구 사항은 섹션을 참조하세요 [의 보안 AWS Elemental MediaTailor](#).

필수 서비스

실행 중인 MediaTailor 채널 어셈블리 채널(SSAI 구성뿐만 아니라)

콘텐츠 전송 네트워크(CDN) 계정(Amazon CloudFront 또는 타사 CDN)

VOD 콘텐츠의 오리진 스토리지(Amazon S3, MediaPackage 또는 기타 오리진 서버)

콘텐츠 요구 사항

VOD 소스는 HLS 또는 DASH 형식으로 적절하게 인코딩되고 패키징됩니다. 소스 위치 및 VOD 소스 작업에 대한 자세한 내용은 섹션을 참조하세요 [소스 위치 작업](#).

세그먼트 지속 시간이 일관된 콘텐츠(권장 최소: 1초)

광고 시간을 위한 광고 슬레이트 콘텐츠(광고 삽입을 구현하는 경우). 슬레이트 구성에 대한 자세한 내용은 섹션을 참조하세요 [MediaTailor 슬레이트 광고 삽입](#).

시작하기 전 준비 사항

Important

이 워크플로에는 실행 중인 MediaTailor 채널 어셈블리 채널이 필요합니다. MediaTailor SSAI 구성만 있으면이 통합에 충분하지 않습니다. CDN 통합을 진행하기 전에 활성 채널 어셈블리 채널이 구성되고 작동 중이어야 합니다.

CDN을 사용하여 MediaTailor 채널 어셈블리를 구현하기 전에 다음 설정 작업을 완료합니다.

1. CDN, MediaTailor 및 오리진 서버 간의 네트워크 연결 구성
2. 보안 콘텐츠 전송을 위한 HTTPS 설정
3. CDN 도메인에 대한 DNS 설정 구성

지식 사전 조건

이 솔루션을 성공적으로 구현하려면 다음이 필요합니다.

- 스트리밍 프로토콜 이해(HLS/DASH)
- CDN 구성 원칙에 대한 기본 지식
- MediaTailor 채널 어셈블리 개념에 대한 지식

기본 MediaTailor 설정은 [설정](#) 및 섹션을 참조하세요 [MediaTailor 채널 어셈블리 시작하기](#). 소스 위치 및 VOD 소스 작업에 대한 자세한 내용은 섹션을 참조하세요 [소스 위치 작업](#). 슬레이트 구성에 대한 자세한 내용은 섹션을 참조하세요 [MediaTailor 슬레이트 광고 삽입](#).

CDN 통합의 이점

채널 어셈블리를 CDN과 통합하면 이러한 주요 이점이 제공됩니다.

시청자 경험 개선

적절하게 구성된 CDN의 선형 채널의 버퍼링, 시작 시간 및 재생 오류를 줄입니다. 이로 인해 시청자의 참여와 만족도가 높아집니다.

비용 절감

효율적인 캐싱 전략은 오리진 요청을 최소화합니다. 이렇게 하면 특히 대용량 선형 채널에 중요한 데이터 전송 비용과 오리진 서버 부하가 줄어듭니다.

확장성

최적화된 CDN 구성은 성능 저하 없이 인기 있는 이벤트 중에 트래픽 급증을 처리하므로 피크 시청 시간 중에도 선형 채널을 계속 사용할 수 있습니다.

글로벌 도달 범위

적절하게 구성된 CDN의 위치에 관계없이 전 세계 시청자에게 짧은 지연 시간으로 콘텐츠를 제공하여 잠재적 시청자를 확장합니다.

원활한 프로그램 전환

최적화된 CDN 구성은 선형 채널의 프로그램 간에 원활한 전환을 보장하여 브로드캐스트 품질 시청 환경을 만듭니다.

주제

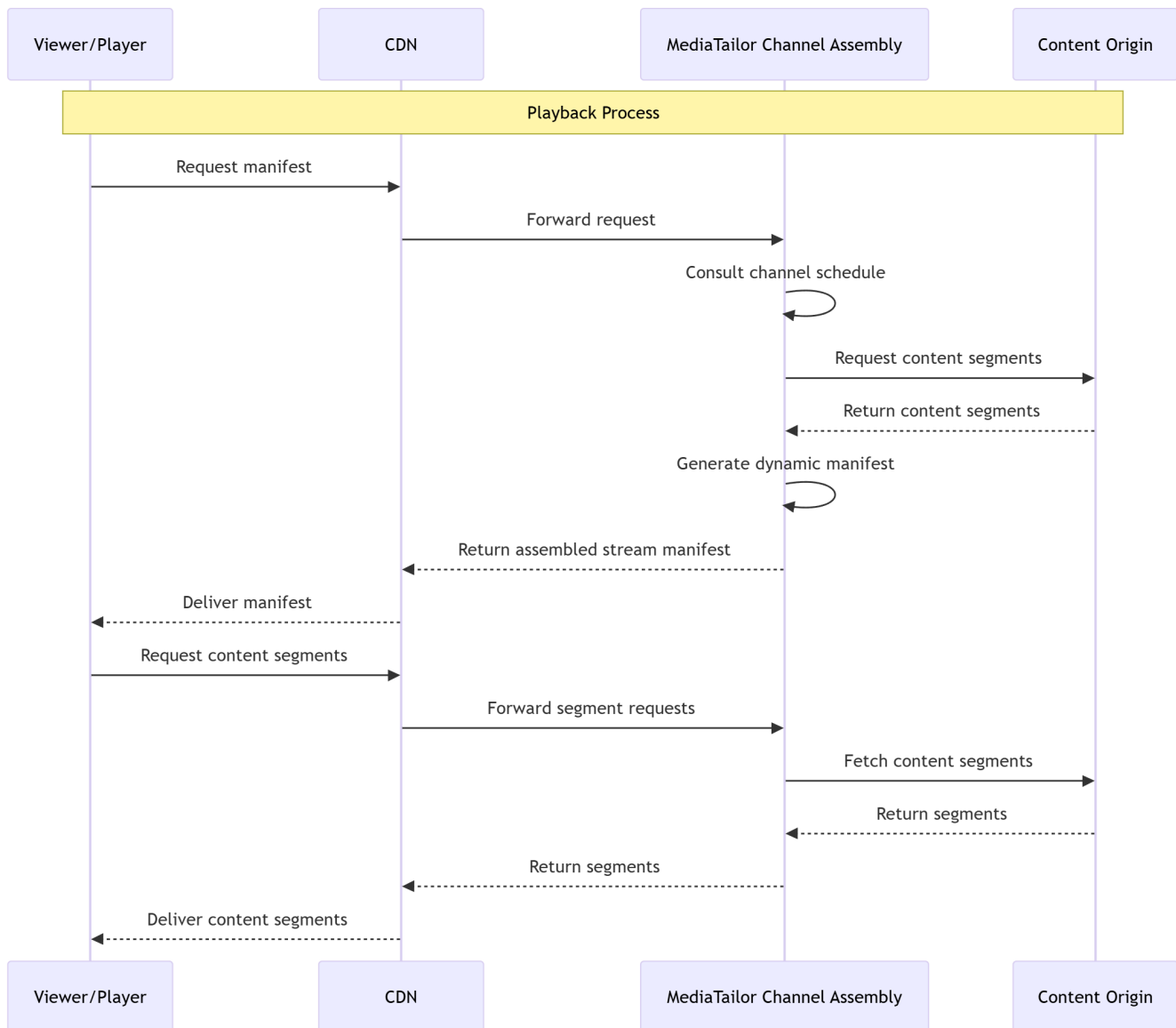
- [MediaTailor 채널 어셈블리 CDN 아키텍처 이해](#)
- [CDN을 사용하여 기본 MediaTailor 채널 어셈블리 설정](#)
- [MediaTailor 채널 어셈블리 CDNURLs 구성](#)
- [채널 어셈블리를 사용하여 MediaTailor 광고 삽입 구현](#)
- [MediaTailor 채널 어셈블리에 대한 시간 이동 보기 구성](#)
- [MediaTailor 채널 어셈블리 CDN 작업 모니터링](#)
- [전체 채널 어셈블리 CDN 최적화](#)

MediaTailor 채널 어셈블리 CDN 아키텍처 이해

AWS Elemental MediaTailor 채널 어셈블리는 콘텐츠 전송 네트워크(CDNs)와 통합되어 향상된 성능과 글로벌 도달 범위로 선형 스트리밍 채널을 제공합니다. 권장 아키텍처는 최종 사용자와 채널 어셈블리 사이에 CDN을 배치하고 채널 어셈블리는 오리진에서 직접 콘텐츠에 액세스합니다. 이 주제에서는 핵심 아키텍처 구성 요소와 이러한 구성 요소가 함께 작동하여 콘텐츠를 제공하는 방법을 설명합니다.

1. CDN의 최종 사용자 요청 매니페스트
2. CDN은 요청을 채널 어셈블리로 전달합니다.
3. 채널 어셈블리는 VOD 소스에서 매니페스트를 어셈블합니다.
4. 채널 어셈블리는 매니페스트를 CDN에 반환하여 최종 사용자에게 전달합니다.
5. 최종 사용자는 CDN을 통해 세그먼트를 요청합니다.
6. CDN은 세그먼트 요청을 적절한 오리진으로 라우팅합니다.

이 아키텍처는 CDN 사용의 보안 및 유연성 이점을 유지하면서 최적의 성능을 보장합니다.



채널 어셈블리에 대한 CDN 용어

이러한 주요 용어를 이해하면 채널 어셈블리 CDN 통합을 구현하고 문제를 해결하는 데 도움이 됩니다.

오리진 CDN 및 엣지 CDN

오리진 CDN: MediaTailor와 콘텐츠 오리진 사이에 위치한 CDN입니다. 오리진 서버의 부하를 줄이기 위해 콘텐츠 세그먼트를 캐싱합니다. 다중 CDN 아키텍처에서이 계층은 오리진과 직접 인터페이스하는 첫 번째 CDN 계층입니다.

엣지 CDN: 최종 사용자와 MediaTailor 사이에 위치한 CDN입니다. 시청자에게 개인화된 매니페스트와 콘텐츠를 제공합니다. 다중 CDN 아키텍처에서이 계층은 최종 사용자와 직접 인터페이스하는 가장 바깥쪽의 CDN 계층입니다.

CDN 구성 용어

캐시 동작: CDN이 캐싱 기간 및 오리진 라우팅을 포함하여 다양한 유형의 요청을 처리하는 방법을 결정하는 규칙입니다.

TTL(Time To Live): 오리진에서 새로 고쳐야 하기 전에 CDN 캐시에서 콘텐츠가 유효한 상태로 유지되는 기간입니다. 자세한 TTL 권장 사항은 섹션을 참조하세요 [CDN 및 MediaTailor 통합을 위한 캐싱 최적화](#).

캐시 키: CDN이 종종 URL 경로, 쿼리 파라미터 및 헤더를 포함하여 캐시된 콘텐츠를 저장하고 검색하는 데 사용하는 고유 식별자입니다.

오리진 실드: 오리진에 대한 요청 수를 줄이는 CDN 엣지 로케이션과 오리진 서버 간의 중간 캐싱 계층입니다.

요청 축소: 동일한 콘텐츠에 대한 여러 동시 요청을 단일 오리진 요청으로 결합하는 CDN 기능입니다.

MediaTailor별 CDN 용어

CDN 콘텐츠 세그먼트 접두사: MediaTailor가 매니페스트의 콘텐츠 세그먼트에 대한 URLs을 생성할 때 사용하는 CDN 도메인 이름입니다.

CDN 광고 세그먼트 접두사: MediaTailor가 매니페스트의 광고 세그먼트에 대한 URLs을 생성할 때 사용하는 CDN 도메인 이름입니다.

MediaTailor를 사용한 CDN 구성에 대한 자세한 내용은 섹션을 참조하세요 [CDN 통합 설정](#).

CDN을 사용하여 기본 MediaTailor 채널 어셈블리 설정

AWS Elemental MediaTailor 채널 어셈블리를 사용하면 콘텐츠 전송 네트워크(CDN)와의 기본 통합을 구성하여 최종 사용자에게 선형 스트리밍 채널을 효율적으로 제공할 수 있습니다. 다음 단계에 따라 채널 어셈블리와 CDN 간의 통합을 설정합니다.

1. 최종 사용자의 매니페스트를 수락하고 MediaTailor 채널 어셈블리로 전달하도록 CDN을 구성합니다.
2. 채널 일정에 액세스하고 현재 프로그래밍을 결정하도록 MediaTailor 채널 어셈블리를 설정합니다.

3. 일정에 따라 오리진 서버에서 콘텐츠 세그먼트를 요청하도록 MediaTailor 채널 어셈블리를 구성합니다.
4. 콘텐츠 오리진이 요청된 세그먼트를 MediaTailor 채널 어셈블리로 전송할 수 있는지 확인합니다.
5. 현재 일정에 따라 동적 매니페스트를 생성하도록 MediaTailor 채널 어셈블리를 설정합니다.
6. 어셈블된 다변량 재생 목록, 미디어 재생 목록 및 MPDs를 최종 사용자에게 전달하도록 CDN을 구성합니다.
7. 적절한 캐시 설정을 사용하여 최종 사용자의 세그먼트 요청을 처리하도록 CDN을 설정합니다.
8. 캐시 누락을 MediaTailor 채널 어셈블리로 전달하도록 CDN을 구성합니다.
9. 콘텐츠 오리진에서 요청된 세그먼트를 검색하도록 MediaTailor 채널 어셈블리를 설정합니다.
10. 재생을 위해 최종 사용자에게 콘텐츠 세그먼트를 전송하도록 CDN을 구성합니다.

MediaTailor 채널 어셈블리 CDNURLs 구성

AWS Elemental MediaTailor 채널 어셈블리를 사용하려면 콘텐츠 전송 네트워크(CDN)를 통한 콘텐츠 라우팅이 올바르게 작동하도록 적절한 기본 URL 구성이 필요합니다. 채널 어셈블리에서 기본 URL 설정을 구성하여 최종 사용자에게 성공적으로 콘텐츠를 전송할 수 있도록 합니다.

콘텐츠 세그먼트 URL 구성

채널 어셈블리 채널 구성에서 기본 URL을 CDN 도메인으로 설정합니다. 이렇게 하면 조립된 매니페스트의 모든 세그먼트 URLs이 오리진 서버에 직접 연결되지 않고 CDN을 가리킵니다.

예를 들어 오리진 콘텐츠가 `http://origin.example.com/content/` 있고 CDN 도메인이 인 경우 기본 URL을 `https://cdn.example.com/`로 설정합니다. `https://cdn.example.com/content/`.

액세스 제한 구성

보안을 강화하려면 오리진 서버에 대한 직접 액세스를 제한하도록 CDN을 구성합니다.

1. CDN에서 오리진 액세스 제어를 설정합니다.
2. CDN의 요청만 수락하도록 오리진 서버를 구성합니다.
3. 필요한 경우 최종 사용자 인증에 서명된 URLs 또는 쿠키를 사용합니다.

Amazon CloudFront의 경우 오리진 액세스 제어(OAC)를 사용하여 오리진에 대한 액세스를 보호할 수 있습니다. CloudFront 통합 보안에 대한 자세한 내용은 [섹션을 참조하세요](#) [CloudFront 통합](#).

채널 어셈블리를 사용하여 MediaTailor 광고 삽입 구현

의 채널 어셈블리는 서버 측 광고 삽입(SSAI) 및 콘텐츠 전송 네트워크(CDNs)와 원활하게 AWS Elemental MediaTailor 통합되어 맞춤형 광고를 통해 수익화된 선형 채널을 생성합니다.

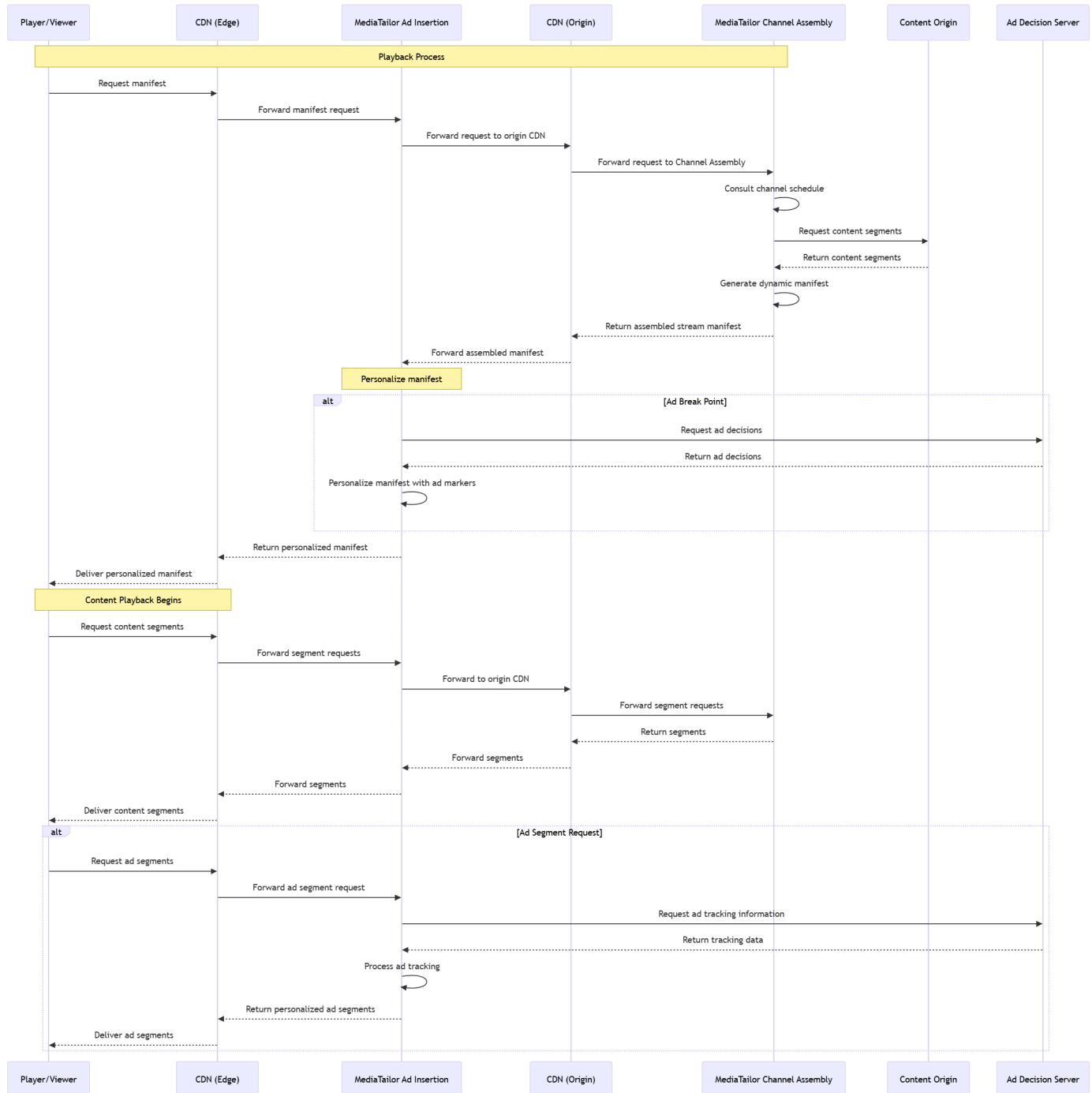
채널 어셈블리를 SSAI와 결합하면 방송 품질 경험을 유지하면서 시청자에게 맞춤형 광고를 제공하는 선형 채널을 구축할 수 있습니다. 이 통합 기능을 사용하여 다음을 수행할 수 있습니다.

- 콘텐츠 수익화 - 선형 채널의 대상 광고를 통해 수익 창출
- 경험 개인화 - 프로필을 기반으로 동일한 채널을 시청하는 시청자에게 다양한 광고 제공
- 품질 유지 - 방송 품질 보기를 위해 콘텐츠와 광고 간에 원활한 전환 보장
- 효율적인 확장 - CDN 전송을 통해 수백만 명의 동시 뷰어 지원

CDNs [CDN을 사용한 광고 삽입](#).

1. 최종 사용자의 매니페스트 요청을 수락하고 MediaTailor 광고 삽입에 전달하도록 엣지 CDN을 구성합니다.
2. MediaTailor 광고 삽입을 설정하여 오리진 CDN에 요청을 전달합니다.
3. MediaTailor 채널 어셈블리에 요청을 전달하도록 오리진 CDN을 구성합니다.
4. 현재 일정에 따라 동적 매니페스트를 생성하도록 MediaTailor 채널 어셈블리를 설정합니다.
5. 구성된 매니페스트를 MediaTailor 광고 삽입에 전달하도록 오리진 CDN을 구성합니다.
6. MediaTailor 광고 삽입을 설정하여 광고 중단 시점에 광고 결정 서버에 광고 결정을 요청합니다.
7. 광고 마커로 매니페스트를 개인화하도록 MediaTailor 광고 삽입을 구성합니다.
8. 최종 사용자에게 개인화된 매니페스트를 제공하도록 엣지 CDN을 설정합니다.
9. 콘텐츠 및 광고 세그먼트 요청을 효율적으로 처리하도록 CDN 아키텍처를 구성합니다.

다음 다이어그램은 이 결합된 워크플로를 보여줍니다.



채널 어셈블리와 SSAI를 결합할 때 최적의 성능을 얻으려면:

- 채널 어셈블리와 SSAI 요청을 구별하는 캐시 동작 구성

- 의 권장 사항에 따라 매니페스트 및 세그먼트에 적절한 TTL 값을 설정합니다. [CDN 및 MediaTailor 통합을 위한 캐싱 최적화](#)
- 채널 어셈블리, 광고 삽입 및 CDN 오리진 간에 적절한 라우팅 보장
- 채널 어셈블리 및 광고 삽입 구성 요소 모두에 대한 성능 지표 모니터링

CDNs.

- [CDN 및 MediaTailor 통합을 위한 광고 삽입 아키텍처 이해](#) - SSAI 아키텍처 및 개념에 대해 알아봅니다.
- [최적의 광고 전송을 위해 CDN을 사용하여 기본 MediaTailor SSAI 설정](#) - Step-by-step SSAI 구성 지침
- [중단 없는 광고 전송을 위해 CDN을 사용한 MediaTailor SSAI 문제 해결](#) - 일반적인 SSAI 통합 문제 해결

MediaTailor 채널 어셈블리에 대한 시간 이동 보기 구성

AWS Elemental MediaTailor 채널 어셈블리는 선형 채널의 일시 중지, 되감기 및 스타트오버와 같은 DVR과 유사한 기능을 활성화하는 시간 이동 보기 기능을 지원합니다. 시청자가 시청 환경을 제어할 수 있도록 시간 이동 시청을 지원하도록 콘텐츠 전송 네트워크(CDN)를 구성하여 이러한 기능을 활성화합니다.

시간 이동 보기 이해

시간 이동 보기는 다음을 포함한 선형 채널에 대해 DVR과 유사한 기능을 활성화합니다.

- 스타트오버: 시청자는 프로그램을 시작한 후 가입하더라도 처음부터 프로그램 시청을 시작할 수 있습니다.
- 일시 중지 및 재개: 최종 사용자는 콘텐츠를 일시 중지하고 나중에 시청을 재개할 수 있습니다.
- 되감기 및 빨리 감기: 최종 사용자는 사용 가능한 콘텐츠를 앞뒤로 탐색할 수 있습니다.
- 보기 지연: 최종 사용자는 채널 일정의 앞부분에서 방송된 콘텐츠를 볼 수 있습니다.

시간 이동 보기는 채널의 재생 URL에 `start` 파라미터를 추가하여 작동합니다. 파라미터는 현재 시간을 기준으로 오프셋을 초 단위로 지정합니다.

- 음수 값은 과거의 시간을 나타냅니다(예: `start=-3600`"1시간 전부터 시작").

- 양수 값은 미래의 시간을 나타냅니다(예: start=3600"향후 1시간부터 시작").

time-shift 파라미터가 있는 URL의 예:

```
https://example-cdn.com/out/v1/channel-name/index.m3u8?start=-3600
```

시간 지연 해결

CDN을 사용하여 시간 이동 보기를 지원하려면:

1. start 쿼리 파라미터를 채널 어셈블리로 전달하도록 CDN을 구성합니다.
2. 캐시 키에 start 파라미터를 포함하는 캐시 동작을 설정합니다.
3. 시간 전환 파라미터가 있는 매니페스트의 경우 짧은 TTL을 사용하거나 캐싱을 사용하지 않습니다.

이렇게 하면 각 최종 사용자가 요청된 시간 위치에 대해 올바른 매니페스트를 수신할 수 있습니다.

시간 이동을 위한 CDN 요구 사항

채널 어셈블리로 시간 이동 보기를 지원하려면 CDN이 다음 요구 사항을 충족해야 합니다.

- 모든 쿼리 파라미터를 채널 어셈블리에 전달합니다.
- 캐시 키에 start 파라미터를 포함합니다.
- 시간 이동 매니페스트에 대한 적절한 캐시 무효화를 지원합니다.
- 쿼리 파라미터를 기반으로 다양한 매니페스트 응답을 처리합니다.

MediaTailor 채널 어셈블리 CDN 작업 모니터링

AWS Elemental MediaTailor 채널 어셈블리는 안정적인 콘텐츠 전송을 보장하기 위해 콘텐츠 전송 네트워크(CDN)와 통합될 때 효과적인 모니터링이 필요합니다. 채널 어셈블리 및 CDN 통합에 대한 모니터링 전략을 구현하여 안정적인 콘텐츠 전송과 빠른 문제 해결을 보장합니다.

모든 MediaTailor 구현에 적용되는 필수 지표, 모니터링 도구 설정, 알림 구성 및 문제 해결 전략을 포함한 포괄적인 CDN 모니터링 지침은 섹션을 참조하세요 [CDN 모니터링](#). 이 섹션에서는 채널 어셈블리별 모니터링 요구 사항에 중점을 둡니다.

채널 어셈블리 및 CDN 통합에 대한 특정 모니터링을 구현합니다.

- 채널 어셈블리에서 매니페스트 생성 지표를 추적합니다.
- 시간 이동 보기 요청과 CDN 캐시 적중률에 미치는 영향을 모니터링합니다.
- 매니페스트 요청에서 비정상적인 패턴에 대한 알림을 구성합니다.
- 콘텐츠 소스에서 세그먼트 가용성에 대한 추적을 구현합니다.

Amazon CloudFront의 경우 CDN 지표를 MediaTailor 지표와 통합하여 [CloudWatch 대시보드를 사용하여 전체 전송 파이프라인을 시각화하는 대시보드를](#) 생성합니다.

채널 어셈블리와 함께 SSAI도 사용하는 경우 광고 삽입과 관련된 추가 모니터링 권장 사항은 [CDN 작업 모니터링](#) 섹션을 참조하세요.

전체 채널 어셈블리 CDN 최적화

캐싱 전략, 요청 라우팅 구성, 성능 벤치마크 및 고급 최적화 기술을 포함하여 채널 어셈블리 구현과 관련된 자세한 CDN 최적화 지침은 [섹션을 참조하세요](#) [CDN 성능 최적화](#).

통합 최적화 가이드는 다음을 제공합니다.

- 권장 TTL 값이 있는 채널 어셈블리별 캐싱 설정
- 매니페스트 및 세그먼트 전송을 위한 라우팅 최적화 요청
- 모든 MediaTailor 구현의 성능 벤치마크 및 대상
- Origin Shield 및 압축을 포함한 고급 최적화 기술
- SSAI를 사용한 채널 어셈블리에 대한 통합 워크플로 지침

MediaTailor를 MediaPackage 및 CDN과 통합

AWS Elemental MediaTailor 는와 통합되어 콘텐츠 전송 네트워크(CDN)를 통해 개인화된 비디오 광고를 AWS Elemental MediaPackage 제공합니다. MediaPackage는 인터넷을 통한 전송을 위해 비디오 콘텐츠를 준비하고 보호하는 just-in-time 비디오 패키징 및 제작 서비스입니다. 라이브 또는 온디맨드 비디오 콘텐츠를 가져와 HLS 및 DASH와 같은 스트리밍 형식으로 패키징하므로 다양한 디바이스에서 시청자가 사용할 수 있습니다.

MediaPackage를 MediaTailor 및 CDN과 결합하면 맞춤형 광고를 대규모로 제공하는 완전한 스트리밍 워크플로가 생성됩니다. CDN은 콘텐츠를 전 세계에 배포하여 지연 시간을 줄이고 시청자 경험을 개선하는 반면 MediaTailor는 대상 광고를 스트림에 삽입합니다.

이 주제에서는 MediaTailor, MediaPackage 및 CDN이 함께 작동하도록 하기 위한 필수 통합 단계에 중점을 둡니다. 고급 구성 옵션, 문제 해결 및 모니터링 지침은 섹션을 참조하세요 [다음 단계](#).

MediaPackage 및 CDN 워크플로 이해

통합을 구성하기 전에 MediaPackage, MediaTailor 및 CDN이 어떻게 함께 작동하는지 이해하는 것이 중요합니다.

1. 콘텐츠 준비: MediaPackage는 라이브 또는 온디맨드 비디오 콘텐츠를 수신하여 스트리밍 형식(HLS 또는 DASH 매니페스트 및 세그먼트)으로 패키징합니다.
2. 광고 삽입: MediaTailor는 MediaPackage에서 매니페스트를 요청하고, 개인화된 광고를 삽입하며, 수정된 매니페스트를 최종 사용자에게 제공합니다.
3. 글로벌 배포: CDN은 콘텐츠 세그먼트(MediaPackage)와 광고 세그먼트(MediaTailor)를 캐싱하고 전 세계 시청자에게 배포합니다.
4. 뷰어 재생: 비디오 플레이어 요청 매니페스트는 MediaTailor(매니페스트의 경우)와 MediaPackage(콘텐츠 세그먼트의 경우) 간에 요청을 적절하게 라우팅하는 CDN을 통해 매니페스트됩니다.

이 아키텍처는 다음과 같은 몇 가지 이점을 제공합니다.

- 확장성: CDN은 오리진 서버에 영향을 주지 않고 높은 최종 사용자 로드를 처리합니다.
- 성능: 시청자와 가장 가까운 엣지 로케이션에서 콘텐츠가 전달됩니다.
- 비용 효율성: 캐싱을 통한 대역폭 비용 절감
- 신뢰성: 여러 엣지 로케이션이 중복성을 제공합니다.

사전 조건

시작하기 전에 다음 구성 요소가 구성되어 있는지 확인합니다.

1. MediaPackage 엔드포인트: 비디오 콘텐츠를 수신하고 패키징하는 구성된 MediaPackage 엔드포인트입니다. 설정 지침은 [MediaPackage 사용 설명서의 MediaPackage 시작하기](#)를 참조하세요. MediaPackage
2. MediaTailor 구성: MediaPackage 엔드포인트를 콘텐츠 오리진으로 사용하는 MediaTailor 구성입니다. 설정 지침은 단원을 참조하십시오 [MediaTailor 광고 삽입을 위한 콘텐츠 소스 통합](#).
3. CDN 배포: 스트리밍 미디어와 함께 작동하도록 구성된 CDN 배포(예: CloudFront)입니다. 설정 지침은 CloudFront 개발자 안내서의 [배포 생성](#)을 참조하세요.

4. 광고 결정 서버: 광고 삽입을 위해 VAST 또는 VMAP 응답을 반환하는 구성된 광고 결정 서버입니다.

1단계: 필수 CDN 설정 구성

MediaPackage 통합을 성공적으로 수행하려면 적절한 CDN 구성이 중요합니다. 설정이 잘못되면 재생 실패, 캐시 성능 저하 및 비용 증가가 발생할 수 있습니다. 올바른 캐시 정책 및 쿼리 파라미터 전달이 없으면 CDN이 매니페스트를 올바르게 전달하지 못하거나 캐싱을 완전히 우회하여 오리진 서버 로드 가 높아지고 최종 사용자 경험이 저하될 수 있습니다.

기본 캐시 설정 구성

MediaPackage는 특정 캐시 제어 헤더를 사용하여 콘텐츠 전송을 최적화하므로 기본 캐싱을 구성하는 것이 중요합니다. 적절한 캐시 설정이 없으면 CDN이 이러한 헤더를 무시하여 불필요한 오리진 요청과 지연 시간 증가로 이어질 수 있습니다. 최적의 캐싱 동작을 보장하려면 다음 단계를 따르세요.

MediaPackage에서 작동하는 기본 캐싱을 구성하려면:

1. CloudFront 콘솔에서 CloudFront 배포 설정을 엽니다.
2. MediaPackage 오리진에 대한 캐시 정책을 선택하거나 생성합니다.
3. "원본 캐시 제어 헤더" 옵션을 활성화합니다.
4. MediaPackage가 캐시 제어 헤더를 통해 캐싱 동작을 제어할 수 있도록 허용합니다.

이 기본 구성을 통해 MediaPackage는 다양한 콘텐츠 유형에 적합한 캐시 기간을 자동으로 설정할 수 있습니다. 특정 TTL 값 및 성능 튜닝으로 고급 캐시 최적화를 구현하려면 먼저 이 기본 설정을 완료한 다음 로 계속 진행합니다 [MediaTailor 및 MediaPackage 콘텐츠 전송을 위한 CDN 캐싱 최적화](#).

필수 쿼리 파라미터 구성

쿼리 파라미터 구성은 MediaPackage 기능에 매우 중요합니다. CDN은 특정 쿼리 파라미터를 전달하여 시간 이동 보기 및 짧은 지연 시간 스트리밍과 같은 기능을 활성화해야 합니다. 쿼리 파라미터 설정이 잘못되면 이러한 기능이 작동하지 않을 수 있으며 캐시 효율성이 저하될까 걱정됩니다. 다음 단계에 따라 쿼리 파라미터 전달을 구성합니다.

CDN이 필요한 쿼리 파라미터를 MediaPackage에 전달하도록 하려면:

1. CloudFront 배포 설정에서 매니페스트 요청에 대한 캐시 정책을 선택하거나 생성합니다.

2. "캐시 키 설정"에서 "지정된 쿼리 문자열 포함"을 선택합니다.
3. 다음과 같은 필수 쿼리 파라미터를 추가합니다.
 - start 및 end - 시간 이동 보기 기능의 경우. 이러한 파라미터는 MediaPackage로 전달되어 스타트오버 및 캐치업 보기를 위한 특정 콘텐츠 기간을 정의합니다.
 - _HLS_msn 및 _HLS_part - LL-HLS 재생 요청 지원
 - m - 엔드포인트의 수정된 ti/compame을 캡처하는 데 사용됩니다. MediaPackage 응답에는 항상 엔드포인트의 수정된 시간을 캡처하는 ?m=### 태그가 포함됩니다. 콘텐츠가이 태그에 대해 다른 값으로 이미 캐시된 경우 CloudFront는 캐시된 버전을 제공하는 대신 새 매니페스트를 요청합니다.
 - aws.manifestfilter - 매니페스트 필터링 기능에 사용됩니다. 매니페스트 필터링을 사용하는 경우 매니페스트 필터링 기능이 작동하는 데 필요한 MediaPackage 오리진으로 aws.manifestfilter 쿼리 문자열을 전달하도록 배포를 구성하려면이 파라미터를 포함해야 합니다.
4. MediaPackage에서 사용하는 쿼리 문자열만 포함합니다. 불필요한 쿼리 문자열을 포함하면 동일한 콘텐츠에 대해 여러 캐시 변형을 생성하여 캐시 효율성이 떨어집니다.

이러한 파라미터를 사용하면 CDN에서 기본 MediaPackage 기능을 사용할 수 있습니다. 다양한 구독 계층 또는 디바이스 유형에 콘텐츠 필터링을 구현해야 하는 경우 먼저이 기본 쿼리 파라미터 설정을 완료한 다음 로 진행합니다 [MediaTailor, MediaPackage 및 CDN을 사용하여 매니페스트 필터링 설정](#).

MediaTailor가 시간 이동 보기를 위해 start 및 end 통해 MediaPackage에 쿼리 파라미터를 전달하는 방법에 대한 자세한 내용은 [오리진에 대한 MediaTailor 쿼리 파라미터 처리](#)의 섹션을 참조하세요 [MediaTailor 매니페스트 쿼리 파라미터](#).

LL-HLS에 대한 응답 제한 시간 구성

LL-HLS는 CDN이 새 콘텐츠 세그먼트를 기다리는 "차단 요청" 메커니즘을 사용하기 때문에 지연 시간이 짧은 HLS의 경우 제한 시간 구성이 중요합니다. 제한 시간이 너무 짧으면 MediaPackage가 새 세그먼트로 응답하기 전에 요청이 실패하여 재생이 중단되고 뷰어 경험이 저하됩니다. 원활한 LL-HLS 재생을 위해 적절한 제한 시간을 구성합니다.

지연 시간이 짧은 HLS를 사용하는 경우 CDN 제한 시간 설정을 구성합니다.

1. CDN 설정에서 오리진 제한 시간 구성을 찾습니다.
2. 응답 제한 시간 값을 파트 지속 시간의 3배 이상으로 설정합니다.
3. 예를 들어 파트 지속 시간이 0.3초인 경우 제한 시간을 0.9초 이상으로 설정합니다.

이렇게 하면 CDN이 요청 차단 메커니즘을 사용할 때 MediaPackage가 응답할 때까지 충분히 오래 기다릴 수 있습니다.

2단계: 통합 확인

통합 테스트는 최종 사용자에게 문제가 발생하기 전에 모든 구성 요소가 올바르게 함께 작동하는지 확인하는 데 필수적입니다. 통합에 실패하면 재생이 끊어지거나 광고가 누락되거나 성능이 저하될 수 있습니다. 이 확인 프로세스는 제어된 환경에서 문제를 식별하고 해결하는 데 도움이 됩니다.

CDN 설정을 구성한 후 콘텐츠 요청부터 광고 삽입까지 전체 워크플로를 테스트하여 통합이 올바르게 작동하는지 확인합니다.

2.1단계: 기본 재생 테스트

기본 재생 테스트는 CDN이 매니페스트 요청을 올바르게 처리하고 MediaTailor에 전달하는지 확인합니다. 이 테스트는 캐시 정책, 쿼리 파라미터 전달 및 매니페스트 처리와 관련된 구성 문제를 식별하는 데 도움이 됩니다. 다음 단계에 따라 기본 매니페스트 전송을 테스트합니다.

CDN을 통해 매니페스트를 요청하여 기본 통합이 작동하는지 테스트합니다.

1. 웹 브라우저 또는 curl을 사용하여 CDN을 통해 매니페스트 URL을 요청합니다.
2. 매니페스트가 성공적으로 로드되고 콘텐츠와 광고 세그먼트가 모두 포함되어 있는지 확인합니다.
3. 매니페스트의 콘텐츠 세그먼트 URLs CDN 도메인을 가리키는지 확인합니다.
4. 광고 세그먼트 URLs CDN 도메인도 가리키는지 확인합니다.

매니페스트가 올바르게 로드되고 예상 URLs이 포함된 경우 기본 통합이 작동하는 것입니다. 포괄적인 테스트 방법 및 고급 검증 절차는 섹션을 참조하세요 [CDN 및 MediaTailor 통합 테스트 및 검증](#). 통합의 성능 및 상태에 대한 포괄적인 모니터링을 설정하려면 섹션을 참조하세요 [MediaPackage, CDN 및 MediaTailor 통합의 성능 모니터링](#).

2.2단계: 비디오 재생 테스트

비디오 재생 테스트를 통해 CDN을 통한 광고 삽입 및 콘텐츠 전송을 포함하여 완전한 통합이 end-to-end로 작동하도록 할 수 있습니다. 이 테스트는 콘텐츠 세그먼트와 광고 세그먼트가 모두 올바르게 캐싱 및 전달되고 최종 사용자 경험이 품질 표준을 충족하는지 확인합니다. 다음 단계에 따라 전체 재생 기능을 테스트합니다.

삽입된 광고에서 비디오 재생이 올바르게 작동하는지 테스트합니다.

1. 비디오 플레이어(예: Video.js 또는 HLS.js)를 사용하여 CDN을 통해 콘텐츠를 재생합니다.
2. 버퍼링 문제 없이 비디오가 원활하게 재생되는지 확인합니다.
3. 재생 중에 광고가 예상 시간에 삽입되었는지 확인합니다.
4. 콘텐츠와 광고 세그먼트가 모두 CDN(오리진에서 직접 로드되지 않음)에서 로드되는지 확인합니다.

광고에서 재생이 원활하게 작동하면 통합이 올바르게 작동하는 것입니다. 포괄적인 테스트 방법 및 고급 검증 절차는 섹션을 참조하세요 [CDN 및 MediaTailor 통합 테스트 및 검증](#). 재생 문제, 버퍼링 또는 광고 삽입 문제가 발생하는 경우 섹션을 참조하세요 [MediaPackage, CDN 및 MediaTailor 통합 문제 해결](#).

다음 단계

기본 통합을 완료한 후 고급 기능 및 최적화를 구현할 수 있습니다.

고급 CDN 최적화

자세한 캐시 최적화, TTL 구성 및 성능 튜닝은 섹션을 참조하세요 [MediaTailor 및 MediaPackage 콘텐츠 전송을 위한 CDN 캐싱 최적화](#).

매니페스트 필터링

계층화된 서비스, 디바이스 최적화 또는 액세스 제어를 위한 콘텐츠 필터링을 구현하려면 섹션을 참조하세요 [MediaTailor, MediaPackage 및 CDN을 사용하여 매니페스트 필터링 설정](#).

문제 해결

통합에 문제가 발생하면 섹션을 참조하세요 [MediaPackage, CDN 및 MediaTailor 통합 문제 해결](#).

성능 모니터링

포괄적인 모니터링을 설정하고 주요 성능 지표를 이해하려면 섹션을 참조하세요 [MediaPackage, CDN 및 MediaTailor 통합의 성능 모니터링](#).

MediaTailor, MediaPackage 및 CDN을 사용하여 매니페스트 필터링 설정

AWS Elemental MediaTailor 는와 함께 매니페스트 필터링 AWS Elemental MediaPackage 을 사용하여 콘텐츠 전송 네트워크(CDN)를 통해 여러 최종 사용자에게 전달되는 매니페스트에 포함되는 오디오 및 비디오 스트림을 사용자 지정합니다. 이는 계층형 서비스 상품, 디바이스별 최적화 또는 콘텐츠 액세스 제어를 구현하는 데 특히 유용합니다.

이 주제에서는 매니페스트 필터링 기능 구현에 특히 중점을 둡니다. 매니페스트 필터링을 구현하기 전에 기본 콘텐츠 전송 네트워크 통합 설정을 완료해야 합니다. 기본 MediaPackage 및 콘텐츠 전송 네트워크 통합을 아직 설정하지 않은 경우 [로 시작합니다](#) [MediaTailor를 MediaPackage 및 CDN과 통합](#).

매니페스트 필터링 기능

매니페스트 필터링을 구현하기 전에 이 기능으로 수행할 수 있는 작업을 이해합니다.

코어 필터링 기능

매니페스트 필터링은 콘텐츠 전송을 제어하는 데 도움이 되는 몇 가지 주요 기능을 제공합니다.

- 프리미엄 콘텐츠(예: 4K HEVC)에 대한 최종 사용자 액세스 제한
- 적절한 스트림을 사용하여 특정 디바이스 유형을 대상으로 지정
- 오디오 샘플 속도, 언어 또는 비디오 코덱을 기반으로 콘텐츠 필터링
- 다양한 구독자에게 다양한 품질 티어 제공

일반 사용 사례

이러한 사용 사례는 매니페스트 필터링이 특정 비즈니스 요구 사항을 해결하는 방법을 보여줍니다.

구독 티어

기본 구독자에게 해상도가 낮은 스트림을 제공하는 동시에 프리미엄 구독자에게 4K 콘텐츠에 대한 액세스 권한 제공

예: 기본 티어는 720p로 제한, 프리미엄 티어는 최대 4K까지 획득

디바이스 최적화

디바이스 기능에 따라 적절한 스트림을 자동으로 제공

예: 모바일 디바이스의 비트 전송률이 낮고 스마트 TVs 품질이 향상됩니다.

대역폭 관리

사용량이 가장 많은 기간 동안 스트림 품질을 제한하여 네트워크 비용 관리

예: 트래픽이 많은 이벤트 발생 시 최대 비트 전송률 감소

리전별 콘텐츠

최종 사용자 위치에 따라 다양한 오디오 언어 또는 콘텐츠 변형 제공

예: 로컬 언어 오디오 트랙 자동 필터링

매니페스트 필터링 개념에 대한 자세한 내용은 AWS Elemental MediaPackage 사용 설명서의 [매니페스트 필터링](#)을 참조하세요.

매니페스트 필터링을 위한 CDN 구성

CDN이 필터링이 작동하려면 MediaPackage에 `aws.manifestfilter` 쿼리 파라미터를 전달해야 하므로 매니페스트 필터링을 위한 CDN 구성이 필수적입니다. 적절한 쿼리 문자열 전달이 없으면 CDN에서 필터 파라미터를 제거하고 모든 뷰어는 구독 계층 또는 디바이스 기능에 관계없이 필터링되지 않은 매니페스트를 수신합니다. 이 구성을 사용하면 필터링 로직이 의도한 대로 MediaPackage 및 함수에 도달합니다.

CDN을 통해 매니페스트 필터링을 활성화하려면 쿼리 문자열 전달을 구성해야 합니다.

1. CloudFront 배포에서 매니페스트 요청에 대한 캐시 동작을 생성하거나 편집합니다.
2. 캐시 정책의 경우 새 정책을 생성하거나 기존 정책을 편집합니다.
3. 캐시 키 설정에서 "지정된 쿼리 문자열 포함"을 선택합니다.
4. `aws.manifestfilter`를 허용된 쿼리 문자열 목록에 추가합니다.
5. 다른 MediaPackage 기능도 사용하는 경우 쿼리 파라미터를 추가합니다.
 - `start` 및 `end` - 시간 이동 보기
 - `time_delay` - 시간 지연 기능의 경우
 - `_HLS_msn` 및 `_HLS_part` - LL-HLS의 경우

배포 생성에 대한 자세한 내용은 Amazon CloudFront 개발자 안내서의 [배포 생성](#)을 참조하세요.

클라이언트 측 필터링 구현

클라이언트 측 구현에서는 비디오 플레이어와 애플리케이션이 필터링된 콘텐츠를 요청하는 방법을 정의합니다. 이 구성은 각 최종 사용자가 구독 수준, 디바이스 기능 또는 기타 기준에 따라 수신하는 콘텐츠를 결정합니다. 적절한 구현을 통해 최종 사용자는 최적의 CDN 캐시 효율성을 유지하면서 액세스해야 하는 콘텐츠만 수신할 수 있습니다.

비디오 플레이어 및 애플리케이션에서 매니페스트 필터링을 구현하려면:

필터링 작동 방식

필터링 프로세스는 다음과 같이 작동합니다.

1. 비디오 플레이어 또는 애플리케이션이 필터 파라미터가 포함된 매니페스트 URL을 요청합니다.
2. CDN은 요청(쿼리 파라미터 포함)을 MediaTailor에 전달합니다.

3. MediaTailor는 오리진 매니페스트를 요청할 때 필터 파라미터를 MediaPackage에 전달합니다.
4. MediaPackage는 필터를 적용하고 기준과 일치하는 변형만 포함하는 사용자 지정 매니페스트를 반환합니다.
5. MediaTailor는 광고 삽입을 위해 필터링된 매니페스트를 처리하고 플레이어에게 반환합니다.

필터링을 위한 URL 형식

올바른 URL 형식을 이해하는 것은 성공적인 필터링 구현에 매우 중요합니다. URL 형식이 잘못되면 필터링 파라미터가 무시되거나 HTTP 오류가 발생합니다. URL 구조에는 CDN이 MediaPackage에 전달하는 쿼리 문자열로 필터 파라미터가 포함되어야 합니다. 다음 단계에 따라 적절한 URL 형식을 구현합니다.

비디오 플레이어에서 매니페스트 필터링을 구현하려면:

1. 적절한 필터 파라미터를 포함하도록 플레이어의 매니페스트 요청 URLs을 수정합니다.
2. 쿼리 파라미터와 함께 다음 URL 형식을 사용합니다.

```
https://CloudFront-Domain/v1/master/MediaTailor-Config/index.m3u8?
aws.manifestfilter=video_codec:h264;audio_language:en-US
```

3. 플레이어가 이 URL을 요청하면 MediaTailor는 이러한 파라미터를 MediaPackage에 전달하여 필터링된 매니페스트를 생성합니다.

일반적인 필터링 시나리오

다음 예제를 사용하여 일반적인 필터링 시나리오를 구현합니다.

디바이스별 콘텐츠 전송

디바이스 기능을 기준으로 필터링하려면 매니페스트 요청에 이 파라미터를 추가합니다.

```
aws.manifestfilter=video_codec:h264;audio_sample_rate:0-44100
```

이 예제에서는 모바일 디바이스에 적합한 최대 44.1kHz의 샘플 속도로 H.264 비디오 및 오디오로 콘텐츠를 제한합니다.

프리미엄 콘텐츠 제한

비트 전송률이 높은 콘텐츠에 대한 액세스를 제한하려면 매니페스트 요청에 이 파라미터를 추가합니다.


```
aws.manifestfilter=video_bitrate:0-9000000
```

이 예제에서는 기본 구독 계층에 적합한 비디오 비트레이트를 9Mbps 이하로 제한합니다.

언어 선택

특정 오디오 언어를 필터링하려면 매니페스트 요청에이 파라미터를 추가합니다.

```
aws.manifestfilter=audio_language:fr,en-US,de
```

이 예제에는 프랑스어, 미국 영어 및 독일어 오디오 트랙만 포함됩니다.

해결 대상 지정

특정 비디오 해상도를 필터링하려면 매니페스트 요청에이 파라미터를 추가합니다.

```
aws.manifestfilter=video_height:240-360,720-1080
```

이 예제에는 중간 해상도를 제외하고 높이가 240~360픽셀~720-1080픽셀인 비디오 스트림이 포함되어 있습니다.

코덱 기반 필터링

특정 비디오 코덱을 필터링하려면 매니페스트 요청에이 파라미터를 추가합니다.

```
aws.manifestfilter=video_codec:h264,h265
```

이 예제에는 다른 코덱을 제외한 H.264 및 H.265 비디오 스트림만 포함됩니다.

특별 고려 사항 및 제한 사항

매니페스트 필터링을 구현할 때 발생하는 일반적인 문제를 방지하려면:

기술적 제한 사항

- TS 매니페스트의 경우 오디오 변환 그룹을 사용하여 필터링된 오디오 스트림과 멀티플렉싱된 비디오 스트림을 제거하지 않도록 합니다.
- TS 및 CMAF 매니페스트에서는 확인을 위해 오디오 샘플 속도와 비디오 비트레이트를 매니페스트에 쉽게 볼 수 없습니다.

- 미디어 재생 목록 또는 세그먼트에 요청 파라미터를 추가하면 HTTP 400 오류가 발생합니다.

오류 조건

- 필터링으로 인해 빈 매니페스트가 발생하는 경우(필터 기준을 충족하는 스트림 없음) MediaPackage는 HTTP 400 오류를 반환합니다.
- 필터 구성(엔드포인트 필터 + 쿼리 파라미터)이 충돌하면 HTTP 404 오류가 발생합니다.
- 잘못된 필터 구문 또는 지원되지 않는 필터 유형으로 인해 HTTP 400 오류가 발생합니다.

성능 고려 사항

- 각 고유 필터 조합은 별도의 캐시 항목을 생성하여 캐시 효율성을 잠재적으로 줄일 수 있습니다.
- 기준이 많은 복잡한 필터는 매니페스트 생성 성능에 영향을 미칠 수 있습니다.
- 정적 필터링 시나리오에 엔드포인트 수준 필터를 사용하여 캐시 성능을 개선하는 것이 좋습니다.

필터링 구현 테스트

매니페스트 필터링 구현을 테스트하는 것은 최종 사용자가 액세스 수준 및 디바이스 기능에 따라 올바른 콘텐츠를 수신하도록 하는 데 매우 중요합니다. 필터링이 실패하면 최종 사용자가 잘못된 품질 수준, 지원되지 않는 형식 또는 액세스할 수 없는 콘텐츠를 수신할 수 있습니다. 포괄적인 테스트를 통해 이러한 문제가 최종 사용자에게 영향을 미치기 전에 식별하고 해결할 수 있습니다.

매니페스트 필터링이 올바르게 작동하는지 확인하려면:

1. 필터 파라미터가 다른 매니페스트를 요청하고 결과를 확인합니다.
2. 필터링된 매니페스트에 예상 스트림만 포함되어 있는지 확인
3. 엣지 케이스(빈 결과, 잘못된 필터)를 테스트하여 적절한 오류 처리를 보장합니다.
4. CDN이 필터 파라미터를 올바르게 전달하고 있는지 확인
5. 서로 다른 디바이스 및 플레이어로 테스트하여 호환성 보장

필터링 문제 해결은 MediaPackage CDN 통합 문제 해결을 참조하세요.

HTTP 400 오류, 빈 매니페스트 또는 예상대로 작동하지 않는 필터링 파라미터가 발생하는 경우 특정 매니페스트 필터링 문제 해결 지침은 [MediaPackage, CDN 및 MediaTailor 통합 문제 해결](#) 섹션을 참조하세요.

MediaTailor 및 MediaPackage 콘텐츠 전송을 위한 CDN 캐싱 최적화

AWS Elemental MediaTailor 를 콘텐츠 오리진 AWS Elemental MediaPackage 으로 사용할 때 최적의 성능을 얻으려면 적절한 콘텐츠 전송 네트워크(CDN) 캐싱 구성이 필요합니다. MediaPackage는 콘텐츠 전송 네트워크에 다양한 유형의 콘텐츠를 캐싱할 기간을 알려주는 특정 캐시 제어 헤더를 제공합니다. 이러한 권장 사항을 따르면 원활한 재생과 효율적인 콘텐츠 전송이 가능합니다.

이 주제에서는 특히 캐싱 동작을 최적화하여 성능을 극대화하고 비용을 최소화하는 데 중점을 둡니다. 고급 캐싱 최적화를 구현하기 전에 기본 콘텐츠 전송 네트워크 통합 설정을 완료했는지 확인합니다. 아직 기본 통합을 설정하지 않은 경우 [로 시작합니다](#) [MediaTailor를 MediaPackage 및 CDN과 통합](#).

MediaPackage 캐시 제어 헤더

MediaPackage는 캐싱 동작을 최적화하기 위해 다양한 콘텐츠 유형에 대해 특정 TTL 값을 설정합니다.

다변량 재생 목록(HLS 및 LL-HLS)

TTL: 미디어 세그먼트 지속 시간의 절반

이유: 이러한 재생 목록은 새 세그먼트를 사용할 수 있게 되면 변경되므로 자주 업데이트해야 합니다.

미디어 재생 목록(일반 HLS)

TTL: 미디어 세그먼트 지속 시간의 절반

이유: 다변량 재생 목록과 마찬가지로 콘텐츠가 진행됨에 따라 이러한 업데이트

미디어 재생 목록(LL-HLS)

TTL: 1초

이유: 지연 시간이 짧은 스트리밍에는 매우 빈번한 업데이트가 필요합니다.

TS 미디어 세그먼트 및 초기화 세그먼트

TTL: 1209600초(14일)

이유: 미디어 세그먼트는 일단 생성되면 변경되지 않으므로 장기간 캐시할 수 있습니다.

CMAF 미디어 세그먼트 및 초기화 세그먼트

TTL: 1209600초(14일)

이유: TS 세그먼트와 마찬가지로 생성된 후에는 변경할 수 없습니다.

모든 MediaTailor 워크플로에 대한 포괄적인 TTL 권장 사항 및 추가 캐싱 최적화 전략은 섹션을 참조하세요 [CDN 및 MediaTailor 통합을 위한 캐싱 최적화](#).

CDN 캐시 정책 구성

최적의 성능과 비용 효율성을 위해서는 적절한 캐시 정책 구성이 필수적입니다. 콘텐츠 유형(매니페스트, 세그먼트, 초기화 파일)마다 캐싱 요구 사항이 다릅니다. 별도의 캐시 동작을 사용하면 각 콘텐츠 유형에 대한 캐싱을 최적화하여 캐시 적중률을 개선하고 오리진 로드를 줄일 수 있습니다. 적절한 캐시 정책이 없으면 불필요한 오리진 요청, 비용 증가 및 재생 성능 저하가 발생할 수 있습니다.

MediaPackage 캐시 제어 헤더를 올바르게 적용하고 캐싱을 최적화하려면:

1. CloudFront 콘솔에서 CloudFront 배포 설정을 엽니다.
2. 다양한 콘텐츠 유형에 대해 별도의 캐시 동작을 생성합니다.
 - 매니페스트 요청(*.m3u8, *.mpd)
 - 미디어 세그먼트(*.ts, *.mp4, *.m4s)
 - 초기화 세그먼트
3. 각 캐시 동작에 대해 다음 설정을 사용하여 캐시 정책을 생성하거나 선택합니다.
 - "원본 캐시 제어 헤더" 옵션 활성화
 - "원본 요청 정책"을 설정하여 필요한 헤더 전달
 - 콘텐츠 유형에 따라 쿼리 문자열 전달 구성

매니페스트 캐시 동작

매니페스트 요청(*.m3u8, *.mpd)의 경우:

- 경로 패턴: *.m3u8 및 *.mpd
- 캐시 정책: 오리진 캐시 제어 헤더 명예
- 쿼리 문자열: 특정 파라미터 전달(참조 [쿼리 문자열 전달 최적화](#))
- 헤더: 모든 헤더 전달(최소 요구 사항은 참조 [MediaTailor CDN 통합에 필요한 헤더](#))

미디어 세그먼트 캐시 동작

미디어 세그먼트(*.ts, *.mp4, *.m4s)의 경우:

- 경로 패턴: *.ts, *.mp4, *.m4s

- 캐시 정책: 오리진 캐시 제어 헤더 명예(14일 TTL)
- 쿼리 문자열: 없음(세그먼트는 쿼리 파라미터를 사용하지 않음)
- 압축: 전송 성능 향상

쿼리 문자열 전달 최적화

불필요한 쿼리 파라미터는 동일한 콘텐츠에 대해 여러 캐시 변형을 생성하므로 쿼리 문자열 최적화는 캐시 효율성에 매우 중요합니다. 각 고유 쿼리 파라미터 조합은 별도의 캐시 항목을 생성하여 캐시 적 중량을 줄이고 오리진 요청을 증가시킵니다. MediaPackage에서 실제로 사용하는 쿼리 문자열만 전달하면 전체 기능을 유지하면서 캐시 효율성을 극대화할 수 있습니다.

MediaPackage가 사용하는 쿼리 문자열만 전달하도록 CDN을 구성하여 캐시 효율성을 개선합니다.

필수 쿼리 문자열

`start` 및 `end` - 시간 이동 보기 창외 경우

`time_delay` - 매니페스트 콘텐츠에 시간 지연 적용

`_HLS_msn`, `_HLS_m` 및 `_HLS_part` - LL-HLS 재생 요청의 경우

기능별 쿼리 문자열

`aws.manifestfilter` - [매니페스트 필터링](#)용

Important

캐시 키에 다른 쿼리 문자열을 포함하지 마십시오. MediaPackage는 인식할 수 없는 파라미터를 무시하며, 이를 포함하면 불필요한 캐시 변형을 생성하여 캐시 효율성이 줄어듭니다.

성능 최적화 기법

이러한 최적화는 MediaPackage 또는 MediaTailor가 아닌 CDN(예: CloudFront)에 구성됩니다. 캐시 성능을 극대화하기 위해 다음과 같은 추가 최적화를 구현합니다.

오리진 실드

오리진 실드는 CDN 엣지 로케이션과 MediaPackage 엔드포인트 간에 추가 캐싱 계층을 제공합니다. 이렇게 하면 MediaPackage 엔드포인트에 도달하는 요청 수가 줄어들어 특히 트래픽이 급증하거나 캐

시 적중률이 최적보다 낮을 때 성능을 개선하고 비용을 절감할 수 있습니다. 오리진 실드는 여러 엣지 로케이션이 동일한 콘텐츠를 동시에 요청할 수 있는 라이브 스트리밍에 특히 유용합니다.

오리진 실드를 활성화하여 MediaPackage 엔드포인트의 부하를 줄입니다.

1. CloudFront 배포에서 MediaPackage 오리진에 대해 Origin Shield를 활성화합니다.
2. MediaPackage 엔드포인트와 가까운 오리진 실드 리전을 선택합니다.
3. 이렇게 하면 MediaPackage에 대한 요청을 줄이는 추가 캐싱 계층이 생성됩니다.

압축 구성

텍스트 기반 응답에 대해 압축을 활성화합니다.

- 매니페스트 파일에 대한 압축 활성화(*.m3u8, *.mpd)
- 미디어 세그먼트를 압축하지 마십시오(이미 압축됨).
- 모든 헤더가 MediaPackage로 전달되는지 확인합니다(최소 요구 사항은 참조 [MediaTailor CDN 통합에 필요한 헤더](#)).

캐시 성능 모니터링

이러한 주요 지표를 추적하여 캐시 성능을 최적화합니다.

캐시 적중률

대상: 미디어 세그먼트의 경우 90%, 매니페스트의 경우 70% 이상

비율이 낮으면 잘못된 TTL 설정 또는 불필요한 쿼리 파라미터를 나타낼 수 있습니다.

오리진 요청 볼륨

MediaPackage 엔드포인트에 대한 요청 모니터링

대용량은 캐싱 문제를 나타낼 수 있습니다.

캐시 키 변형

캐시 키 패턴을 검토하여 불필요한 변형 식별

변형이 너무 많으면 캐시 효율성이 저하됩니다.

이러한 캐시 최적화를 구현한 후 효율성을 추적하도록 모니터링을 설정합니다. 캐시 적중률, 오리진 요청 패턴 및 기타 주요 성능 지표 모니터링에 대한 지침은 섹션을 참조하세요 [MediaPackage, CDN 및 MediaTailor 통합의 성능 모니터링](#). 캐시 성능이 좋지 않거나 예기치 않은 오리진 요청이 관찰되는 경우 문제 해결 단계는 섹션을 참조 [MediaPackage, CDN 및 MediaTailor 통합 문제 해결](#)하세요.

MediaPackage, CDN 및 MediaTailor 통합의 성능 모니터링

AWS Elemental MediaTailor에서는 및 콘텐츠 전송 네트워크(CDN) 통합의 최적의 성능을 유지하기 위해 효과적인 모니터링이 AWS Elemental MediaPackage 필요합니다. 이 주제에서는 추적할 주요 지표, 사용할 모니터링 도구, 사전 문제 감지에 대한 알림을 설정하는 방법에 대한 지침을 제공합니다.

모니터링을 설정하기 전에 기본 통합이 올바르게 작동하는지 확인합니다. 기본 콘텐츠 전송 네트워크 통합 설정을 완료하지 않은 경우로 시작합니다 [MediaTailor를 MediaPackage 및 CDN과 통합](#). 모니터링을 통해 식별된 문제를 해결해야 하는 경우 섹션을 참조하세요 [CDN 통합 문제 해결](#).

주요 성능 지표

이러한 필수 지표를 모니터링하여 MediaPackage 및 CDN 통합의 최적의 성능을 보장합니다.

CDN 성능 지표

캐시 적중률 대상, 오리진 요청 볼륨 모니터링 및 응답 시간 벤치마크를 포함한 포괄적인 CDN 성능 지표는 CDN 최적화 가이드 [CDN 및 MediaTailor 통합의 성능 벤치마크](#)의 섹션을 참조하세요.

CDN 지표에 대한 주요 EMP별 고려 사항:

EMP 캐시 제어 헤더

확인 사항: CDN이 최적의 TTL 동작을 위해 EMP의 캐시 제어 헤더를 준수하는지 확인

예상 동작: EMP의 헤더에 따라 콘텐츠 유형에 따라 캐시 기간이 달라야 합니다.

EMP 캐시 최적화에 대한 자세한 지침은 섹션을 참조하세요 [MediaTailor 및 MediaPackage 콘텐츠 전송을 위한 CDN 캐싱 최적화](#).

쿼리 파라미터 영향

모니터링 대상: EMP별 쿼리 파라미터가 캐시 효율성에 미치는 영향 추적

최적화 대상: 필요한 EMP 쿼리 파라미터만 캐시 키에 포함되는지 확인

CDN 응답 시간

모니터링 대상: 다양한 콘텐츠 유형(매니페스트 대 세그먼트)에 대한 응답 시간을 추적합니다.

대상 값:

- 캐시된 콘텐츠: 100ms 미만
- 오리진 요청: 500ms 미만

MediaPackage 성능 지표

오류 발생률

모니터링 대상: CDN 및 MediaPackage 엔드포인트 모두에서 HTTP 오류율을 모니터링합니다. 구성 문제를 나타낼 수 있는 4xx 오류에 특히 주의하십시오.

키 오류 코드:

- 400 오류: 매니페스트 필터링 문제와 관련된 경우가 많음
- 404 오류: 라우팅 또는 구성 문제를 나타낼 수 있음
- 504 오류: 제한 시간 문제, 특히 LL-HLS 관련 문제

요청 볼륨 및 패턴

모니터링 대상: MediaPackage 엔드포인트에 대한 요청 패턴을 추적하여 사용 추세 및 용량 요구 사항을 식별합니다.

관찰할 패턴:

- 피크 사용 시간
- 요청의 지리적 분포
- 콘텐츠 유형 배포(라이브 vs. 온디맨드)

지연 시간 지표

End-to-end 지연 시간

모니터링할 사항: LL-HLS 구현의 경우 콘텐츠 수집부터 최종 사용자 재생까지의 end-to-end 지연 시간을 모니터링합니다. 지연 시간이 길면 CDN 구성 문제가 발생할 수 있습니다.

대상 값:

- LL-HLS: 3초 미만의 glass-to-glass 지연 시간
- 일반 HLS: 30초 미만

매니페스트 생성 시간

모니터링할 사항: 특히 필터링이 적용된 경우 MediaPackage에서 매니페스트를 생성하는 데 걸리는 시간입니다.

대상 값: 매니페스트 생성의 경우 200ms 미만

모니터링 도구 및 설정

포괄적인 모니터링 도구를 설정하는 것은 최적의 성능을 유지하고 최종 사용자에게 영향을 미치기 전에 문제를 신속하게 식별하는 데 필수적입니다. 적절한 모니터링이 없으면 최종 사용자가 재생 품질이 저하될 때까지 성능 저하, 캐시 비효율성 또는 통합 문제가 눈에 띄지 않을 수 있습니다. 올바른 모니터링 설정은 MediaPackage 및 CDN 통합의 모든 측면에 대한 가시성을 제공합니다.

다음 AWS 서비스 및 도구를 사용하여 MediaPackage 및 CDN 통합을 모니터링합니다.

Amazon CloudWatch

Amazon CloudWatch는 두 서비스에서 지표를 수집하고 저장하여 MediaPackage 및 CDN 통합을 모니터링하는 기반을 제공합니다. 적절한 CloudWatch 구성을 사용하면 성능 추세를 식별하고, 문제를 해결하고, 통합을 최적화하는 데 필요한 데이터를 확보할 수 있습니다. CloudWatch 지표가 없으면 시스템 성능에 대한 가시성이 부족해지고 문제가 심각해질 때까지 감지되지 않을 수 있습니다.

포괄적인 지표 수집을 위한 CloudWatch 모니터링을 설정합니다.

1. MediaPackage 지표: MediaPackage 엔드포인트에 대한 CloudWatch 지표를 활성화하여 요청 볼륨, 오류율 및 응답 시간을 추적합니다.
2. CDN 지표: 캐시 적중률, 오리진 요청 수 및 오류율을 포함한 CloudFront 지표를 수집하도록 CloudWatch를 구성합니다.
3. 사용자 지정 지표: 최종 사용자 참여 또는 콘텐츠 인기도와 같은 비즈니스별 KPIs에 대한 사용자 지정 지표를 생성합니다.

CloudWatch 대시보드

포괄적인 대시보드를 생성하여 지표를 시각화합니다.

1. 개요 대시보드: 전체 시스템 상태를 보여주는 상위 수준 지표
2. CDN 성능 대시보드: 캐시 성능 및 지리적 배포를 포함한 세부 CDN 지표

3. MediaPackage 성능 대시보드: 요청 패턴 및 오류율을 포함한 MediaPackage별 지표
4. 지연 시간 대시보드: 다양한 콘텐츠 유형 및 리전에 대한 End-to-end 지연 시간 지표

로그 분석

자세한 문제 해결을 위해 로그 분석을 설정합니다.

1. CDN 액세스 로그: CDN 액세스 로그를 활성화 및 분석하여 요청 패턴을 이해하고 문제를 식별합니다.
2. MediaPackage CloudWatch 로그: 오류 및 성능 문제에 대한 MediaPackage 로그 모니터링
3. 로그 집계: Amazon CloudWatch Logs Insights 또는 타사 도구를 사용하여 로그 패턴 분석

알림 및 알림 설정

알림 구성은 사전 문제 감지 및 해결에 매우 중요합니다. 적절한 알림이 없으면 최종 사용자 경험에 상당한 영향을 미치거나 서비스 중단을 일으킬 때까지 문제가 눈에 띄지 않을 수 있습니다. 잘 구성된 알림은 최종 사용자에게 영향을 미치기 전에 문제를 식별하고 해결하고 팀에 즉각적인 조치가 필요한 중요한 문제를 알리는 데 도움이 됩니다.

최종 사용자에게 영향을 미치기 전에 문제를 식별하도록 사전 예방 알림을 구성합니다.

중요 알림

중요한 문제에 대한 즉각적인 알림을 설정합니다.

- 높은 오류율: 5분 동안 4xx 또는 5xx 오류율이 5%를 초과하면 알림
- 캐시 적중률 저하: 캐시 적중률이 매니페스트의 경우 70%, 세그먼트의 경우 85% 미만으로 떨어질 때 알림
- 높은 지연 시간: end-to-end 지연 시간이 목표 임계값을 초과할 때 알림
- 오리진 요청 급증: 오리진 요청이 기준과 비교하여 50% 이상 증가할 때 알림

경고 알림

문제 발생을 나타낼 수 있는 추세에 대한 경고 알림을 설정합니다.

- 점진적 성능 저하: 30분 동안 응답 시간이 20% 증가할 때 알림
- 캐시 효율성 추세: 캐시 적중률이 시간 경과에 따른 감소 추세를 표시할 때 알림

- 비정상적인 트래픽 패턴: 요청 볼륨 또는 지리적 배포의 예기치 않은 변경에 대한 알림

최적화를 위해 모니터링 데이터 사용

모니터링 데이터를 활용하여 성능을 지속적으로 개선합니다.

정기적인 성능 검토

1. 주간 검토: 주간 성능 추세 분석 및 최적화 기회 식별
2. 월별 용량 계획: 트래픽 패턴을 사용하여 용량 요구 사항 및 CDN 최적화 계획
3. 분기별 아키텍처 검토: 전반적인 아키텍처 효율성을 평가하고 개선이 필요한 영역을 식별합니다.

일반적인 최적화 작업

데이터 모니터링을 기반으로 다음 최적화 작업을 고려합니다.

- 캐시 정책 조정: 실제 콘텐츠 업데이트 패턴을 기반으로 TTL 값을 수정합니다. 자세한 TTL 최적화 지침은 섹션을 참조하세요 [CDN 및 MediaTailor 통합을 위한 캐싱 최적화](#).
- 지리적 최적화: 트래픽이 많은 리전에 CDN 엣지 로케이션 추가
- 쿼리 파라미터 최적화: 캐시를 조각화하는 불필요한 쿼리 파라미터 제거
- 오리진 실드 구성: 오리진 요청 볼륨이 많은 리전에서 오리진 실드 구현

MediaPackage와 관련된 자세한 모니터링 지침은 [MediaPackage 사용 설명서의 MediaPackage 모니터링](#)을 참조하세요. MediaPackage

MediaPackage, CDN 및 MediaTailor 통합 문제 해결

AWS Elemental MediaTailor AWS Elemental MediaPackage 및 콘텐츠 전송 네트워크(CDN)와 통합하면 재생, 캐싱 또는 기타 통합 기능에 영향을 미치는 일반적인 문제가 발생할 수 있습니다. 재생 문제, 캐싱 문제 또는 기타 통합 관련 오류가 발생할 때 이 가이드를 사용합니다.

모든 MediaTailor 구현에 적용되는 범용 캐시 성능 문제, HTTP 오류 해결, 테스트 절차 및 진단 기술을 포함한 포괄적인 CDN 문제 해결 지침은 섹션을 참조하세요 [CDN 통합 문제 해결](#). 이 섹션에서는 MediaPackage 관련 문제 해결 요구 사항에 중점을 둡니다.

문제를 해결하기 전에 기본 통합 설정을 올바르게 완료했는지 확인합니다. 아직 통합을 설정하지 않았거나 설정 단계를 검토해야 하는 경우 섹션을 참조하세요 [MediaTailor를 MediaPackage 및 CDN과 통합](#). 문제 해결 후 캐시 성능 최적화에 대한 지침은 섹션을 참조하세요 [CDN 캐싱](#).

매니페스트 필터링 오류

문서화된 오류 조건에 따른 MediaPackage 매니페스트 필터링 기능 관련 문제:

매니페스트 필터링과 관련된 HTTP 400 오류

증상: `aws.manifestfilter` 파라미터가 있는 요청은 HTTP 400 잘못된 요청을 반환합니다.

검증된 원인(설명서에서 AWS):

- 필터 기준은 빈 매니페스트를 생성합니다(필터 조건과 일치하는 스트림 없음).
- 잘못된 필터 파라미터 이름 또는 값
- 형식이 잘못된 쿼리 문자열 구문
- 중복 또는 반복 필터 파라미터
- 필터 파라미터 문자열이 1,024자를 초과함
- 미디어 재생 목록 또는 세그먼트에 적용된 쿼리 파라미터(지원되지 않음)

솔루션:

1. 필터 파라미터를 검토하여 사용 가능한 콘텐츠 스트림과 일치하는지 확인합니다. 필터링으로 일치하는 스트림이 없는 경우 MediaPackage는 HTTP 400을 반환합니다.
2. 지원되는 파라미터 이름 및 값 형식과 비교하여 필터 구문을 검증합니다.
3. 쿼리 문자열에 중복 파라미터가 있는지 확인합니다.
4. 필터 파라미터가 미디어 재생 목록 또는 세그먼트가 아닌 다변량 재생 목록에만 적용되는지 확인합니다.
5. 총 파라미터 문자열이 1024자 미만인지 확인합니다.

참조: [AWS Elemental MediaPackage 매니페스트 필터링 오류 조건](#)

매니페스트 필터링이 작동하지 않음(HTTP 200이지만 필터링이 적용되지 않음)

증상: 요청은 HTTP 200을 반환하지만 매니페스트에는 필터링된 하위 집합 대신 모든 스트림이 포함됩니다.

가능한 원인:

- MediaPackage에 `aws.manifestfilter` 쿼리 파라미터를 전달하지 않는 CDN
- 사용 가능한 스트림에서 필터 파라미터를 찾을 수 없음(HTTP 200으로 필터링되지 않은 매니페스트 반환)

솔루션:

1. CDN 캐시 정책이 전달된 쿼리 문자열 목록에 `aws.manifestfilter`를 포함하는지 확인합니다.
2. MediaPackage 엔드포인트(CDN 우회)에 대해 직접 필터 파라미터를 테스트하여 예상대로 작동하는지 확인합니다.
3. 필터 값이 콘텐츠 스트림의 실제 특성과 일치하는지 확인합니다.

참조: [AWS Elemental MediaPackage 매니페스트 필터링 오류 조건](#)

진단 절차

체계적인 진단 절차를 통해 통합 문제의 근본 원인을 빠르고 효율적으로 식별할 수 있습니다. 구조화된 접근 방식을 따르면 잘못된 가정으로 인한 시간 낭비를 방지하고 증상보다는 실제 문제를 해결할 수 있습니다. 이러한 증거 기반 진단 단계는 문제를 격리하고 적절한 솔루션으로 안내하도록 설계되었습니다.

다음 증거 기반 진단 단계에 따라 문제를 식별합니다.

캐시 성능 분석

캐시 성능 분석은 EMP 통합에 매우 중요합니다. 캐시 효율성이 좋지 않으면 오리진 로드가 증가하고 비용이 증가하며 재생 문제가 발생할 수 있기 때문입니다.

캐시 적중률 분석, 캐시 키 최적화 및 체계적인 진단 단계를 포함한 포괄적인 캐시 성능 문제 해결은 기본 CDN 문제 해결 안내서 [CDN 캐시 성능 문제](#)의 섹션을 참조하세요.

EMP별 캐시 고려 사항:

- EMP 캐시 제어 헤더: CDN이 EMP의 캐시 제어 헤더를 재정의하지 않고 준수하는지 확인합니다.
- EMP 쿼리 파라미터: 필요한 EMP 쿼리 파라미터만 캐시 키에 포함되는지 확인합니다.
- EMP TTL 동작: 다양한 EMP 콘텐츠 유형에 적절한 캐시 기간이 있는지 확인

EMP 캐시 정책 및 TTL 설정 최적화에 대한 자세한 지침은 섹션을 참조하세요 [MediaTailor 및 MediaPackage 콘텐츠 전송을 위한 CDN 캐싱 최적화](#).

매니페스트 필터링 구성 검증

매니페스트 필터링 검증은 필터링 문제로 인해 최종 사용자가 잘못된 콘텐츠, 지원되지 않는 형식 또는 액세스해서는 안 되는 콘텐츠를 수신할 수 있기 때문에 필수적입니다. 체계적 테스트는 문제가 CDN 구성, 필터 파라미터 구문 또는 콘텐츠 가용성과 관련이 있는지 식별하는 데 도움이 됩니다.

매니페스트 필터링 기능을 체계적으로 테스트합니다.

1. MediaPackage 엔드포인트(CDN 우회)에 대해 직접 필터 파라미터를 테스트하여 올바르게 작동하는지 확인합니다.
2. 필터링된 매니페스트와 필터링되지 않은 매니페스트를 비교하여 예상 스트림이 포함/제외되었는지 확인합니다.
3. CDN 캐시 정책이 `aws.manifestfilter` 쿼리 파라미터를 전달하는지 확인합니다.
4. HTTP 400 오류를 확인하고 문서화된 오류 조건과 일치시킵니다.

문제 해결 후 매니페스트 필터링을 구현하거나 수정해야 하는 경우 전체 설정 지침은 섹션을 참조 [MediaTailor, MediaPackage 및 CDN을 사용하여 매니페스트 필터링 설정](#)하세요.

쿼리 파라미터 구성 검증

CDN이 필요한 쿼리 파라미터만 전달하는지 확인합니다.

1. CDN 캐시 정책을 검토하여 AWS 권장 파라미터만 포함하는지 확인합니다.
 - `aws.manifestfilter` - 매니페스트 필터링용
 - `aws.manifestsettings` - 시간 이동 보기
 - `_HLS_msn` 및 `_HLS_part` - LL-HLS 지원용
2. MediaPackage는 쿼리 파라미터를 무시하고 캐시 효율성을 줄이므로 캐시 키에서 다른 쿼리 파라미터를 제거합니다.

참조: [AWS Elemental MediaPackage 및 CDN의 작업](#)

오류 코드 참조

문서화된 오류 조건 및 원인에 대한 참조:

HTTP 400 잘못된 요청(매니페스트 필터링)

문서화된 원인:

- 필터를 적용하면 빈 매니페스트가 생성됩니다.
- 잘못된 파라미터 이름 또는 값
- 형식이 잘못된 쿼리 문자열 구문
- 중복 필터 파라미터

- 파라미터 문자열이 1024자를 초과함
- 미디어 재생 목록 또는 세그먼트의 쿼리 파라미터

참조: [MediaPackage 매니페스트 필터링 오류 조건](#)

HTTP 200 OK(필터링이 적용되지 않음)

문서화된 원인:

- 사용 가능한 스트림에서 필터 파라미터를 찾을 수 없음(필터링되지 않은 매니페스트 반환)
- 필터링 후 존재하는 자막 스트림만(필터링되지 않은 매니페스트 반환)

참조: [MediaPackage 매니페스트 필터링 오류 조건](#)

추가 문제 해결 리소스

이 주제에서 다루지 않은 문제는 다음 공식 AWS 리소스를 참조하세요.

- [에서 매니페스트 미리 보기 AWS Elemental MediaPackage](#) - 매니페스트 미리 보기를 사용하여 콘텐츠 패키징 문제 해결
- [CloudFront 캐시 적중률 증가](#) - CDN 캐시 최적화에 대한 포괄적인 가이드
- [매니페스트 필터링](#) - MediaPackage 필터링 기능에 대한 전체 가이드

Amazon CloudFront AWS Elemental MediaTailor 와 통합

AWS Elemental MediaTailor 는 Amazon CloudFront와 통합되어 콘텐츠 전송 성능과 신뢰성을 개선합니다. CloudFront는 엣지 로케이션이라는 전 세계 데이터 센터 네트워크를 통해 콘텐츠를 배포하는 콘텐츠 전송 네트워크(CDN)입니다. 최종 사용자가 MediaTailor에서 콘텐츠를 요청하면 CloudFront는 가장 가까운 엣지 로케이션으로 요청을 라우팅합니다. 이 접근 방식은 지연 시간을 줄이고 최종 사용자의 성능을 개선합니다.

MediaTailor를 CloudFront와 통합하면 다음과 같은 몇 가지 주요 이점이 있습니다.

- 개인화된 콘텐츠에 액세스하는 시청자의 지연 시간 단축
- 대규모 대상 크기 처리를 위한 확장성 향상
- 중복 전송 경로를 통한 안정성 향상
- 효율적인 캐싱 전략을 통한 비용 최적화
- Media Quality-Aware Resiliency(MQAR)를 사용한 다중 리전 장애 조치와 같은 고급 기능

CloudFront 기능에 대한 자세한 내용은 [CloudFront 개발자 안내서](#)를 참조하세요. CloudFront 요금에 대한 자세한 내용은 [CloudFront 요금](#)을 참조하세요.

간결하게 하기 위해 "매니페스트"를 사용하여 다변량 재생 목록, 미디어 재생 목록 및 MPDs를 총체적으로 참조하는 경우가 있습니다.

주제

- [MediaTailor와의 기본 CloudFront 통합 설정](#)
- [CloudFront 기능으로 MediaTailor 성능 최적화](#)
- [MQAR을 사용하여 MediaTailor에 대한 다중 리전 복원력 구현](#)
- [CloudFront 및 MediaTailor 통합 모니터링 및 문제 해결](#)

MediaTailor와의 기본 CloudFront 통합 설정

AWS Elemental MediaTailor Amazon CloudFront와 통합하면 최종 사용자의 콘텐츠 전송 성능이 향상됩니다. 이 주제에서는 MediaTailor에 대한 기본 CloudFront 배포를 설정하는 방법을 안내합니다. 이 통합을 통해 최종 사용자는 CloudFront 네트워크를 통해 개인화된 콘텐츠에 액세스할 수 있습니다. 또한 다양한 콘텐츠 유형에 적합한 캐싱을 구성하는 방법도 알아봅니다.

권한 부여 및 라우팅을 위해 CloudFront를 통해 쿼리 파라미터를 전달하는 방법에 대한 자세한 내용은 섹션을 참조하세요 [MediaTailor 매니페스트 쿼리 파라미터](#). 동적 변수를 사용한 고급 라우팅은 섹션을 참조하세요 [여러 콘텐츠 소스에 대한 MediaTailor 도메인 변수](#).

사전 조건

MediaTailor로 CloudFront를 구성하기 전에 다음이 있는지 확인합니다.

- CloudFront 배포를 생성하고 관리할 수 있는 권한이 있는 활성 AWS 계정
- 구성된 MediaTailor 재생 구성(참조 [AWS Elemental MediaTailor 를 사용하여 광고 삽입](#))
- 콘텐츠 오리진 서버를 올바르게 설정하고 액세스할 수 있습니다.
- 비디오 스트리밍 개념에 대한 기본 이해(HLS/DASH)

CloudFront 배포 구성

다음 단계에 따라 MediaTailor용 CloudFront 배포를 생성하고 구성합니다.

MediaTailor용 CloudFront 배포를 생성하려면

1. AWS Management Console에 로그인하고 <https://console.aws.amazon.com/cloudfront/v3/home> CloudFront 콘솔을 엽니다.
2. 배포 생성(Create Distribution)을 선택합니다.
3. 오리진 도메인에 MediaTailor 엔드포인트 URL(예: a1b2c3d4.mediataylor.us-west-2.amazonaws.com)을 입력합니다.
4. 프로토콜에서 HTTPS만 선택합니다.
5. 이름에이 오리진을 식별하는 데 도움이 되는 이름(예: mediataylor-origin)을 입력합니다.
6. 기본 캐시 동작 설정을 구성합니다.
 - a. 경로 패턴에는 기본값()을 사용합니다*.
 - b. 객체 자동 압축에서 예를 선택합니다.
 - c. 뷰어 프로토콜 정책에서 HTTP를 HTTPS로 리디렉션을 선택합니다.
 - d. 허용된 HTTP 메서드에서 GET, HEAD를 선택합니다.
 - e. 캐시 정책에서 CachingDisabled 선택합니다.
 - f. 오리진 요청 정책에서 AllViewer를 선택하여 기본 동작에 대한 모든 헤더를 전달합니다.

Note

기본 동작은 AllViewer를 사용하여 특정 경로 패턴과 일치하지 않는 모든 콘텐츠를 안전하게 처리합니다. 매니페스트 및 세그먼트에 대한 특정 캐시 동작은 적절한 정책을 사용하여 별도로 구성됩니다.

7. 배포 설정을 구성합니다.
 - a. 가격 클래스에서 대상 위치와 가장 일치하는 옵션을 선택합니다.
 - b. AWS WAF 웹 ACL의 경우 기존 웹 ACL을 선택하거나 보안 보호 활성화 안 함으로 둡니다.
 - c. 기본 루트 객체의 경우 비워 둡니다.
 - d. 표준 로깅에서 켜기를 선택하여 로깅을 활성화합니다.
8. 배포 생성(Create Distribution)을 선택합니다.

캐시 동작 구성

배포를 생성한 후에는 다양한 유형의 콘텐츠를 적절하게 처리하도록 추가 캐시 동작을 구성해야 합니다. 이 섹션에서는 CloudFront의 기본 캐시 동작 설정을 다룹니다.

고급 TTL 설정, 캐시 키 구성 및 성능 튜닝을 포함한 포괄적인 캐싱 최적화는 CDN 최적화 가이드 [CDN 및 MediaTailor 통합을 위한 캐싱 최적화](#)의 섹션을 참조하세요.

매니페스트 캐시 동작 구성

매니페스트에는 개인 맞춤형 콘텐츠가 포함되어 있으므로 캐시하지 마세요. 다음 단계에 따라 캐시 동작을 구성합니다.

매니페스트 캐시 동작을 구성하려면

1. CloudFront 콘솔에서 배포를 선택합니다.
2. 동작 탭을 선택합니다.
3. 동작 만들기(Create behavior)를 선택합니다.
4. 경로 패턴에 *.m3u8를 입력하여 HLS 다변량 및 미디어 재생 목록과 일치시킵니다.
5. 오리진에서 MediaTailor 오리진을 선택합니다.
6. 캐시 정책에서 CachingDisabled 선택합니다.
7. 오리진 요청 정책에서 AllViewer를 선택하여 동적 콘텐츠에 필요한 모든 헤더를 전달합니다.
8. 생성(Create)을 선택합니다.
9. 경로 패턴을 사용하여 MPDs*.mpd와 일치하는 DASH 매니페스트에 대해 이 단계를 반복합니다.

이 구성을 사용하면 각 최종 사용자가 특정 광고 콘텐츠가 포함된 개인화된 매니페스트를 수신할 수 있습니다. CDN은 이러한 매니페스트를 캐싱하지 않으므로 각 요청은 MediaTailor로 직접 전달됩니다.

세그먼트 캐시 동작 구성

성능을 최적화하고 적절한 CORS 처리를 보장하기 위해 광고 세그먼트 및 콘텐츠 세그먼트에 대해 별도의 캐시 동작을 구성합니다.

광고 세그먼트 캐시 동작 구성

/tm/* 경로 패턴을 통해 제공되는 광고 세그먼트는 CORS를 올바르게 처리하려면 특정 구성이 필요합니다. 다음 단계를 따릅니다.

광고 세그먼트 캐시 동작을 구성하려면

1. CloudFront 콘솔에서 배포를 선택합니다.
2. 동작 탭을 선택합니다.
3. 동작 만들기(Create behavior)를 선택합니다.
4. 경로 패턴/`/tm/*`에 MediaTailor에서 제공하는 광고 세그먼트와 일치하도록 입력합니다.
5. 오리진에서 MediaTailor 세그먼트 오리진
(`segments.mediatailor.region.amazonaws.com`호스트 이름 사용)을 선택합니다.
6. 캐시 정책에서 `CachingOptimized` 선택합니다.
7. 오리진 요청 정책에서 `없음`을 선택합니다.
8. 응답 헤더 정책에서 `CORS-with-preflight-and-SecurityHeadersPolicy`를 선택하여 응답에 적절한 CORS 헤더가 포함되도록 합니다.
9. 생성(Create)을 선택합니다.

콘텐츠 세그먼트 캐시 동작 구성

콘텐츠 세그먼트는 최적의 성능을 위해 표준 캐싱 정책을 사용할 수 있습니다. 다양한 세그먼트 형식에 대해 별도의 동작을 구성합니다.

콘텐츠 세그먼트 캐시 동작을 구성하려면

1. CloudFront 콘솔에서 배포를 선택합니다.
2. 동작 탭을 선택합니다.
3. 동작 만들기(Create behavior)를 선택합니다.
4. 경로 패턴에 `*.ts`를 입력하여 HLS 콘텐츠 세그먼트와 일치시킵니다.
5. 오리진에서 콘텐츠 오리진을 선택합니다.
6. 캐시 정책에서 `CachingOptimized` 선택합니다.
7. 오리진 요청 정책에서 `없음`을 선택합니다.
8. 응답 헤더 정책에서 `CORS-with-preflight-and-SecurityHeadersPolicy`를 선택하여 모든 콘텐츠 유형에서 일관된 CORS 처리를 보장합니다.
9. 생성(Create)을 선택합니다.
10. 적절한 경로 패턴을 사용하여 다른 콘텐츠 세그먼트 형식에 대해 다음 단계를 반복합니다.
 - `*.mp4` MP4 세그먼트용

- *.m4s DASH 세그먼트용
- *.cmfv CMAF 세그먼트*.cmfa의 경우 및

이 구성을 통해 광고 세그먼트와 콘텐츠 세그먼트가 적절한 CORS 처리로 적절하게 캐싱됩니다. 광고 세그먼트는 CORS 보호와 함께 MediaTailor 세그먼트 오리진을 사용하는 반면, 콘텐츠 세그먼트는 최적화된 캐싱 정책과 함께 콘텐츠 오리진을 사용합니다.

MediaTailor 구성 업데이트

CloudFront 배포를 설정한 후 CloudFront 도메인을 사용하도록 MediaTailor 구성을 업데이트합니다.

MediaTailor 구성을 업데이트하려면

1. [MediaTailor 콘솔](#)을 엽니다.
2. 업데이트할 구성을 선택합니다.
3. CDN 구성 섹션의 CDN 콘텐츠 세그먼트 접두사 필드에 CloudFront 배포 도메인 이름(예: d1234abcdef.cloudfront.net)을 입력합니다.
4. 변경 내용을 저장합니다.

이 구성을 사용하면 MediaTailor는 오리진을 직접 가리키는 대신 CloudFront 배포를 가리키는 URLs을 사용하여 매니페스트를 생성합니다.

통합 테스트

CloudFront 배포를 구성하고 MediaTailor 구성을 업데이트한 후 통합을 테스트합니다.

CloudFront 및 MediaTailor 통합을 테스트하려면

1. CloudFront 배포를 통해 매니페스트를 요청합니다(예: https://d1234abcdef.cloudfront.net/v1/master/12345/my-config/index.m3u8).
2. 매니페스트에 CloudFront 도메인을 가리키는URLs이 포함되어 있는지 확인합니다.
3. 비디오 플레이어를 통해 콘텐츠를 재생하고 콘텐츠와 광고가 모두 올바르게 재생되는지 확인합니다.
4. CloudFront 로그를 확인하여 요청이 올바르게 라우팅되고 있는지 확인합니다.

구성의 예

다음은 적절한 캐시 동작이 있는 MediaTailor에 대한 CloudFront 배포 구성의 예입니다.

Example CloudFront 배포 구성 예제

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

이 예제는 다음을 보여줍니다.

- 기본 동작: `CachingDisabled` 및를 사용하여 특정 경로 패턴과 일치하지 않는 콘텐츠를 `AllViewer` 안전하게 처리합니다.
- 매니페스트 동작(*.m3u8): 동적 콘텐츠 `AllViewer`에 `CachingDisabled` 및를 사용합니다.
- 세그먼트 동작(*.ts): `CachingOptimized` 및를 사용합니다. `CORS-with-preflight-and-SecurityHeadersPolicy`

다음 단계

MediaTailor와의 기본 CloudFront 통합을 설정한 후 다음 단계를 고려하세요.

- 추가 CloudFront 기능으로 성능 최적화(참조 [CloudFront 기능으로 MediaTailor 성능 최적화](#))
- MQAR을 사용하여 다중 리전 복원력 구현(참조 [MQAR을 사용하여 MediaTailor에 대한 다중 리전 복원력 구현](#))
- 모니터링 및 문제 해결 설정(참조 [CloudFront 및 MediaTailor 통합 모니터링 및 문제 해결](#))

CloudFront 기능으로 MediaTailor 성능 최적화

AWS Elemental MediaTailor 기본 구성 이외의 추가 기능을 통해 Amazon CloudFront의 성능을 향상시킬 수 있습니다. MediaTailor를 사용하여 기본 CloudFront 구성을 설정한 후 추가 기능을 구현하여 성능, 안정성 및 사용자 지정 옵션을 개선할 수 있습니다. 이러한 최적화는 더 나은 시청 환경을 제공하는 데 도움이 됩니다. 또한 비용과 오리진 부하를 줄입니다.

동적 라우팅 및 파라미터 처리를 사용한 성능 최적화는 [섹션을 참조하세요](#) [ADS 요청에 대한 MediaTailor 동적 광고 변수](#). 쿼리 파라미터 최적화 전략은 [섹션을 참조하세요](#) [MediaTailor 매니페스트 쿼리 파라미터](#).

Origin Shield를 사용하여 오리진 로드 줄이기

Origin Shield는 CloudFront 엣지 로케이션과 오리진 서버 사이에 캐싱 계층을 추가합니다. 이 기능은 라이브 스트리밍 및 인기 VOD 콘텐츠에 유용합니다. 많은 시청자가 동시에 동일한 콘텐츠를 요청할 때 도움이 됩니다.

Origin Shield는 여러 엣지 로케이션의 요청을 통합하여 MediaTailor 및 콘텐츠 오리진에 대한 부하를 줄입니다.

MediaTailor 오리진에 대해 Origin Shield를 활성화하려면

1. CloudFront 콘솔을 열고 배포로 이동합니다.

2. MediaTailor 재생 구성을 가리키는 오리진을 선택합니다.
3. Origin Shield에서 예를 선택합니다.
4. 드롭다운 메뉴에서 MediaTailor 엔드포인트에 가장 가까운 AWS 리전을 선택합니다.
5. 변경 내용을 저장합니다.

트래픽이 많은 이벤트의 경우 Origin Shield는 오리진에 대한 요청 부하를 크게 줄이고 신뢰성을 개선합니다. 자세한 지침은 CloudFront 개발자 안내서의 [Origin Shield 활성화](#)를 참조하세요.

CloudFront 함수를 사용하여 콘텐츠 전송 사용자 지정

CloudFront 함수를 사용하면 엣지에서 경량 JavaScript 코드를 실행하여 최종 사용자 요청 및 응답을 수정할 수 있습니다. URL 수정, 헤더 조작 또는 기본 인증과 같은 간단한 사용자 지정에 이러한 함수를 사용할 수 있습니다. MediaTailor 워크플로의 경우 함수는 복잡한 처리가 필요하지 않은 작업을 지원합니다.

CloudFront 함수는 엣지에서 콘텐츠 전송을 사용자 지정하는 간단한 방법을 제공합니다. MediaTailor 통합을 위해 이를 구현하는 방법은 다음과 같습니다.

MediaTailor용 CloudFront 함수를 구현하려면

1. CloudFront 콘솔에서 함수로 이동합니다.
2. 새 함수를 생성하고 적절한 용도를 선택합니다.
 - URL 조작 - MediaTailor에 도달하기 전에 다변량 재생 목록, 미디어 재생 목록 및 MPD 요청 URLs 수정하려면
 - 헤더 조작 - 요청 헤더를 추가하거나 수정하려면
 - 단순 인증 - 토큰 또는 쿼리 파라미터를 검증하려면
3. JavaScript 함수 코드를 작성합니다.
4. 샘플 MediaTailor 요청으로 함수를 테스트합니다.
5. 함수를 게시하고 배포의 캐시 동작과 연결합니다.

Example URL 정규화를 위한 샘플 CloudFront 함수

```
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;
}

```

자세한 내용과 코드 예제는 [CloudFront 개발자 안내서의 CloudFront 함수를 사용하여 엣지에서 사용자 지정을 참조하세요](#). CloudFront

Lambda@Edge를 사용하여 고급 사용자 지정 구현

CloudFront Functions가 제공할 수 있는 것보다 더 복잡한 처리 기능이 필요한 경우 Lambda@Edge를 사용합니다. 이 서비스를 사용하면 CloudFront 엣지 로케이션에서 Node.js 또는 Python 함수를 실행할 수 있습니다.

Lambda@Edge 함수는 복잡한 인증, 대규모 응답 수정 또는 타사 API 통합과 같은 정교한 작업을 수행할 수 있습니다.

보다 복잡한 사용자 지정을 위해 MediaTailor 및 CloudFront 통합과 함께 Lambda@Edge 함수를 사용합니다.

MediaTailor를 사용하여 Lambda@Edge를 구현하려면

1. 미국 동부(버지니아 북부) 리전에 Lambda 함수를 생성합니다.
2. 다음 사용 사례 중 하나에 대한 함수 코드를 작성합니다.
 - URL 조작 - MediaTailor에 도달하기 전에 다변량 재생 목록, 미디어 재생 목록 및 MPD 요청 URLs 수정하려면
 - A/B 테스트 - 사용자를 다른 광고 결정 서버로 라우팅하려면
 - 인증 요청 - 인증 헤더를 추가하려면
 - 응답 헤더 수정 - CORS 헤더를 추가하려면
3. 함수 버전을 게시하고 함수 별칭을 생성합니다.
4. 함수를 적절한 트리거 지점(최종 사용자 요청 또는 최종 사용자 응답)에서 CloudFront 배포와 연결합니다.

Note

MediaTailor와 함께 Lambda@Edge를 사용하는 경우 현재 호환되지 않으므로 Media Quality-Aware Resiliency(MQAR) 기능을 사용할 계획이라면 오리진 대상 트리거(오리진 요청 및 오리진 응답)를 사용하지 마세요.

자세한 내용과 코드 예제는 CloudFront 개발자 안내서 [의 Lambda@Edge를 사용하여 엣지에서 사용자 지정을 참조하세요.](#)

추가 성능 최적화 팁

성능을 더욱 개선하려면 다음과 같은 추가 최적화를 고려하세요.

캐시 적중률 최적화

CloudFront 지표에서 캐시 적중률을 모니터링하고 개선할 기회를 찾습니다.

- URL 패턴을 표준화하여 캐시 키 일관성 향상
- 쿼리 문자열 화이트리스트를 사용하여 캐시 키에 필요한 파라미터만 포함
- CloudFront 함수를 사용하여 URL 정규화 구현 고려

지연 시간 단축

다음 기술을 구현하여 지연 시간을 최소화합니다.

- 텍스트 기반 응답에 대해 Brotli 압축 활성화
- 연결 효율성 향상을 위해 HTTP/2 또는 HTTP/3 사용
- 최신 네트워크에 대한 IPv6 지원 활성화 고려

비용 최적화

성능과 비용 효율성의 균형을 맞춥니다.

- Origin Shield를 사용하여 중복 오리진 요청 감소
- 대상 지역에 따라 가격 등급 선택 고려
- 세그먼트에 대해 공격적인 캐싱을 구현하여 오리진 트래픽 감소

다음 단계

CloudFront 기능으로 성능을 최적화한 후 다음 단계를 고려하세요.

- MQAR을 사용하여 다중 리전 복원력 구현(참조[MQAR을 사용하여 MediaTailor에 대한 다중 리전 복원력 구현](#))
- 모니터링 및 문제 해결 설정(참조[CloudFront 및 MediaTailor 통합 모니터링 및 문제 해결](#))

MQAR을 사용하여 MediaTailor에 대한 다중 리전 복원력 구현

AWS Elemental MediaTailor MediaTailor 제공하는 데 도움이 되는 고급 Amazon CloudFront 기능인 Media Quality-Aware Resiliency(MQAR)를 통해 다중 리전 복원력이 향상되었습니다. 서로 다른 AWS 리전에 오리진이 여러 개 있는 경우 품질 점수가 가장 높은 오리진을 자동으로 선택합니다. 이 기능은 중단 없는 서비스가 필요한 라이브 스트리밍에 특히 유용합니다.

MQAR 작동 방식

MQAR을 사용하면 CloudFront에서 품질 점수가 가장 높은 오리진을 자동으로 선택할 수 있습니다. 이렇게 하면 시청자가 최상의 스트리밍 경험을 누릴 수 있습니다. 올바르게 구성되면 MQAR은 다음과 같은 이점을 제공합니다.

- 최고 품질의 오리진 자동 선택
- 중단 중 AWS 리전 간 원활한 장애 조치
- 중단을 최소화하면서 시청자 경험 개선
- 실시간 품질 모니터링 및 적응

1단계: MQAR 요구 사항 확인

MQAR을 구현하기 전에 인프라가 이러한 요구 사항을 충족하는지 확인합니다. MQAR은 여러 오리진의 품질 점수를 비교하여 작동하므로 여러 AWS 리전에서 동일한 콘텐츠를 사용할 수 있어야 합니다.

- 에포크 잠금 CMAF 수집 스트림 세트를 사용하여 정렬된 수집 스트림을 모든 MediaPackage 채널로 전송하는 인코더
- 서로 다른 AWS 리전에 동일한 오리진 엔드포인트가 있는 두 개의 동일한 MediaPackage 채널
- MediaPackage 채널에 대한 CMAF 수집(기본적으로 활성화됨)
- MQAR을 지원하도록 구성된 CloudFront 배포
- 각 MediaPackage 엔드포인트에 대한 MediaTailor 구성 MediaPackage

2단계: MQAR용 인코더 구성

MQAR이 효과적으로 작동하려면 인코더가 모든 리전에서 일관되고 동기화된 출력을 생성해야 합니다. 이러한 일관성을 통해 CloudFront는 오리진 간에 정확한 품질을 비교할 수 있습니다.

다음 설정을 사용하여 MediaLive 출력을 구성합니다.

- CMAF 출력 그룹의 모든 비디오 프레임 속도가 일치하는지 확인합니다(모든 소수 또는 모든 정수 프레임 속도).
- 인코딩 세션 간에 소수 프레임 속도와 정수 프레임 속도 간의 전환을 방지합니다.
- 인코딩 세션 간에 역방향으로 이동하지 않도록 출력 세그먼트 시퀀스 번호를 구성합니다.
- 모든 리전에서 동일한 인코더 출력 이름을 사용합니다.

MQAR용 MediaLive 구성에 대한 자세한 내용은 MediaLive 사용 설명서의 [MQCS 작업을](#) 참조하세요.

3단계: MQAR용 MediaPackage 구성

다음 구성을 사용하여 MediaPackage 채널 및 엔드포인트를 설정합니다.

MQAR용 MediaPackage를 구성하려면

1. 각 AWS 리전에서 동일한 채널 및 엔드포인트 구성을 생성합니다.
2. CMAF를 채널 입력 유형으로 사용합니다.
3. 기본 MediaPackage 오리진의 경우 다음 설정으로 엔드포인트 강제 적용 오류 구성을 활성화합니다.
 - 오래된 다변량 재생 목록, 미디어 재생 목록 또는 MPDs
 - 불완전한 다변량 재생 목록, 미디어 재생 목록 또는 MPD
 - 슬레이트 입력
4. 백업 MediaPackage 오리진의 경우 장애 조치 성공 가능성을 극대화하기 위해 이러한 오류 구성을 활성화하지 마십시오.

MQAR용 MediaPackage 구성에 대한 자세한 내용은 MediaPackage 사용 설명서의 [를 사용하여 미디어 품질 점수 활용 AWS Elemental MediaPackage](#)을 참조하세요.

4단계: MQAR용 CloudFront 구성

CloudFront 구성에서 MQAR을 활성화하고 오리진 중에서 선택하는 방법을 정의합니다. 미디어 품질 점수 옵션이 활성화된 오리진 그룹을 생성합니다.

MQAR용 CloudFront를 구성하려면

1. 오리진이 MediaTailor 엔드포인트를 가리키는 CloudFront 배포를 생성합니다.
2. 이러한 오리진을 포함하는 오리진 그룹을 생성합니다.
3. 오리진 그룹 설정에서 미디어 품질 점수 옵션을 활성화합니다.
4. 404 찾을 수 없음 응답 코드를 포함하도록 장애 조치 기준을 구성합니다. 선택적으로 다른 4xx/5xx 코드를 포함할 수 있습니다.
5. 각 채널의 경로 패턴에 대해 별도의 캐시 동작을 생성합니다. 이렇게 하면 동일한 오리진 그룹이 여러 채널을 제공할 때 점수가 섞이지 않습니다.

Note

배포의 캐시 동작과 연결된 오리진 방향 트리거(오리진 요청 및 오리진 응답)에서 Lambda@Edge 함수를 사용하는 경우 MQAR을 사용할 수 없습니다.

MQAR용 CloudFront 구성에 대한 자세한 내용은 CloudFront 개발자 안내서의 [미디어 품질 인식 복원력](#)을 참조하세요.

5단계: MQAR용 MediaTailor 구성

MQAR 설정을 완료하려면 다중 리전 아키텍처와 함께 작동하도록 각 리전에서 MediaTailor를 구성합니다. 이렇게 하면 CloudFront에서 어떤 오리진을 선택하든 일관된 광고 삽입이 보장됩니다.

MQAR용 MediaTailor를 구성하려면

1. 각 리전에서 해당 MediaPackage 엔드포인트를 가리키는 동일한 MediaTailor 구성을 생성합니다. MediaPackage
2. CloudFront 배포 도메인을 사용하도록 CDN 콘텐츠 세그먼트 접두사를 구성합니다.
3. 광고 결정 서버 구성이 리전 간에 동일한지 확인합니다.

이 설정을 통해 CloudFront가 품질 점수를 기반으로 선택하는 오리진에 관계없이 MediaTailor는 지속적으로 광고를 개인화할 수 있습니다.

6단계: MQAR 구성 테스트

MQAR을 설정한 후 구성을 테스트하여 예상대로 작동하는지 확인합니다.

MQAR 구성을 테스트하려면

1. CloudFront 배포를 통해 콘텐츠를 요청합니다.
2. 실시간 로그를 모니터링하여 CloudFront가 품질 점수를 기반으로 오리진을 선택하고 있는지 확인합니다.
3. 기본 AWS 리전의 장애를 시뮬레이션하여 장애 조치 동작을 테스트합니다.
4. 장애 조치 중에 광고 삽입이 계속 올바르게 작동하는지 확인합니다.

다음 단계

MQAR을 구현한 후 다음 단계를 고려하세요.

- MQAR 구성에 대한 모니터링 및 문제 해결 설정(참조 [CloudFront 및 MediaTailor 통합 모니터링 및 문제 해결](#))
- 를 사용하여 자동 배포 구현 AWS CloudFormation (참조 [를 사용하여 MediaTailor 및 CDN 자동화 CloudFormation](#))

CloudFront 및 MediaTailor 통합 모니터링 및 문제 해결

AWS Elemental MediaTailor Amazon CloudFront와 통합하려면 최적의 성능을 유지하기 위한 지속적인 모니터링 및 문제 해결이 필요합니다. CloudFront와 MediaTailor 통합을 구현한 후 성능을 모니터링하고 문제를 해결할 준비를 합니다. 콘텐츠 전송 네트워크(CDN)는 배포의 성능을 이해하고 잠재적 문제를 식별하는 데 도움이 되는 도구를 제공합니다.

통합에 대한 모니터링 설정

CloudFront 및 MediaTailor 통합의 성능을 추적하려면 다음 모니터링 전략을 구현합니다.

CloudFront 및 MediaTailor 통합에 대한 모니터링을 설정하려면

1. CloudFront 표준 로깅 활성화:

- a. CloudFront 콘솔에서 배포를 선택합니다.
 - b. 로그 탭을 선택합니다.
 - c. 표준 로그에서 편집을 선택합니다.
 - d. 켜기를 선택하고 로그 스토리지용 Amazon S3 버킷을 구성합니다.
 - e. 변경 사항 저장을 선택합니다.
2. CloudFront 실시간 로그 설정:
- a. CloudFront 콘솔에서 배포를 선택합니다.
 - b. 로그 탭을 선택합니다.
 - c. 실시간 로그에서 편집을 선택합니다.
 - d. 켜기를 선택하고 Amazon Kinesis Data Streams 또는 Amazon Data Firehose 전송 스트림을 구성합니다.
 - e. 로그 구성에 다음 필드를 포함합니다.
 - time-to-first-byte - 응답 시간
 - sc-status - HTTP 상태 코드
 - c-ip - 최종 사용자 IP 주소
 - cs-uri-stem - 요청 URI 경로
 - cs-user-agent - 사용자 에이전트
 - x-edge-result-type - 결과 유형(Hit, Miss, Error)
 - f. 변경 사항 저장을 선택합니다.
3. CloudWatch 대시보드 생성:
- a. CloudWatch 콘솔에서 대시보드를 선택합니다.
 - b. 대시보드 생성을 선택합니다.
 - c. 이러한 CloudFront 지표에 대한 위젯을 추가합니다.
 - Requests
 - BytesDownloaded
 - 4xxErrorRate
 - 5xxErrorRate
 - TotalErrorRate

- CacheHitRate

4. CloudWatch 경보 설정:

- a. CloudWatch 콘솔에서 경보를 선택합니다.
- b. 경보 생성을 선택하세요.
- c. 다음 조건에 대한 경보를 생성합니다.
 - 5xxErrorRate > 5분 동안 1%
 - 5분 동안 4xxErrorRate가 5%보다 큼
 - CacheHitRate 30분 동안 80% 미만

MQAR 성능 모니터링

MQAR을 사용할 때 모니터링을 통해 CloudFront에서 오리진을 선택하는 방법과 품질 점수가 기대치를 충족하는지 여부를 이해할 수 있습니다. 실시간 로그에는 이러한 결정이 발생하는 즉시 표시됩니다.

MQAR 성능을 모니터링하려면

1. CloudFront 배포에 대한 실시간 로그를 활성화합니다.
2. 로그 구성에 다음 필드를 포함합니다.
 - r-host - 선택한 오리진의 호스트 이름
 - sr-reason - 오리진 선택 이유
 - x-edge-mqcs - 미디어 품질 신뢰도 점수
3. Amazon Kinesis Data Streams 또는 Amazon Data Firehose에서 로그 대상을 설정합니다.
4. 이러한 지표를 기반으로 대시보드 또는 알림을 생성하여 품질 점수 및 오리진 선택 결정을 모니터링합니다.

Example MQAR 모니터링을 위한 샘플 CloudWatch 대시보드

다음 지표를 사용하여 CloudWatch 대시보드를 생성합니다.

- 리전별 오리진 선택 수
- 시간 경과에 따른 평균 품질 점수
- 장애 조치 이벤트
- 오리진별 4xx 및 5xx 오류 발생률

실시간 로그 설정에 대한 자세한 내용은 CloudFront 개발자 안내서의 [실시간 로그](#)를 참조하세요.

일반적인 문제 해결

신중하게 계획하더라도 CloudFront 및 MediaTailor 통합에 문제가 발생할 수 있습니다. 일반적인 문제와 솔루션을 이해하면 문제를 신속하게 해결하고 최종 사용자에게 미치는 영향을 최소화할 수 있습니다.

CloudFront 및 MediaTailor 통합에 문제가 발생하면 다음과 같은 일반적인 문제와 해결 방법을 확인하세요.

매니페스트 캐싱 문제

증상: 오래된 매니페스트 또는 광고가 업데이트되지 않음

해결 방법: 다변량 재생 목록, 미디어 재생 목록 및 MPD 경로에 캐시 CachingDisabled 정책을 사용하고 있는지 확인합니다. 구성이 쿼리 파라미터를 올바르게 전달하는지 확인합니다.

CORS 오류

증상: 콘텐츠에 액세스할 때 브라우저 콘솔에 CORS 오류가 표시됨

해결 방법: 적절한 CORS 헤더로 응답 헤더 정책을 구성하고 캐시 동작과 연결합니다.

MQAR이 작동하지 않음

증상: 오리진 선택 시 품질 점수 무시

해결 방법: 오리진 그룹 설정에서 미디어 품질 점수 옵션을 활성화했는지 확인합니다. Lambda@Edge 오리진 방향 트리거를 사용하지 않는지 확인합니다.

광고 삽입 실패

증상: 광고가 제대로 삽입되지 않음

해결 방법: MediaTailor 구성이 콘텐츠 세그먼트 접두사에 대한 CloudFront 배포를 가리키는지 확인합니다. 설정이 필요한 모든 헤더를 전달하는지 확인합니다.

보다 복잡한 문제의 경우 다음과 같은 문제 해결 접근 방식을 사용할 수 있습니다.

1. CloudFront 배포 로그에서 오류 패턴 확인
2. 브라우저 개발자 도구를 사용하여 네트워크 요청 검사
3. MediaTailor의 매니페스트 콘텐츠를 CloudFront를 통해 제공되는 콘텐츠와 직접 비교
4. 세부 로깅을 지원하는 간단한 플레이어로 테스트

문제 해결에 대한 자세한 내용은 MediaTailor 사용 설명서의 [문제 해결](#) 섹션을 참조하세요.

워크플로 문제 해결

이 체계적인 접근 방식을 따라 CloudFront 및 MediaTailor 통합 문제를 해결합니다.

CloudFront 및 MediaTailor 통합 문제를 해결하려면

1. 특정 문제를 식별합니다.
 - a. 문제가 모든 최종 사용자 또는 일부에만 영향을 미치는지 확인
 - b. 영향을 받는 콘텐츠 유형 식별(매니페스트, 세그먼트 또는 둘 다)
 - c. 오류 메시지 또는 증상 기록
2. CloudFront 로그 확인:
 - a. 오류 상태 코드(4xx 또는 5xx) 찾기
 - b. 캐시 적중/누락 패턴 확인
 - c. 요청이 올바른 오리진으로 라우팅되고 있는지 확인
3. 구성 확인:
 - a. 캐시 동작에서 올바른 경로 패턴 확인
 - b. 캐시 정책이 올바르게 적용되었는지 확인
 - c. 오리진 요청 정책이 필요한 헤더를 전달하는지 확인
4. 직접 액세스 테스트:
 - a. MediaTailor에서 직접 콘텐츠에 액세스해 보세요(CloudFront 우회).
 - b. 직접 액세스와 CloudFront 액세스 간의 응답 비교
5. 솔루션 구현:
 - a. 조사 결과에 따라 적절한 수정 사항 적용
 - b. 테스트하여 문제가 해결되었는지 확인
 - c. 향후 참조를 위해 문제 및 솔루션 문서화

다음 단계

CloudFront 및 MediaTailor 통합에 대한 모니터링 및 문제 해결을 설정한 후 다음 단계를 고려하세요.

- 를 사용하여 자동 배포 구현 AWS CloudFormation (참조 [를 사용하여 MediaTailor 및 CDN 자동화 CloudFormation](#))
- 일반적인 운영 시나리오 및 문제 해결 절차를 위한 실행서 생성
- 일반적인 문제에 대한 자동 문제 해결 설정

MediaTailor 광고 전송을 위한 타사 CDN 설정

Akamai 및 Fastly와 같은 타사 CDN 대역폭 비용을 줄이면서 AWS Elemental MediaTailor 광고 전송의 성능과 확장성을 크게 개선할 수 있습니다. 그러나 개인 맞춤형 광고를 위한 CDN 구성에는 표준 비디오 전송과 다른 특정 설정이 필요합니다.

타사 CDN [MediaTailor 매니페스트 쿼리 파라미터](#). 동적 변수를 사용하는 고급 라우팅 구성은 섹션을 참조하세요 [ADS 요청에 대한 MediaTailor 동적 광고 변수](#).

이 가이드에서는 MediaTailor와 최적으로 작동하도록 타사 CDN을 설정하는 전체 프로세스를 안내합니다. 다음 두 가지 필수 동작을 구성하는 방법을 알아봅니다.

- 매니페스트 우회: 매니페스트 캐싱을 방지하여 각 최종 사용자가 개인화된 광고 삽입을 수신하도록 보장합니다.
- 세그먼트 캐싱: 비디오 콘텐츠를 효율적으로 캐싱하여 성능을 극대화하고 비용을 절감합니다.

구성 프로세스에는 일반적으로 30~60분이 소요되며 공급자 선택, 설정, 확인 및 최적화가 포함됩니다. 완료되면 최적의 시청자 경험을 유지하면서 맞춤형 광고를 효율적으로 제공하는 CDN 구성이 제공됩니다.

Note

이 가이드는 Akamai 및 Fastly 구성에 중점을 둡니다. CloudFront 설정 지침은 섹션을 참조하세요 [CloudFront 통합](#).

사전 조건

MediaTailor를 사용하여 타사 CDN을 설정하기 전에 다음이 있는지 확인합니다.

- 콘텐츠 오리진 및 광고 결정 서버를 포함하는 활성 MediaTailor 구성

- CDN의 구성 인터페이스에 대한 액세스
- 콘텐츠에 사용되는 파일 확장명 목록(.m3u8, .mpd, .ts 등)
- 참조용 CDN 공급자 설명서

타사 CDN 대신 CloudFront 설정 지침은 섹션을 참조하세요 [CloudFront 통합](#). CDN

용어

CDN 구성 요구 사항을 이해하려면 다음 매니페스트 유형을 알아야 합니다.

- HLS 매니페스트:
 - 다변량 재생 목록: 미디어 재생 목록 링크가 포함된 최상위 매니페스트
 - 미디어 재생 목록: 콘텐츠 세그먼트 링크가 있는 2단계 매니페스트
- DASH 매니페스트:
 - MPD(미디어 프레젠테이션 설명): DASH 매니페스트의 표준 용어

이 가이드는 공통 구성 요구 사항을 논의할 때 모든 매니페스트 파일(다변량 재생 목록, 미디어 재생 목록 및 MPDs)을 총칭하여 매니페스트라고 합니다.

모든 공급자에 적용되는 일반적인 CDN 구성 원칙은 섹션을 참조하세요 [MediaTailor와 CDN 통합 설정](#).

CDN 최적화 지침은 섹션을 참조하세요 [CDN 및 MediaTailor 통합을 위한 성능 최적화 가이드](#).

CDN 캐싱 규칙 구성

CDN 캐싱 구성은 콘텐츠가 최종 사용자에게 도달하는 방식을 결정하기 때문에 MediaTailor 광고 전송에 매우 중요합니다. 적절한 구성을 통해 매니페스트가 각 최종 사용자에게 맞게 개인화된 상태로 유지되는 동시에 세그먼트를 효율적으로 캐싱하여 오리진 로드를 줄이고 성능을 개선할 수 있습니다.

이 구성은 일반적으로 CDN 공급자당 15~30분이 소요되며 두 가지 고유한 동작이 필요합니다.

- 매니페스트 처리: 캐싱을 방지하여 각 시청자가 개인화된 광고 삽입을 받을 수 있도록 합니다.
- 세그먼트 캐싱: 비디오 콘텐츠의 캐시 효율성을 극대화하여 전송 성능을 개선합니다.

다음 단계에 따라 최적의 광고 전송을 위해 CDN의 캐싱 규칙을 구성합니다.

특정 지침을 보려면 다음 탭에서 CDN 공급자를 선택합니다.

Akamai

Akamai 속성에서 다음 두 가지 동작을 구성합니다.

- 캐싱 방지를 위한 매니페스트 처리
- 최적의 성능을 위한 세그먼트 캐싱

매니페스트 전송 구성

각 최종 사용자가 개인화된 광고를 수신하도록 캐싱 매니페스트를 방지하도록 Akamai CDN을 구성합니다.

매니페스트 파일에는 MediaTailor가 각 뷰어에 대해 생성하는 개인화된 광고 삽입 지점이 포함됩니다. 이러한 파일을 캐싱하면 모든 시청자가 동일한 광고를 보게 되어 맞춤형 광고의 목적을 무너뜨릴 수 있습니다.

매니페스트 요청(.m3u8, .mpd 또는 .smil로 끝나는 파일)에 대해 다음 단계를 따릅니다.

1. 매니페스트 파일 확장명(.m3u8, .mpd, .smil)과 일치하는 동작 생성
2. 캐싱 옵션을 스토어 없음으로 설정
3. 모든 쿼리 파라미터를 포함하도록 캐시 키 구성
4. 적절한 오리진 라우팅을 위해 전달 호스트 헤더 활성화
5. 모든 헤더에 대한 헤더 전달을 구성합니다. 최소 요구 사항은 섹션을 참조하세요 [MediaTailor CDN 통합에 필요한 헤더](#).

세그먼트 전송 구성

비디오 세그먼트를 캐싱하도록 Akamai CDN을 구성하여 CDN 효율성을 극대화하고 오리진 로드를 줄입니다.

비디오 세그먼트는 개인화된 정보가 포함되어 있지 않기 때문에 안전하게 캐싱할 수 있는 실제 콘텐츠 파일입니다. 적절한 세그먼트 캐싱은 대역폭 비용을 줄이고 최종 사용자의 재생 성능을 개선합니다.

세그먼트 요청(.ts, .mp4, .m4s 등으로 끝나는 파일)에 대해 다음 단계를 따릅니다.

1. 세그먼트 파일 확장명(.ts, .mp4, .m4s)과 일치하는 동작 생성
2. 오리진 캐시 제어를 예로 설정

3. 오리진 헤더가 누락된 경우에 대한 기본 TTL(time-to-live) 설정을 구성합니다.

- 기본 TTL: 86400초(24시간)
- 최대 TTL: 604800초(7일)

Note

이러한 동작을 구성한 후 Akamai Control Center에서 속성 변경 사항을 활성화합니다. 변경 사항은 활성화 후 적용됩니다.

Fastly

Fastly 서비스에서 다음 두 가지 구성을 생성합니다.

- 캐싱 방지를 위한 매니페스트 처리
- 최적의 성능을 위한 세그먼트 캐싱

매니페스트 전송 구성

매니페스트 파일의 캐싱을 우회하도록 Fastly CDN을 구성하여 각 시청자가 개인화된 광고 콘텐츠를 수신하도록 합니다.

적절한 광고 개인화를 보장하려면 매니페스트 파일이 각 요청에 대해 MediaTailor에 도달해야 합니다. 이러한 파일의 캐시를 우회하면 각 최종 사용자의 고유한 대상 지정 파라미터가 올바르게 처리됩니다.

매니페스트 요청에 대해 다음 단계를 따릅니다.

1. 매니페스트 경로를 식별하기 위한 요청 조건 생성
2. 캐시 조건을 이러한 요청에 대해 캐시하지 않음으로 설정
3. 모든 쿼리 파라미터를 포함하도록 전달 설정 구성
4. 헤더 전달 구성 User-Agent에 추가

세그먼트 전송 구성

전송 성능을 개선하고 오리진 트래픽을 줄이기 위해 비디오 세그먼트를 캐싱하도록 Fastly CDN을 구성합니다.

세그먼트 캐싱은 비용 효율적인 전송과 최적의 최종 사용자 경험에 필수적입니다. 이러한 파일은 모든 최종 사용자에게 동일하며 CDN 캐싱의 이점을 크게 누릴 수 있습니다.

세그먼트 요청에 대해 다음 단계를 따릅니다.

1. 세그먼트 경로를 식별하기 위한 요청 조건 생성
2. 캐시 설정을 오리진 캐시 헤더 명어로 설정
3. 오리진 헤더가 누락된 경우 기본 TTL(time-to-live)을 86,400초(24시간)로 구성

Note

이러한 변경을 수행한 후 Fastly 서비스의 새 버전을 활성화합니다.
구성은 활성화 후 적용됩니다.

CDN 구성 확인

CDN 구성을 통해 프로덕션 트래픽을 전달하기 전에 CDN 구성이 올바르게 작동하는지 확인합니다. 이러한 테스트는 광고 개인 맞춤이 제대로 작동하고 캐싱이 예상 성능 이점을 제공하는지 확인합니다.

전체 확인은 일반적으로 10~15분이 소요되며 가능하면 여러 지리적 위치에서 수행해야 합니다.

CDN을 설정한 후 다음 검사를 수행하여 올바르게 작동하는지 확인합니다.

1. 매니페스트 개인 맞춤 테스트:
 - a. 다른 광고 파라미터를 사용하여 동일한 콘텐츠 URL 요청
 - b. 각 요청이 서로 다른 광고 삽입을 반환하는지 확인
2. 세그먼트 캐싱 테스트:
 - a. CDN 지표에서 세그먼트 캐시 적중률 확인(90%보다 커야 함)
 - b. 오리진 트래픽을 모니터링하여 직접 전송보다 낮은지 확인
3. 재생 성능 테스트:
 - a. 다양한 위치에서 CDN을 통해 콘텐츠 재생
 - b. 버퍼링 문제 없이 원활한 재생 확인

포괄적인 테스트 방법 및 고급 검증 절차는 섹션을 참조하세요 [CDN 및 MediaTailor 통합 테스트 및 검증](#).

CDN 성능 최적화

기본 구성을 확인한 후 이러한 최적화를 구현하여 성능을 극대화하고 비용을 최소화합니다.

주요 성능 지표 모니터링

이러한 지표를 추적하여 최적의 성능을 보장합니다.

캐시 적중률

대상: 비디오 세그먼트의 경우 90% 초과

영향: 비율이 높을수록 오리진 로드가 줄어들고 최종 사용자 경험이 개선됩니다.

모니터링: CDN 공급자의 분석 대시보드를 매일 확인

오리진 응답 시간

대상: 매니페스트 요청의 경우 200ms 미만

영향: 더 빠른 매니페스트 전송으로 최종 사용자의 시작 시간 단축

모니터링: 500ms를 초과하는 응답 시간에 대한 알림 설정

오류 발생률

대상: 모든 요청에 대해 0.1% 미만

영향: 높은 오류율은 구성 문제 또는 오리진 문제를 나타냅니다.

모니터링: 1%를 초과하는 오류 발생률에 대한 알림 설정

캐싱 동작 미세 조정

콘텐츠 특성 및 뷰어 패턴에 따라 이러한 설정을 조정합니다.

세그먼트 TTL 최적화

라이브 콘텐츠: 더 짧은 TTL(1~4시간)을 사용하여 시기 적절한 업데이트 보장

VOD 콘텐츠: 더 긴 TTL(24~48시간)을 사용하여 캐시 효율성 극대화

광고 세그먼트: 자주 업데이트되는 광고 콘텐츠의 경우 더 짧은 TTL(30분~2시간) 고려

모든 MediaTailor 워크플로에 대한 포괄적인 TTL 권장 사항 및 캐싱 전략은 섹션을 참조하세요 [CDN 및 MediaTailor 통합을 위한 캐싱 최적화](#).

지리적 최적화

다중 리전 오리진: 최종 사용자 위치를 기반으로 오리진 선택 구성

엣지 로케이션: 시청자 집중도가 높은 리전에서 추가 엣지 로케이션 활성화

장애 조치: 고가용성을 위해 백업 오리진 구성

비용 최적화

성능을 유지하면서 CDN 비용을 줄이려면 다음 전략을 구현합니다.

- 압축: 매니페스트 파일에 대해 gzip 압축을 활성화하여 대역폭 사용량을 줄입니다.
- 제거 전략: 전체 캐시 지우기 대신 선택적 캐시 제거 구현
- 트래픽 분석: 매월 트래픽 패턴을 검토하여 최적화 기회를 식별합니다.
- 티어 선택: 성능 요구 사항에 따라 적절한 CDN 서비스 티어 사용

타사 CDN 문제 해결

CDN 구성 문제는 일반적으로 광고 개인화 문제 또는 성능 저하로 나타납니다. 이 체계적인 접근 방식을 사용하여 MediaTailor 광고 전송에 영향을 미치는 가장 일반적인 문제를 식별하고 해결합니다.

대부분의 문제 해결은 특정 증상을 확인하고 해당 솔루션을 적용하여 15~30분 이내에 완료할 수 있습니다.

최종 사용자에게 광고 전송 또는 재생 품질 문제가 발생하는 경우 이 가이드를 사용하여 일반적인 CDN 구성 문제를 식별하고 해결합니다.

Akamai: 캐시 매니페스트

증상: 다양한 대상 지정 파라미터를 구성하더라도 최종 사용자는 동일한 광고를 볼 수 있습니다.

해결 방법: 매니페스트 경로에 No Store 캐싱 옵션을 적용했는지 확인합니다.

또한 캐시 키에 쿼리 파라미터를 포함했는지 확인합니다.

Fastly: 잘못된 캐시 키

증상: 최종 사용자는 일관성 없는 광고 개인 맞춤을 경험합니다.

최종 사용자는 다른 최종 사용자를 위한 광고를 볼 수도 있습니다.

해결 방법: 캐시 키에 모든 쿼리 파라미터를 포함하도록 전달 설정을 구성했는지 확인합니다.

일반: 높은 오리진 트래픽

증상: 오리진 서버의 트래픽이 예기치 않게 높음

해결 방법: 세그먼트 캐싱 설정 및 TTL(time-to-live) 값을 확인합니다.

CDN 지표에서 캐시 적중률을 확인합니다.

일반: 재생 오류

증상: 최종 사용자에게 버퍼링 또는 재생 실패 발생

해결 방법: CDN 라우팅 규칙 및 오리진 상태를 확인합니다.

필요한 모든 헤더가 올바르게 전달되고 있는지 확인합니다.

Note

이러한 해결 방법으로 문제가 해결되지 않는 경우 CDN 공급자의 설명서를 확인하세요. 추가 문제 해결 단계는 지원 팀에 문의할 수도 있습니다.

일반적인 CDN 문제 해결 지침은 섹션을 참조하세요 [MediaTailor 및 CDN 통합 관련 문제 해결](#).

CDN 및 MediaTailor 통합을 위한 성능 최적화 가이드

AWS Elemental MediaTailor 체계적인 콘텐츠 전송 네트워크(CDN) 최적화를 통해 성능을 극대화할 수 있습니다. 서버 측 광고 삽입(SSAI), 채널 어셈블리 또는 결합된 워크플로를 구현하든 관계없이 최적화 원칙과 성능 목표는 일관되게 유지됩니다. 이 가이드는 모든 MediaTailor 구현에 적용되는 포괄적인 최적화 기법과 벤치마크를 제공합니다.

동적 변수 및 구성 별칭을 사용한 고급 라우팅 최적화는 섹션을 참조하세요 [ADS 요청에 대한 MediaTailor 동적 광고 변수](#). 쿼리 파라미터 최적화 전략은 섹션을 참조하세요 [MediaTailor 매니페스트 쿼리 파라미터](#).

최적화 워크플로 개요:

1. 캐싱 구성 - 적절한 TTL 값 및 캐시 동작 설정
2. 라우팅 최적화 - 요청 라우팅 및 오리진 정책 구성

3. 성능 측정 - 설정된 벤치마크를 기준으로 추적
4. 고급 기법 적용 - 추가 최적화 기능 구현

주제

- [CDN 및 MediaTailor 통합을 위한 캐싱 최적화](#)
- [CDN 및 MediaTailor 통합을 위한 라우팅 최적화 요청](#)
- [CDN 및 MediaTailor 통합의 성능 벤치마크](#)
- [CDN 및 MediaTailor 통합을 위한 고급 최적화 기술](#)

CDN 및 MediaTailor 통합을 위한 캐싱 최적화

AWS Elemental MediaTailor 캐싱 요구 사항은 워크플로 유형 및 콘텐츠 형식에 따라 다릅니다. 최적의 성능, 비용 효율성 및 최종 사용자 경험을 위해서는 적절한 캐싱 구성이 중요합니다. 다음 섹션에서는 다양한 MediaTailor 구현에 대한 자세한 캐싱 지침을 제공합니다.

서버 측 광고 삽입(SSAI) 캐싱

서버 측 광고 삽입 워크플로의 경우 개인화된 매니페스트는 캐싱할 수 없지만 콘텐츠 및 광고 세그먼트는 적극적으로 캐싱해야 합니다.

SSAI CDN 캐싱 설정

콘텐츠 유형	TTL	경로 패턴	캐시 키 요소
다변량 재생 목록	0초	/v1/마스터/*	URL 경로 + 모든 쿼리 파라미터
미디어 재생 목록	0초	/v1/매니페스트/*	URL 경로 + 모든 쿼리 파라미터
대시 MPDs	0초	/v1/dash/*	URL 경로 + 모든 쿼리 파라미터
콘텐츠 세그먼트	24시간 이상	콘텐츠별 경로	URL 경로만
광고 세그먼트	24시간 이상	/v1/segment/*	URL 경로만

- 최종 사용자가 최신 광고 콘텐츠를 수신up-to-date 0초로 설정합니다.
- 콘텐츠 및 광고 세그먼트에 대해 더 긴 TTL 값을 구성하여 캐시 효율성 극대화
- 대상 광고를 지원하는 경우 캐시 키에 개인화 파라미터를 포함하는 캐시 동작을 설정합니다.
- CDN 수준에서 요청 축소를 구현하여 동시 요청을 효율적으로 처리

권장 TTL 구성 설정

최적의 SSAI 성능을 위해 다음과 같은 특정 TTL 설정으로 CDN 캐시 정책을 구성합니다.

SSAI TTL 구성 설정

콘텐츠 유형	TTL 설정	권장 값
광고 세그먼트	최소 TTL	1초
광고 세그먼트	최대 TTL	86400초(24시간)
광고 세그먼트	기본 TTL	86400초(24시간)
콘텐츠 세그먼트	최소 TTL	1초
콘텐츠 세그먼트	최대 TTL	86400초(24시간)
콘텐츠 세그먼트	기본 TTL	86400초(24시간)

이러한 설정은 다음을 보장합니다.

- 최소 TTL 1초: 과도한 오리진 요청을 방지하면서 필요한 경우 빠른 캐시 무효화 허용
- 최대 TTL 24시간: 캐시 효율성과 콘텐츠 최신성 요구 사항의 균형을 맞춥니다.
- 기본 TTL 24시간: 명시적 캐시 제어 헤더가 없는 세그먼트에 대해 최적의 캐싱 제공

서버 가이드 광고 삽입(SGAI) 캐싱

서버 가이드 광고 삽입(SGAI)을 사용하면 예측 가능한 URL 패턴을 사용하는 캐시 가능한 미디어 매니페스트를 통해 효율적인 CDN 캐싱이 가능합니다. 이 섹션에서는 최적의 SGAI 성능을 위한 CDN별 구성 요구 사항에 중점을 둡니다.

SGAI에 대한 CDN 캐싱 구성

다음과 같은 SGAIB별 캐싱 동작으로 CDN을 구성합니다.

SGAI CDN 캐싱 설정

콘텐츠 유형	TTL	경로 패턴	캐시 키 요소
SGAI 다변량 재생 목록(캐싱 금지)	0초(캐싱 금지)	/v1/마스터/*	URL 경로 + 선택한 쿼리 파라미터
SGAI 미디어 재생 목록	1~4초(세그먼트 길이의 절반)	/v1/i-media/*	URL 경로 + 선택한 쿼리 파라미터
자산 목록 응답(캐싱 금지)	0초(캐싱 금지)	/v1/간질/*	URL 경로 + 모든 쿼리 파라미터
광고 세그먼트	24시간 이상	광고별 경로	URL 경로만

캐시 동작 구성

SGAI 콘텐츠에 대한 전용 캐시 동작을 설정합니다.

- SGAIB 매니페스트 동작 - 1~4초 TTL을 사용하여 /v1/i-media/* 경로에 대한 캐시 동작 생성
- 자산 목록 동작 - 0초 TTL이 있는 /v1/interstitials/* 경로에 대한 캐시 동작 생성
- 쿼리 파라미터 처리 - 캐시 효율성을 극대화하기 위해 캐시 키에 필수 대상 지정 파라미터만 포함
- 오리진 요청 헤더 - 캐시 가능성을 유지하면서 광고 타겟팅에 필요한 헤더 전달

채널 어셈블리 캐싱

채널 어셈블리 워크플로의 경우 매니페스트를 단기간 캐싱할 수 있지만 세그먼트는 적극적으로 캐싱해야 합니다.

채널 어셈블리 CDN 캐싱 설정

콘텐츠 유형	VOD TTL	라이브 TTL	경로 패턴	캐시 키 요소
다변량 재생 목록	5~30분	5~10초	채널별 경로	URL 경로 + 모든 쿼리 파라미터
미디어 재생 목록	5~30분	2~5초	채널별 경로	URL 경로 + 모든 쿼리 파라미터
대시 MPDs	5~30분	5~10초	채널별 경로	URL 경로 + 모든 쿼리 파라미터
콘텐츠 세그먼트	24시간 이상	5~15분	콘텐츠별 경로	URL 경로만
광고 세그먼트	24시간 이상	24시간 이상	광고별 경로	URL 경로만

- 최종 사용자가 up-to-date 프로그래밍을 받을 수 있도록 매니페스트에 대한 짧은 TTL 값을 설정합니다.
- 콘텐츠 세그먼트에 대해 더 긴 TTL 값을 구성하여 캐시 효율성 극대화
- 시간 이동 보기를 지원하는 경우 캐시 키에 시간 이동 파라미터를 포함하는 캐시 동작 설정
- 캐시 키에 쿼리 파라미터를 포함하여 시간 이동 보기 요청을 올바르게 처리

자세한 TTL 구성 설정 및 모범 사례는 섹션을 참조하세요 [the section called “캐싱 최적화”](#).

SSAI 및 채널 어셈블리 캐싱 결합

채널 어셈블리와 SSAI를 모두 구현할 때는 충돌을 방지하고 성능을 최적화하기 위해 캐싱 전략이 두 서비스 모두에 일관되게 적용되어야 합니다.

결합된 워크플로 캐싱 설정 비교

콘텐츠 유형	채널 어셈블리	SSAI	통합 권장 사항
VOD 매니페스트	5~30분	0초	(별도 구성 사용)
라이브 매니페스트	2~10초	0초	(별도 구성 사용)
SGAI VOD 매니페스트	5~30분	5~30분	5~30분(캐싱 가능한 매니페스트)
SGAI Live 매니페스트	2~4초	2~4초	2~4초(캐싱 가능한 매니페스트)
콘텐츠 세그먼트	24시간 이상	24시간 이상	24시간 이상(일관성)
광고 세그먼트	24시간 이상	24시간 이상	24시간 이상(일관성)

이 구성은 캐시 효율성을 극대화하는 동시에 최종 사용자가 맞춤형 광고 삽입을 위해 up-to-date 매니페스트를 수신하도록 합니다.

CDN 및 MediaTailor 통합을 위한 라우팅 최적화 요청

모든 AWS Elemental MediaTailor CDN 통합에 대해 다음과 같은 라우팅 최적화를 구현합니다.

- 매니페스트 및 세그먼트 요청에 대해 별도의 캐시 동작 생성
- 헤더 전달을 제어하도록 오리진 요청 정책 구성
- 적절한 오류 처리 및 장애 조치 메커니즘 설정
- CDN에서 사용 가능한 경우 오리진 실드를 구현하여 오리진 로드를 줄입니다.
- CDN 수준에서 요청 축소를 구현하여 동시 요청을 효율적으로 처리

CDN 및 MediaTailor 통합의 성능 벤치마크

AWS Elemental MediaTailor CDN 통합을 최적화할 때는 이러한 성능 벤치마크를 목표로 하세요. 이러한 대상은 SSAI, 채널 어셈블리 및 결합된 워크플로를 포함한 모든 MediaTailor 구현에 적용됩니다.

캐시 적중률 대상

콘텐츠 세그먼트: 캐시 적중률 95% 초과

광고 세그먼트: 캐시 적중률 90% 초과

매니페스트: SSAI에는 적용되지 않음(개인화된 광고 삽입을 위해 캐시해서는 안 됨), 채널 어셈블리의 경우 85% 이상

지연 시간 벤치마크

매니페스트 요청 지연 시간: 100ms 미만(P95)

콘텐츠 세그먼트 전송: 50ms 미만(P95)

광고 세그먼트 전송: 75ms 미만(P95)

End-to-end 시작 시간: 2초 미만

오리진 로드 지표

최종 사용자당 오리진 요청: 최종 사용자당 분당 요청 0.1개 미만

뷰어당 오리진 대역폭: 총 뷰어 대역폭의 5% 미만

오류율 대상

매니페스트 오류: 0.1% 미만

세그먼트 오류: 0.01% 미만

플레이어 보고 재버퍼링: 1% 미만

확장성 벤치마크

성능 저하 없이 피크 이벤트 중 정상 트래픽의 10배 지원

채널당 초당 1,000개 이상의 요청을 처리할 수 있는 기능

Amazon CloudWatch 지표를 사용하여 이러한 성능 지표를 추적합니다. 자세한 모니터링 지침은 섹션을 참조하세요 [모니터링 도구 설정](#).

CDN 및 MediaTailor 통합을 위한 고급 최적화 기술

기본 캐싱 및 라우팅 최적화를 구현한 후 다음과 같은 고급 기술을 고려하여 성능을 더욱 개선하세요.

Origin Shield 구현

Origin Shield는 CDN 엣지 로케이션과 오리진 서버 사이에 캐싱 계층을 추가하여 오리진 로드를 줄이고 성능을 개선합니다.

- 트래픽이 많은 콘텐츠 및 라이브 스트리밍을 위해 Origin Shield 활성화
- MediaTailor 리전과 가까운 Origin Shield 위치를 선택합니다.
- Origin Shield 캐시 적중률 모니터링 및 필요에 따라 조정
- 글로벌 배포를 위해 여러 Origin Shield 위치 고려

콘텐츠 압축 최적화

콘텐츠 압축을 최적화하여 대역폭을 줄이고 전송 속도를 개선합니다.

- 매니페스트 파일에 대해 gzip 압축 활성화
- MediaTailor 매니페스트 압축에 대한 수락 인코딩 헤더 전달 구성
- 추가 대역폭 절감을 위해 지원되는 경우 Brotli 압축 사용
- 이미 압축된 비디오 세그먼트 압축 방지

리전 최적화 전략

리전별 전략을 통해 글로벌 대상의 성능을 최적화합니다.

- 글로벌 대상을 위해 여러 리전에 MediaTailor 구성 배포
- 지오 라우팅을 사용하여 최종 사용자를 가장 가까운 MediaTailor 리전으로 안내
- 고가용성을 위해 리전 장애 조치 구성
- 리전별 성능 지표를 별도로 모니터링

CDN 및 MediaTailor 통합을 위한 아키텍처 고려 사항

최적의 성능과 신뢰성을 보장하기 위해 아키텍처에 콘텐츠 전송 네트워크(CDN)를 올바르게 배치합니다. AWS Elemental MediaTailor. 권장 아키텍처는 MediaTailor와 오리진이 아닌 최종 사용자와 MediaTailor 사이에 CDN을 배치합니다.

자세한 아키텍처 다이어그램 및 워크플로 설명은 다음 주제를 참조하세요.

- [CDN을 사용한 광고 삽입](#) 광고 삽입 아키텍처 다이어그램 및 세부 워크플로용
- [CDN 아키텍처 이해](#) 채널 어셈블리 아키텍처 다이어그램 및 워크플로용

아키텍처에 CDN을 올바르게 배치합니다.

1. 플레이어와 MediaTailor 사이에 CDN을 배치합니다(MediaTailor와 오리진 사이에는 CDN을 배치하지 않음).

이 아키텍처를 사용하면 CDN이 광고 세그먼트와 콘텐츠 세그먼트를 캐싱할 수 있습니다. 동시에 MediaTailor는 각 뷰어에 대해 개인화된 매니페스트를 생성할 수 있습니다.

2. 다양한 요청 유형에 대해 별도의 캐시 동작을 생성합니다.

- 매니페스트 요청(캐싱 없음)
- 콘텐츠 세그먼트(긴 TTL)
- 광고 세그먼트(긴 TTL)

3. 적절한 오류 처리를 구성합니다.

- 서비스 중단 중에 반복되는 요청으로 오리진에 부담을 주지 않도록 음의 캐싱(오류 응답 임시 저장)을 설정합니다. 음수 캐싱은 CDN이 일시적으로 오류 응답(예: 404 또는 500 오류)을 저장하여 존재하지 않거나 일시적으로 사용할 수 없는 콘텐츠에 대한 반복 요청을 방지함을 의미합니다.
- 적절한 오류 응답 코드 및 재시도 동작 구성

4. 중간 캐싱(오리진 실드)을 구현합니다.

오리진 실드는 CDN 엣지 로케이션과 오리진 서버 간에 추가 캐싱 계층을 생성하는 기능입니다. 이렇게 하면 오리진 서버에 도달하는 중복 요청 수가 줄어듭니다.

- 엣지 로케이션과 오리진 간에 중간 캐싱 계층 구성
- 캐시 누락 시 오리진에 대한 중복 요청 수 감소
- CDN 인프라 전반의 캐시 적중률 개선

MediaTailor의 고급 CDN 기능

기본 콘텐츠 전송 네트워크(CDN) 구성을 구현한 후 이러한 고급 기능을 탐색하여 AWS Elemental MediaTailor 스트리밍 플랫폼의 성능과 신뢰성을 더욱 개선합니다.

미디어 품질 인식 라우팅(MQAR)

MQAR은 실시간 네트워크 성능 지표를 기반으로 최고 품질의 콘텐츠 소스를 자동으로 선택하는 Amazon CloudFront 기능입니다. MQAR은 고정 오리진 서버를 사용하는 대신 지연 시간 및 처리량

과 같은 요인을 기반으로 요청을 최적의 오리진으로 동적으로 라우팅합니다. 이를 통해 최종 사용자는 네트워크 변동 중에도 가능한 최고 품질의 스트림을 수신할 수 있습니다.

Amazon CloudFront를 사용하는 경우 MQAR을 구현하여 실시간 지표를 기반으로 최고 품질의 콘텐츠 소스를 자동으로 선택합니다. 자세한 내용은 CloudFront 통합 섹션 [CloudFront 통합](#)의 섹션을 참조하세요.

매니페스트 필터링

매니페스트 필터링을 사용하면 MediaTailor가 최종 사용자에게 제공하는 매니페스트에 포함되는 변환(동일한 콘텐츠의 다양한 품질 버전)을 사용자 지정할 수 있습니다. 필터링은 특정 디바이스 또는 네트워크 조건에 적합하지 않은 변환을 제거하여 대역폭 사용을 최적화하는 데 도움이 됩니다. 예를 들어 모바일 디바이스 또는 저대역폭 연결에 대한 4K 변환을 제거할 수 있습니다.

를 사용한 매니페스트 필터링 구현에 대한 자세한 내용은 섹션을 AWS Elemental MediaPackage참조하세요 [MediaPackage CDN 통합](#).

다중 CDN 전략

다중 CDN 전략은 여러 CDN 공급자를 동시에 사용하여 안정성과 성능을 개선합니다. 한 CDN에 문제가 발생하면 트래픽이 자동으로 다른 공급자로 이동할 수 있습니다. 이 접근 방식은 신뢰성이 중요한 중요한 중요한 라이브 이벤트에 특히 유용합니다.

MediaTailor를 사용하여 다중 CDN 전략을 구현하는 방법에 대한 자세한 내용은 섹션을 참조하세요 [CDN 통합 계획](#).

MediaTailor CDN 작업 및 성능 모니터링

AWS Elemental MediaTailor 및 콘텐츠 전송 네트워크(CDN) 통합을 효과적으로 모니터링하면 안정적인 콘텐츠 전송, 최적의 성능 및 빠른 문제 감지가 보장됩니다. 이 모니터링 접근 방식은 서버 측 광고 삽입(SSAI), 채널 어셈블리 및 결합된 워크플로를 포함한 모든 MediaTailor 구현에 적용됩니다.

CDN 통합을 모니터링하면 다음을 수행할 수 있습니다.

- 최종 사용자에게 영향을 미치기 전에 문제 감지 및 해결
- 주요 성능 지표 추적 및 서비스 품질 유지
- 최적의 캐시 성능 유지 및 오리진 로드 감소
- 광고 삽입 성공률이 비즈니스 요구 사항을 충족하는지 확인

모니터링 데이터에 나타날 수 있는 파라미터 관련 문제를 해결하려면 섹션을 참조하세요 [MediaTailor 파라미터 문제 해결 가이드](#). 쿼리 파라미터 사용량 모니터링에 대한 자세한 내용은 섹션을 참조하세요 [MediaTailor 매니페스트 쿼리 파라미터](#).

필수 CDN 성능 지표

MediaTailor 구현을 통해 이러한 핵심 지표를 추적하여 CDN의 효과를 평가합니다.

캐시 적중률

측정 대상: CDN 캐시와 오리진에서 제공된 요청의 백분율입니다.

대상 값:

- 콘텐츠 세그먼트: 캐시 적중률 95% 이상
- 광고 세그먼트: 캐시 적중률 90% 이상
- 매니페스트: 구현에 따라 다름(개인화된 매니페스트는 캐시해서는 안 됨)

중요한 이유: 캐시 적중률이 높을수록 오리진 로드가 줄어들고 응답 시간이 개선되며 대역폭 비용이 절감됩니다.

오리진 요청 볼륨

측정 대상: MediaTailor 오리진 서버에 대한 요청 수입입니다.

대상 패턴: 캐시 누락 또는 새 콘텐츠가 간헐적으로 급증하면서 낮고 안정적으로 유지되어야 합니다.

중요한 이유: 높은 오리진 요청 볼륨은 캐싱 비효율성을 나타내며 MediaTailor 성능에 영향을 미칠 수 있습니다.

응답 지연 시간

측정 대상: CDN이 최종 사용자 요청에 응답하는 속도입니다.

대상 값:

- 캐시된 콘텐츠: 100ms 미만(P95)
- 오리진 요청: 500ms 미만(P95)
- 매니페스트 요청: 100ms 미만(P95)
- 세그먼트 요청: 50ms 미만(P95)

중요한 이유: 짧은 지연 시간으로 원활한 재생과 우수한 시청자 경험을 보장합니다.

오류 발생률

측정 대상: HTTP 오류가 발생하는 요청의 백분율입니다.

대상 값:

- 4xx 오류: 총 요청의 0.1% 미만
- 5xx 오류: 총 요청의 0.01% 미만
- 오리진 오류: 오리진 요청의 0.05% 미만

중요한 이유: 오류율이 높으면 최종 사용자 경험에 영향을 미치는 구성 문제 또는 서비스 문제를 나타냅니다.

MediaTailor 특정 지표

이러한 MediaTailor 지표를 CDN 지표와 함께 모니터링하여 완전한 가시성을 확보하세요.

광고 채우기 속도

주요 지표: Avail.FillRate 및 AdDecisionServer.FillRate

목표 값: 두 지표 모두 90% 초과

중요한 이유: 광고 수익 및 시청자 경험에 직접 영향을 미칩니다.

매니페스트 생성 성능

주요 지표: GetManifest.Latency 및 GetManifest.Errors

목표 값: 200ms 미만의 지연 시간, 1% 미만의 오류율

중요한 이유: 재생 시작 시간과 신뢰성에 영향을 미칩니다.

광고 결정 서버 상태

주요 지표: AdDecisionServer.Latency, AdDecisionServer.Errors 및 AdDecisionServer.Timeouts

목표 값: 1000ms 미만의 지연 시간, 5% 미만의 오류율, 최소 제한 시간

중요한 이유: ADS 성능은 광고 삽입 성공에 직접적인 영향을 미칩니다.

모니터링 도구 설정

MediaTailor 및 CDN 통합을 효과적으로 모니터링하도록 다음 도구를 구성합니다.

Amazon CloudWatch 통합

Amazon CloudWatch는 MediaTailor 및 CDN 통합을 모니터링하기 위한 기반을 제공합니다.

MediaTailor 지표

MediaTailor는 요청, 응답 및 오류를 추적하는 지표를 CloudWatch에 자동으로 게시합니다. 주요 지표는 다음과 같습니다.

- RequestCount: MediaTailor에 대한 총 요청 수
- ResponseTime: MediaTailor 응답 지연 시간
- 4xxErrorCount 및 5xxErrorCount: 오류 추적

MediaTailor 지표의 전체 목록은 섹션을 참조하세요 [Amazon CloudWatch 지표 AWS Elemental MediaTailor를 사용한 모니터링](#).

CDN 지표

다음에 포함하여 CloudWatch에서 CDN 지표 수집을 활성화합니다.

- 다양한 콘텐츠 유형에 대한 캐시 적중률
- 오리진 요청 수 및 응답 시간
- 상태 코드별 오류 발생률

CloudFront 특정 지표의 경우 배포 설정에서 세부 모니터링을 활성화합니다.

대시보드 구성

MediaTailor 및 CDN 성능에 대한 가시성을 제공하는 대시보드를 생성합니다.

1. 통합 대시보드 생성: MediaTailor와 CDN 지표를 단일 CloudWatch 대시보드에서 결합하여 완벽한 가시성을 확보합니다.
2. 워크플로별 구성: 구현 유형(SSAI, 채널 어셈블리 또는 결합된 워크플로)별로 지표를 그룹화합니다.
3. 주요 성과 지표 포함:
 - 시간 경과에 따른 캐시 적중률 추세
 - 응답 지연 시간 백분위수(P50, P95, P99)

- 오류율 추세 및 급증

자세한 대시보드 생성 지침은 [CloudWatch 대시보드를 참조하세요](#).

알림 구성

최종 사용자에게 영향을 미치기 전에 문제를 감지하도록 알림을 구성합니다.

중요 알림

심각한 문제에 대한 즉각적인 알림을 구성합니다.

- 높은 오류율: 5xx 오류가 5분 동안 요청의 0.1%를 초과하면 알림
- 오리진 서버 문제: 오리진 응답 시간이 2초를 초과하면 알림
- 캐시 적중률 저하: 캐시 적중률이 매니페스트의 경우 70%, 세그먼트의 경우 85% 미만으로 떨어질 때 알림

경고 알림

성능 저하에 대한 조기 경고를 구성합니다.

- 지연 시간 증가: P95 응답 시간이 200ms를 초과하면 알림
- 캐시 효율성 저하: 세그먼트의 캐시 적중률이 90% 미만으로 떨어질 때 알림

구현 체크리스트

이 체크리스트를 사용하여 포괄적인 모니터링 범위를 보장합니다.

1. 지표 컬렉션:

- ✓ CloudWatch에서 MediaTailor 지표 활성화
- ✓ CDN 세부 모니터링 활성화

2. 대시보드 설정:

- ✓ 통합 CloudWatch 대시보드 생성
- ✓ 적절한 시간 범위로 시각화된 주요 지표
- ✓ 대시보드를 관련 팀과 공유

3. 알림 구성:

- ✓ 즉시 알림으로 구성된 중요 알림
- ✓ 조기 탐지를 위한 경고 설정

- ✓ 알림 에스컬레이션 절차 문서화

4. 운영 절차:

- ✓ 인시던트 대응 절차 문서화
- ✓ 정기 검토 일정 설정
- ✓ 팀 교육 완료

관련 주제

특정 모니터링 시나리오에 대한 추가 지침은 다음과 같습니다.

- 성능 최적화: 모니터링 데이터를 기반으로 하는 자세한 최적화 기법은 섹션을 참조하세요 [CDN 성능 최적화](#).
- 문제 해결: 모니터링 데이터를 사용하는 자세한 문제 해결 절차는 워크플로별 문제 해결 설명서를 참조하세요.
- 로그 분석: 포괄적인 로그 분석 및 모니터링은 섹션을 참조하세요 [MediaTailor에 대한 CDN 통합 로그 분석 및 오류 코드 참조](#).

CDN 및 MediaTailor 통합 테스트 및 검증

AWS Elemental MediaTailor 콘텐츠 전송 네트워크(CDN) 통합에는 안정적인 광고 전송과 최적의 성능을 보장하기 위한 철저한 테스트가 필요합니다. 체계적인 테스트를 통해 통합 문제가 최종 사용자에게 영향을 미치기 전에 식별할 수 있습니다. 이 포괄적인 접근 방식을 사용하여 다양한 시나리오, 구성 및 로드 조건에서 콘텐츠 전송 네트워크 및 MediaTailor 통합을 검증합니다.

관련 주제:

- 모니터링 및 로깅을 포함한 운영 설정은 섹션을 참조하세요. [CDN 모니터링](#)
- 문제가 발생할 때 문제를 해결하려면 섹션을 참조하세요. [CDN 통합 문제 해결](#)
- 성능 최적화 지침은 섹션을 참조하세요. [CDN 및 MediaTailor 통합을 위한 성능 최적화 가이드](#)

주제

- [CDN 및 MediaTailor 통합을 위한 사전 조건 및 설정 테스트](#)
- [CDN 및 MediaTailor 통합을 위한 체계적인 테스트 방법론](#)
- [CDN 및 MediaTailor 통합을 위한 배포 전 테스트 체크리스트](#)

- [CDN 및 MediaTailor 통합을 위한 도구 및 유틸리티 참조 테스트](#)

CDN 및 MediaTailor 통합을 위한 사전 조건 및 설정 테스트

AWS Elemental MediaTailor 콘텐츠 전송 네트워크(CDN) 통합 테스트에는 적절한 준비 및 리소스 할당이 필요합니다. 체계적인 테스트를 시작하기 전에 필요한 리소스와 도구가 있는지 확인합니다.

필요한 리소스:

- 프로덕션 설정을 미러링하도록 구성된 CDN 배포 테스트
- 알려진 콘텐츠 및 광고 소스를 사용하여 MediaTailor 구성 테스트
- 예측 가능한 특성(기간, 형식, 광고 중단 마커)이 있는 콘텐츠 테스트
- 광고 결정 서버 또는 모의 ADS 응답 테스트
- 여러 테스트 디바이스 및 플레이어 유형

테스트 도구:

- curl HTTP 요청 테스트용
- ffprobe HLS 매니페스트 검증용
- mp4box DASH 매니페스트 검증용
- 네트워크 분석을 위한 브라우저 개발자 도구
- end-to-end 테스트를 위한 비디오 플레이어

CDN 및 MediaTailor 통합을 위한 체계적인 테스트 방법론

AWS Elemental MediaTailor 콘텐츠 전송 네트워크(CDN) 통합 테스트는 포괄적인 적용 범위를 보장하기 위해 체계적인 단계별 접근 방식을 따라야 합니다. 이 구조화된 접근 방식을 따라 콘텐츠 전송 네트워크 및 MediaTailor 통합을 포괄적으로 테스트합니다. 각 단계는 이전 단계를 기반으로 구축되어 잠재적 문제를 격리합니다.

체계적인 테스트 접근 방식에 대한 추가 지침은 AWS Well-Architected Framework의 [신뢰성 테스트](#)를 참조하세요.

1단계: 직접 MediaTailor 연결 테스트

먼저 CDN 개입 없이 MediaTailor 기능을 테스트하여 기준을 설정합니다.

1. MediaTailor 엔드포인트로 직접 매니페스트 요청을 테스트합니다.

- HLS 다변량 재생 목록 요청을 테스트합니다. `curl -v "https://your-emt-endpoint.mediatailor.region.amazonaws.com/v1/master/hls/config-name/master.m3u8"`
- DASH MPD 요청 테스트: `curl -v "https://your-emt-endpoint.mediatailor.region.amazonaws.com/v1/dash/config-name/manifest.mpd"`
- 매니페스트 응답에 예상 광고 중단 마커가 포함되어 있는지 확인
- 세그먼트 URLs 오리진을 가리키는지 확인

2. 광고 삽입이 올바르게 작동하는지 확인합니다.

- 다양한 광고 대상 지정 파라미터로 테스트
- 광고 세그먼트가 올바르게 트랜스코딩되고 사용 가능한지 확인
- 광고 중단 시간 및 기간 확인
- 광고를 사용할 수 없는 경우 폴백 동작 테스트

3. 기준 성능 측정:

- 매니페스트 요청 응답 시간 기록
- 광고 결정 서버 응답 시간 측정
- 세션 생성 및 관리 테스트

성공 기준: 모든 직접 MediaTailor 요청은 예상 광고 콘텐츠를 포함하는 올바른 형식의 매니페스트와 함께 HTTP 200 응답을 반환합니다.

2단계: 기본 CDN 통합 테스트

요청 경로에 CDN을 추가하고 기본 기능을 테스트합니다.

1. CDN을 통해 매니페스트 요청을 테스트합니다.

- 간단한 라우팅 규칙을 사용하여 CDN 구성
- CDN 엔드포인트를 통해 매니페스트 요청 테스트
- CDN이 MediaTailor에 요청을 올바르게 전달하는지 확인
- 매니페스트 응답이 캐시되지 않았는지 확인(TTL = 0)

2. 세그먼트 라우팅 테스트:

- 콘텐츠 세그먼트가 오리진 서버로 라우팅되는지 확인

- 광고 세그먼트가 MediaTailor 광고 스토리지로 라우팅되는지 확인
- 세그먼트 캐싱 동작 테스트

3. CDN과 직접 성능 비교:

- 응답 시간 차이 측정
- 응답의 콘텐츠 차이 확인
- 오류 처리가 올바르게 작동하는지 확인

성공 기준: CDN은 성능 영향을 최소화하면서 MediaTailor 및 오리진 서버에 요청을 성공적으로 프록시 처리해야 합니다.

3단계: 쿼리 파라미터 전달 테스트

쿼리 파라미터 전달을 추가하고 광고 개인화를 테스트합니다.

1. CDN에서 쿼리 파라미터 전달을 구성합니다.

- MediaTailor로 모든 쿼리 파라미터 전달 활성화
- 테스트 세션 초기화(세션 ID는 첫 요청 시 MediaTailor에서 자동으로 생성됨)
- 사용자 지정 대상 지정 파라미터로 테스트

2. 광고 개인화 테스트:

- 다른 파라미터가 다른 광고 응답을 생성하는지 확인
- 파라미터 인코딩 및 특수 문자 테스트
- 파라미터가 ADS에 올바르게 전달되었는지 확인

3. 세션 관리 검증:

- 세션 생성 및 지속성 테스트
- 요청 간 세션 ID 일관성 확인
- 세션 만료 처리 테스트

성공 기준: 광고 콘텐츠는 쿼리 파라미터에 따라 다르며 세션이 올바르게 관리됩니다.

4단계: 헤더 전달 테스트

CDN에 헤더 전달을 추가하고 디바이스별 대상을 테스트합니다.

1. 모든 헤더에 대한 헤더 전달을 구성합니다. 최소 요구 사항은 섹션을 참조하세요 [MediaTailor CDN 통합에 필요한 헤더](#).

2. 테스트 디바이스 대상:

- 다양한 사용자 에이전트 문자열(모바일, 데스크톱, TV)로 테스트
- 디바이스별 광고 응답 확인
- IP 주소가 다른 지리적 대상 테스트

3. CORS 처리 검증:

- CORS 헤더의 웹 플레이어 호환성 테스트
- 사전 OPTIONS 요청이 올바르게 작동하는지 확인
- 다른 도메인에서 테스트

성공 기준: 디바이스 및 지리적 타겟팅이 올바르게 작동해야 하며 웹 플레이어에게 CORS 오류가 발생하지 않아야 합니다.

5단계: 포괄적인 시나리오 테스트

여러 시나리오에서 테스트하여 강력한 작동을 보장합니다.

1. 다양한 플레이어 유형으로 테스트합니다.

- 웹 브라우저의 HLS.js 플레이어
- HLS 및 DASH를 지원하는 Video.js 플레이어
- 모바일 디바이스의 네이티브 플레이어
- 스마트 TV 및 셋톱 박스 플레이어

2. 다양한 디바이스 및 플랫폼에서 테스트:

- 모바일 디바이스(iOS, Android)
- 데스크톱 브라우저(Chrome, Firefox, Safari, Edge)
- 스마트 TVs 및 스트리밍 디바이스
- 다양한 운영 체제 버전

3. 다양한 콘텐츠 유형을 테스트합니다.

- 라이브 스트리밍 콘텐츠
- 온디맨드 비디오(VOD) 콘텐츠
- 다양한 비디오 형식 및 비트 전송률
- 광고 중단 패턴이 다른 콘텐츠

4. 광고 타겟팅 시나리오 테스트:

- 다양한 인구 통계 대상 지정 파라미터

- 여러 리전의 지리적 대상 지정
- 시간 기반 대상 지정(하루 중 서로 다른 시간)
- 사용자 지정 비즈니스 로직 파라미터

성공 기준: 모든 플레이어 및 디바이스 조합이 적절한 광고 타겟팅과 함께 올바르게 작동해야 합니다.

6단계: 로드 및 성능 테스트

실제 로드 조건에서 성능을 검증합니다.

Important

로드 테스트 전에 [AWS Support](#)에 문의하세요. 로드 및 성능 테스트를 수행하기 전에 AWS 지원 티켓을 생성하여 MediaTailor 서비스 팀에 계획된 테스트를 알립니다. 이렇게 하면 다음이 보장됩니다.

- 서비스가 예상 로드 수준에 맞게 준비됨
- 테스트 기간 동안 적절한 용량을 사용할 수 있습니다.
- 테스트가 프로덕션 인시던트로 오해되지 않습니다.
- 테스트 모범 사례 및 제한 사항에 대한 지침을 받습니다.

지원 티켓에 예상 동시 사용자, 테스트 기간, 지리적 리전 및 테스트하려는 특정 시나리오를 포함합니다.

1. 동시 사용자 시나리오 테스트:

- 여러 동시 뷰어 시뮬레이션
- CDN 조정 및 캐시 성능 테스트
- 로드 시 오리진 서버 성능 모니터링
- MediaTailor가 동시 세션을 처리할 수 있는지 확인

2. 성능 지표 측정:

- 로드 시 응답 시간 모니터링
- 캐시 적중률이 기대치를 충족하는지 확인(대중 콘텐츠의 경우 >80%)
- 다양한 시나리오의 첫 번째 프레임까지의 시간 측정
- 피크 로드 중 오류율 추적

3. 장애 조치 시나리오를 테스트합니다.

- ADS를 사용할 수 없는 경우 동작 테스트
- 오리진 서버 장애 조치 테스트
- 오류 처리 및 복구 확인
- CDN 엣지 로케이션 장애 조치 테스트

성공 기준: 시스템은 장애 발생 시 정상적인 성능 저하와 함께 예상 부하에서 허용 가능한 성능을 유지해야 합니다. 실행 전에 [AWS Support](#)에 문의하여 로드 테스트 계획을 승인해야 합니다.

CDN 및 MediaTailor 통합을 위한 배포 전 테스트 체크리스트

AWS Elemental MediaTailor 콘텐츠 전송 네트워크(CDN) 통합은 프로덕션 배포 전에 포괄적인 테스트를 통과해야 합니다. 구성 변경 사항을 프로덕션에 배포하기 전에 이 체크리스트를 사용합니다.

기본 기능:

- " 매니페스트 요청은 HTTP 200 응답을 반환합니다.
- " 콘텐츠 세그먼트가 올바르게 로드됨
- " 광고 세그먼트가 올바르게 로드됨
- " 광고 시간이 예상 시간에 표시됨
- " 재생은 콘텐츠와 광고 간에 원활하게 전환됩니다.

구성 검증:

- " 쿼리 파라미터가 올바르게 전달됨
- " 필수 헤더가 올바르게 전달됨
- " 매니페스트 캐싱이 비활성화됨(TTL = 0)
- " 세그먼트 캐싱이 적절하게 구성됨
- " CORS 헤더가 웹 플레이어용으로 구성됨

교차 플랫폼 테스트:

- " 모바일 디바이스에서 테스트됨
- " 데스크톱 브라우저에서 테스트됨
- " 다양한 플레이어 유형으로 테스트됨

- " HLS 및 DASH 형식 모두 테스트됨

성능 검증:

- 로드 테스트 승인을 위해 생성된 AWS 지원 티켓
- " 응답 시간이 성능 목표 충족
- " 캐시 적중률 허용
- " 오류 발생률이 허용 한도 내에 있음
- " 모니터링 및 알림이 구성됨

CDN 및 MediaTailor 통합을 위한 도구 및 유틸리티 참조 테스트

AWS Elemental MediaTailor 콘텐츠 전송 네트워크(CDN) 통합 테스트에는 포괄적인 검증 및 디버깅을 위한 다양한 도구가 필요합니다. 콘텐츠 전송 네트워크 및 MediaTailor 통합 테스트에 일반적으로 사용되는 도구에 대한 참조 가이드입니다.

curl - HTTP 요청 테스트

매니페스트 요청 테스트: `curl -v "https://your-cdn-domain.com/v1/master/hls/config/master.m3u8"`

헤더로 테스트: `curl -H "User-Agent: TestAgent/1.0" "https://your-cdn-domain.com/..."`

파라미터로 테스트: `curl "https://your-cdn-domain.com/...?aws.sessionId=test123"`

ffprobe - HLS 매니페스트 검증

HLS 구문 검증: `ffprobe -v quiet -print_format json -show_format "https://your-cdn-domain.com/master.m3u8"`

세그먼트 정보 확인: `ffprobe -v quiet -show_entries packet=pts_time "segment.ts"`

mp4box - DASH 매니페스트 검증

DASH MPD 검증: `mp4box -info "https://your-cdn-domain.com/manifest.mpd"`

세그먼트 타이밍 확인: `mp4box -info segment.m4s`

브라우저 개발자 도구

네트워크 탭에서 네트워크 요청 모니터링

콘솔 탭에서 CORS 오류 확인

요청/응답 헤더 검사

타이밍 및 성능 지표 분석

추가 리소스:

- [CloudFront 문제 해결](#) - 포괄적인 CDN 문제 해결 가이드
- [CloudFront 캐시 적중률 증가](#) - 성능 최적화 지침
- [AWS 리소스 모니터링](#) - 성능 모니터링 모범 사례

MediaTailor 및 CDN 통합 관련 문제 해결

이 포괄적인 문제 해결 가이드에서는 서버 측 광고 삽입(SSAI), 채널 어셈블리 및 통합을 포함한 모든 AWS Elemental MediaTailor 구현에서 일반적인 콘텐츠 전송 네트워크(CDN) AWS Elemental MediaPackage 통합 문제를 다룹니다. CDN 및 MediaTailor 통합에 문제가 발생하면이 체계적인 진단 접근 방식을 사용하여 근본 원인을 신속하게 식별하고 검증된 솔루션을 구현합니다.

이 가이드는 특정 워크플로에 관계없이 모든 MediaTailor CDN 통합에 적용됩니다. 특정 서비스 또는 워크플로와 관련된 문제는이 가이드의 끝에 참조된 관련 문제 해결 섹션을 참조하세요.

시작하기 전에: 효율적인 문제 해결을 위해 다음 항목을 준비합니다.

- 문제를 보여주는 샘플 재생 URLs
- 문제가 발생한 기간의 CDN 액세스 로그
- MediaTailor 구성 이름 및 AWS 리전
- 플레이어 유형 및 버전(예: HLS.js 1.4.0, Video.js 8.0)
- 문제가 발생하는 디바이스 및 브라우저 정보

관련 주제:

- 운영 설정 및 문제 해결 준비는 섹션을 참조하세요. [CDN 통합 문제 해결](#)

- 로그 분석 및 오류 코드 참조는 섹션을 참조하세요. [CDN 통합 로그 분석 참조](#)
- 에스컬레이션 및 추가 도움말은 섹션을 참조하세요. [CDN 통합 지원 받기](#)

MediaTailor 및 CDN 통합을 위한 진단 체크리스트

AWS Elemental MediaTailor 콘텐츠 전송 네트워크(CDN) 통합 문제는 다양한 방식으로 나타날 수 있습니다. 이 체크리스트를 사용하여 발생하는 문제 유형을 빠르게 식별할 수 있습니다.

1. 문제가 모든 최종 사용자 또는 특정 최종 사용자에게 영향을 미치나요?

- 모든 뷰어 → 가능한 CDN 또는 MediaTailor 구성 문제
- 특정 뷰어 → 개인화 또는 타겟팅 문제일 가능성이 높음

2. 매니페스트가 올바르게 로드되고 있습니까?

- 아니요 → CDN 라우팅 또는 MediaTailor 연결 문제
- 예, 하지만 잘못된 콘텐츠 → 캐싱 또는 개인화 문제

3. 세그먼트가 올바르게 로드되고 있습니까?

- 콘텐츠 세그먼트 실패 → 오리진 연결 문제
- 광고 세그먼트 실패 → 광고 전송 또는 트랜스코딩 문제

4. 광고가 올바르게 삽입되고 있습니까?

- 광고가 표시되지 않음 → ADS 연결 및 구성 확인
- 잘못된 광고 표시 → 광고 대상 지정 파라미터 및 개인화 확인
- 광고 재생 실패 → 광고 트랜스코딩 및 세그먼트 가용성 확인

5. 재생이 원활하고 중단되지 않나요?

- 버퍼링 문제 → CDN 캐시 성능 및 오리진 응답 시간 확인
- 재생 오류 → 매니페스트 구문 및 세그먼트 가용성 확인
- 광고 전환 문제 → 광고 중단 타이밍 및 세그먼트 정렬 확인

6. 특정 오류 코드 또는 메시지가 있습니까?

- HTTP 4xx 오류 → CDN 라우팅 및 구성 확인
- HTTP 5xx 오류 → 오리진 서버 및 MediaTailor 서비스 상태 확인
- 플레이어별 오류 → 매니페스트 형식 및 플레이어 호환성 확인

진단에 따른 다음 단계:

CDN 구성 문제

CDN 라우팅 및 캐싱 문제 해결에 대한 자세한 내용은 섹션을 참조하세요 [MediaTailor 및 CDN 통합 관련 문제 해결](#).

매니페스트 및 재생 문제

매니페스트 검증 및 재생 문제 해결은 섹션을 참조하세요 [CDN 통합 테스트 절차](#).

광고 삽입 및 타겟팅 문제

ADS 연결 및 광고 전송을 포함한 광고별 문제 해결은 워크플로별 문제 해결 설명서를 참조하세요.

성능 및 모니터링 문제

성능 분석 및 모니터링 설정은 섹션을 참조하세요 [MediaTailor CDN 작업 및 성능 모니터링](#).

로그 분석 및 오류 코드

자세한 로그 분석 및 오류 코드 참조는 섹션을 참조하세요 [MediaTailor에 대한 CDN 통합 로그 분석 및 오류 코드 참조](#).

테스트 및 검증

포괄적인 테스트 절차는 섹션을 참조하세요 [CDN 및 MediaTailor 통합 테스트 및 검증](#).

즉각적인 지원이 필요하거나 연결된 리소스를 사용하여 문제를 해결할 수 없는 경우 에스컬레이션 절차는 섹션을 참조 [CDN 및 MediaTailor 통합에 대한 지원 및 문제 해결 도움말 받기](#)하세요.

CDN 통합 테스트 절차

MediaTailor CDN 통합을 프로덕션에 배포하기 전에 적절한 테스트가 필수적입니다. 이러한 테스트 절차는 다양한 디바이스 및 플랫폼에서 구성 문제, 성능 문제 및 호환성 문제를 식별하는 데 도움이 됩니다.

기본 통합 검증

다음 기본 테스트를 수행하여 CDN 통합이 올바르게 작동하는지 확인합니다.

1. 매니페스트 전송 테스트:

- CDN을 통해 매니페스트를 요청하고 유효한 응답을 반환하는지 확인합니다.
- 매니페스트에 예상 콘텐츠와 광고 삽입 지점이 포함되어 있는지 확인합니다.
- 매니페스트 URLs이 아닌 CDN 도메인을 사용하는지 확인

- HLS 또는 DASH 검증 도구를 사용하여 매니페스트 구문 검증
2. URL 재작성 확인:
 - 매니페스트의 콘텐츠 세그먼트 URLs이 CDN 도메인을 가리키는지 확인
 - 광고 세그먼트 URLs CDN 도메인을 가리키는지 확인
 - 모든 관련 URLs 제대로 확인되었는지 확인
 3. 콘텐츠 재생 테스트:
 - 비디오 플레이어를 통해 콘텐츠 재생 및 원활한 재생 확인
 - 콘텐츠와 광고가 중단 없이 재생되는지 확인
 - 콘텐츠와 광고 간의 적절한 전환 확인
 - 탐색 및 스크러빙 기능 테스트
 4. CDN 라우팅 검증:
 - CDN 액세스 로그를 모니터링하여 요청이 올바르게 라우팅되었는지 확인합니다.
 - 캐시 적중/누락 패턴이 예상대로인지 확인
 - 오리진 요청이 캐시 누락에 대해서만 발생하는지 확인

고급 테스트 절차

포괄적인 검증을 위해 다음 추가 테스트를 수행합니다.

1. 교차 플랫폼 호환성 테스트:
 - 여러 디바이스(데스크톱, 모바일, 태블릿, 스마트 TV)에서 테스트
 - 여러 브라우저 간의 호환성 확인
 - 다양한 비디오 플레이어(HLS.js, Video.js, 네이티브 플레이어)로 테스트
 - 다양한 운영 체제에서 검증
2. 성능 테스트:
 - 매니페스트 요청 응답 시간 측정(대상: 캐시된 경우 <100ms)
 - 다양한 비트 전송률에서 세그먼트 로드 성능 테스트
 - 시작 시간이 성능 목표를 충족하는지 확인
 - 다양한 네트워크 조건에서 테스트
3. 광고 추적 검증:
 - 광고 추적 비컨이 올바르게 실행되는지 확인
 - 광고 분석 데이터의 정확성 확인

- 노출 및 완료 추적 테스트
- 클릭 기능 검증

4. 오류 조건 테스트:

- 오리진을 일시적으로 사용할 수 없는 경우 동작 테스트
- 잘못된 형식의 요청에 대한 정상적인 처리 확인
- CDN 장애 조치 시나리오 테스트
- 오류 메시지의 명확성과 유용성 검증

테스트 환경 생성

포괄적인 검증을 위해 프로덕션 구성을 미러링하는 테스트 환경을 설정합니다.

1. 테스트를 위해 별도의 CDN 배포를 설정합니다.

- 프로덕션과 동일한 캐시 동작으로 테스트 CDN 배포 생성
- 프로덕션 설정을 미러링하는 테스트 오리진 구성
- 별도의 도메인 이름을 사용하여 프로덕션 트래픽과의 충돌 방지

2. 테스트 MediaTailor 구성을 생성합니다.

- 프로덕션과 동일한 설정으로 테스트 재생 구성 설정
- 테스트 광고 결정 서버 엔드포인트 구성
- 프로덕션 광고 형식과 일치하는 테스트 광고 콘텐츠 사용

3. 체계적인 테스트 프로세스를 구현합니다.

- 구성 변경에 대한 테스트 체크리스트 생성
- 팀의 테스트 절차 문서화
- 가능한 경우 자동 테스트 설정

여러 시나리오에서 테스트

다양한 시나리오 및 조건에서 통합을 검증하여 포괄적인 적용 범위를 보장합니다.

1. 여러 플레이어 유형 및 디바이스로 테스트:

- 다양한 비디오 플레이어(웹, 모바일, 연결된 TV)로 테스트
- 다양한 운영 체제 및 브라우저에서 검증
- 다양한 네트워크 조건 및 연결 속도에 대한 테스트

2. 자동 테스트 스크립트 생성:

- 매니페스트 요청 검증 자동화
- 스크립트를 생성하여 광고 삽입 시나리오 테스트
- 트래픽이 많은 시나리오에 대한 성능 테스트 구현

3. 광고 타겟팅 및 개인화 검증:

- 다양한 사용자 프로필 및 대상 지정 파라미터로 테스트
- 광고 결정 서버 통합 검증
- 광고를 사용할 수 없을 때 대체 시나리오 테스트

도구 및 기법 테스트

효과적인 테스트를 위해 다음 도구와 기법을 사용합니다.

브라우저 개발자 도구

네트워크 탭을 사용하여 HTTP 요청 및 응답 검사

콘솔에서 JavaScript 오류 및 경고 모니터링

응답 헤더 및 캐싱 동작 확인

성능 분석을 위한 타이밍 정보 확인

명령줄 테스트

curl을 사용하여 특정 URLs 테스트하고 헤더를 검사합니다.

```
curl -I "https://your-cdn-domain.com/path/to/manifest.m3u8"
```

테스트 및 타이밍 분석 다운로드에 wget 사용

보다 읽기 쉬운 HTTP 테스트를 위해 httpie와 같은 도구 사용

비디오 플레이어 테스트

여러 플레이어 구현으로 테스트

플레이어 디버그 모드를 사용하여 내부 동작 검사

플레이어 이벤트 및 오류 콜백 모니터링

적응형 비트 전송률 전환 동작 검증

CDN 분석 및 모니터링

테스트 중 실시간 CDN 지표 모니터링

요청 패턴에 대한 액세스 로그 검토

사용 가능한 경우 CDN별 테스트 도구 사용

테스트 검증을 위한 임시 알림 설정

추가적인 포괄적인 테스트 방법론 및 체계적인 검증 접근 방식은 [섹션을 참조하세요](#) [CDN 및 MediaTailor 통합 테스트 및 검증](#).

MediaTailor의 CDN 매니페스트 404 오류 문제 해결

AWS Elemental MediaTailor 콘텐츠 전송 네트워크(CDN) 매니페스트 404 오류는 재생을 시작하지 못하게 하는 일반적인 통합 문제입니다. 이 섹션에서는 매니페스트 전송 실패에 대한 step-by-step 문제 해결을 제공합니다.

다변량 재생 목록, 미디어 재생 목록 또는 MPD 요청은 404 오류를 반환합니다.

빠른 수정(먼저 시도):

1. URL의 MediaTailor 구성 이름이 정확히 일치하는지 확인(대/소문자 구분)
2. CDN이 없는 MediaTailor에 대해 매니페스트 URL을 직접 테스트합니다. `curl -v "https://your-emt-endpoint.mediatailor.region.amazonaws.com/v1/master/hls/config-name/master.m3u8"`
3. 직접 테스트가 작동하는 경우 매니페스트 요청에 대한 CDN 라우팅 규칙 확인

빠른 수정이 작동하지 않는 경우:

증상: 플레이어는 재생을 시작하지 못하고 매니페스트 요청이 CDN 로그에 HTTP 404 오류를 반환합니다.

오류 메시지의 예:

- 브라우저 콘솔: "Failed to load resource: the server responded with a status of 404 (Not Found)"
- 플레이어 오류: "MANIFEST_LOAD_ERROR" 또는 "NETWORK_ERROR"
- CDN 로그: GET /v1/master/hls/example-config/master.m3u8 404

해결 방법:

CDN 라우팅 규칙이 다변량 재생 목록, 미디어 재생 목록 및 MPD 요청을 MediaTailor로 전달하도록 올바르게 구성되어 있는지 확인합니다.

MediaTailor 구성이 존재하고 올바르게 설정되었는지 확인합니다.

CDN 동작 패턴이 예상 매니페스트 요청 경로(예: , *.m3u8*.mpd)와 일치하는지 확인합니다.

MediaTailor에 대한 CDN 매니페스트 전송 문제 및 오류 진단

AWS Elemental MediaTailor 콘텐츠 전송 네트워크(CDN) 매니페스트 전송 문제는 적절한 광고 삽입 및 재생을 방해할 수 있습니다. 최종 사용자가 다변량 재생 목록, 미디어 재생 목록 또는 MPDs에서 잘못된되거나 일관되지 않은 광고를 수신하는 경우:

1. 캐시된 매니페스트 확인:

- 모든 다변량 재생 목록, 미디어 재생 목록 및 MPD 경로에 대해 TTL 설정이 0으로 설정되어 있는지 확인
- TTL 설정에도 불구하고 CDN이 다변량 재생 목록, 미디어 재생 목록 또는 MPDs를 캐싱하지 않는지 확인
- CDN 로그에서 캐시 상태 확인 - 매니페스트 요청은 Miss 또는를 표시해야 하며 RefreshHit Hit

2. CDN 라우팅 구성을 확인합니다.

- 매니페스트 요청이 오리진에서 캐싱되거나 제공되지 않고 MediaTailor 엔드포인트로 라우팅되고 있는지 확인
- CDN 동작 패턴이 매니페스트 경로(*.m3u8, *.mpd)와 올바르게 일치하는지 확인합니다.
- 쿼리 파라미터가 개인화를 위해 MediaTailor로 전달되고 있는지 확인
- MediaTailor에 대해 매니페스트 URLs 직접 테스트하여 CDN과 서비스 문제를 격리합니다.

3. 헤더 전달 구성을 확인합니다.

- 필요한 헤더가 전달되고 있는지 확인(참조 [MediaTailor CDN 통합에 필요한 헤더](#))
- 디바이스별 광고 타겟팅을 위해 사용자 에이전트 헤더가 전달되었는지 확인
- 지리 대상 지정에 X-Forwarded-For 헤더가 전달되었는지 확인
- 압축 지원을 위해 Accept-Encoding 헤더가 전달되었는지 확인

4. 매니페스트 콘텐츠 및 구조를 검증합니다.

- 매니페스트에 예상 광고 삽입 마커가 포함되어 있는지 확인(HLS의 경우EXT-X-CUE-OUT/IN)

- 매니페스트의 세그먼트 URLs 오리진 도메인이 아닌 CDN 도메인을 사용하는지 확인
- 광고 세그먼트가 제대로 삽입되고 액세스 가능한지 확인
- 검증 도구를 사용하여 매니페스트 구문 테스트(HLS의 경우 ffprobe, DASH의 경우 mp4box)

5. 다양한 시나리오를 테스트합니다.

- 다른 세션 IDs로 테스트하여 개인 맞춤이 작동하는지 확인
- 다양한 지리적 위치에서 테스트하여 지리적 대상 확인
- 다양한 User-Agent 문자열로 테스트하여 디바이스 대상 확인
- CDN이 있는 매니페스트 응답과 없는 매니페스트 응답을 비교하여 차이점 식별

추가 문제 해결 리소스:

- 자세한 CDN 캐싱 구성은 섹션을 참조하세요. [CDN 및 MediaTailor 통합을 위한 캐싱 최적화](#)
- 포괄적인 CDN 라우팅 설정은 섹션을 참조하세요. [MediaTailor에 대한 CDN 라우팅 동작 설정](#)
- 헤더 전달 요구 사항은 섹션을 참조하세요. [MediaTailor CDN 통합에 필요한 헤더](#)
- 로그 분석 및 오류 코드는 섹션을 참조하세요. [MediaTailor에 대한 CDN 통합 로그 분석 및 오류 코드 참조](#)
- 테스트 절차 및 검증은 섹션을 참조하세요. [CDN 및 MediaTailor 통합 테스트 및 검증](#)

성공 기준: 해결되면 플레이어는 정상적으로 재생을 시작하고 광고가 예상대로 표시되어야 합니다. 매니페스트 요청은 CDN 로그에 HTTP 200 상태 코드를 반환해야 하며 매니페스트에는 적절하게 개인화된 광고 콘텐츠가 포함되어야 합니다.

MediaTailor의 CDN 세그먼트 전송 및 로드 문제 해결

AWS Elemental MediaTailor 콘텐츠 전송 네트워크(CDN) 세그먼트 전송 문제로 인해 버퍼링 및 재생이 중단될 수 있습니다. 플레이어가 세그먼트를 로드하거나 버퍼링을 경험할 수 없는 경우:

1. CDN 라우팅 규칙 확인:

- 콘텐츠 세그먼트가 올바른 오리진으로 라우팅되고 있는지 확인
- 광고 세그먼트가 올바른 MediaTailor 광고 스토리지 위치로 라우팅되고 있는지 확인
- 세그먼트 파일 확장명이 CDN 동작 패턴과 일치하는지 확인
- 매니페스트URLs이 올바른 CDN 도메인을 사용하는지 확인
- 라우팅 및 동작 경로 패턴 설정에 대한 자세한 지침은 섹션을 참조하세요. [MediaTailor에 대한 CDN 라우팅 동작 설정](#)

2. CORS 구성 확인:

- 웹 플레이어의 경우 CDN이 CORS 헤더를 통과하거나 올바르게 설정되었는지 확인합니다.
- 브라우저 개발자 도구를 사용하여 테스트하여 CORS 관련 오류 식별
- preflight OPTIONS 요청이 올바르게 처리되었는지 확인

3. 세그먼트 접근성 및 성능 테스트:

- 개별 세그먼트 URLs 직접 테스트하여 액세스할 수 있는지 확인
- 세그먼트 응답 시간 확인 및 성능 병목 현상 식별
- 세그먼트 파일 크기가 대역폭 대상에 적합한지 확인
- 서로 다른 지리적 위치에서 세그먼트 로드 테스트

4. 세그먼트에 대한 CDN 캐싱 동작을 검증합니다.

- 콘텐츠 세그먼트에 적절한 TTL 설정이 있는지 확인(일반적으로 매니페스트보다 김)
- 광고 세그먼트가 개인화 요구 사항에 따라 적절하게 캐시되었는지 확인
- 콘텐츠 및 광고 세그먼트 모두에 대한 캐시 적중률 모니터링
- 캐시 키에 캐시 효율성을 줄이는 불필요한 파라미터가 포함되어 있지 않은지 확인합니다.

5. 오리진 서버 연결 및 상태 확인:

- 오리진 서버가 세그먼트 요청에 올바르게 응답하는지 확인
- 로드 중인 오리진 서버 용량 및 응답 시간 확인
- 오리진 서버에 사용 가능한 예상 세그먼트 파일이 있는지 확인
- 여러 오리진이 구성된 경우 오리진 장애 조치 시나리오 테스트

6. 광고 세그먼트별 문제 해결:

- MediaTailor에서 광고 세그먼트가 제대로 트랜스코딩되고 사용 가능한지 확인
- 매니페스트에서 광고 세그먼트 URLs 올바르게 생성되었는지 확인
- 다양한 광고 대상 지정 파라미터로 광고 세그먼트 로드 테스트
- 세그먼트를 사용할 수 없게 만들 수 있는 광고 트랜스코딩 지연 모니터링

7. 플레이어 호환성 및 동작을 검증합니다.

- 다양한 플레이어 유형 및 버전으로 세그먼트 로드 테스트
- 플레이어 버퍼 설정 및 세그먼트 요청 패턴 확인
- 실패한 세그먼트 요청에 대한 플레이어 오류 처리 확인
- 적응형 비트 전송률 전환 및 세그먼트 선택 로직 테스트

추가 문제 해결 리소스:

- CDN 라우팅 및 동작 구성은 섹션을 참조하세요. [MediaTailor에 대한 CDN 라우팅 동작 설정](#)
- CDN 캐싱 최적화는 섹션을 참조하세요. [CDN 및 MediaTailor 통합을 위한 캐싱 최적화](#)
- CORS 구성 지침은 섹션을 참조하세요. [MediaTailor에 대한 CDN 통합 보안 모범 사례](#)
- 성능 모니터링 및 분석은 섹션을 참조하세요. [MediaTailor CDN 작업 및 성능 모니터링](#)
- 포괄적인 테스트 절차는 섹션을 참조하세요. [CDN 및 MediaTailor 통합 테스트 및 검증](#)
- 로그 분석 및 오류 진단은 섹션을 참조하세요. [MediaTailor에 대한 CDN 통합 로그 분석 및 오류 코드 참조](#)

성공 기준: 해결되면 플레이어는 버퍼링 중단 없이 세그먼트를 원활하게 로드해야 합니다. 세그먼트 요청은 적절한 응답 시간과 함께 HTTP 200 상태 코드를 반환해야 하며 콘텐츠 및 광고 세그먼트 모두에 액세스하고 적절하게 캐싱해야 합니다.

MediaTailor에 대한 CDN 세션 관리 및 추적 문제 해결

AWS Elemental MediaTailor 콘텐츠 전송 네트워크(CDN) 세션 관리는 적절한 광고 개인화 및 추적에 매우 중요합니다. 요청 간에 세션 관련 오류 또는 일관되지 않은 동작이 발생하는 경우:

1. 세션 ID 일관성을 확인합니다.

- 플레이어가 단일 재생 세션에 대한 모든 요청에서 동일한 세션 ID를 유지하는지 확인
- CDN 로그를 확인하여 세션 IDs 올바르게 전달되고 있는지 확인합니다.
- 쿼리 파라미터에서 세션 IDs 제대로 URL 인코딩되었는지 확인
- CloudWatch Logs를 사용하여 요청 간 세션 ID 일관성을 확인합니다(아래 검증 단계 참조).

2. 세션 초기화 검증:

- 첫 번째 매니페스트 요청이 세션을 성공적으로 생성하는지 확인
- 적절한 세션 파라미터 전달 확인(예: `aws.sessionId`)
- 디버그 로그를 사용하여 세션 초기화 확인(아래 디버그 로그 설정 참조)

3. 자세한 세션 문제 해결을 위해 디버그 로깅을 활성화합니다.

- 서버 측 보고의 경우: 재생 요청에 `?aws.logMode=DEBUG` 추가:

```
GET <mediatailorURL>/v1/master/<hashed-account-id>/<origin-id>/<asset-id>?
aws.logMode=DEBUG
```

- 클라이언트 측 보고의 경우: 세션 초기화 요청 본문 "logMode": "DEBUG"에 포함
- 중요: DEBUG 값은 대/소문자를 구분합니다.

- 동시에 허용되는 최대 10개의 활성 디버그 세션

4. CloudWatch Logs 쿼리를 사용하여 세션 동작을 검증합니다.

- 디버그 세션이 활성 상태인지 확인합니다.

```
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
```

- 세션의 모든 이벤트 보기:

```
fields @timestamp, @message, eventType, mediaTailorPath
| filter sessionId = "your-session-id-here"
| sort @timestamp asc
```

- 세션의 매니페스트 생성을 확인합니다.

```
fields @timestamp, responseBody, @message
| filter mediaTailorPath like "/v1/master/" and eventType = "GENERATED_MANIFEST"
and sessionId = "your-session-id-here"
```

5. CDN을 통한 세션 파라미터 전달 테스트:

- MediaTailor에 직접 세션 파라미터를 사용하여 매니페스트 요청 테스트(CDN 우회)
- CDN이 있는 세션 동작과 없는 세션 동작을 비교하여 전달 문제 식별
- CDN 쿼리 파라미터 전달 구성에 세션 관련 파라미터가 포함되어 있는지 확인
- CDN이 세션별 응답을 캐싱하지 않는지 확인

일반적인 세션 오류 메시지:

- **ConflictException (HTTP 409)** - 동일한 세션에 대한 여러 동시 재생 목록 요청입니다. 해결 방법: 플레이어에 HLS 사양에 따라 한 번에 하나씩 재생 목록을 요청하는지 확인합니다.
- **NotFoundException (HTTP 404)** - 세션을 사용할 수 없거나 구성이 존재하지 않습니다. 해결 방법: 구성 유효성을 확인하고 세션을 다시 초기화합니다.
- **BadRequestException (HTTP 400)** - 잘못된 세션 ID 또는 잘못된 형식의 요청입니다. 해결 방법: 요청 형식 및 세션 ID 유효성 확인

추가 문제 해결 리소스:

- 전체 디버그 로깅 설정 및 필드 참조는 섹션을 참조하세요. [AWS Elemental MediaTailor 디버그 로그 생성](#)
- CloudWatch Logs 쿼리 예제 및 로그 분석은 섹션을 참조하세요. [Amazon CloudWatch Logs에 직접 AWS Elemental MediaTailor 로그 작성](#)
- CDN 쿼리 파라미터 전달 구성은 섹션을 참조하세요. [MediaTailor에 대한 CDN 라우팅 동작 설정](#)
- 포괄적인 오류 코드 참조는 섹션을 참조하세요. [MediaTailor에서 재생 문제 해결](#)

성공 기준: 해결되면 세션이 올바르게 초기화되고, 요청 간에 일관된 세션 IDs 유지하며, 디버그 로그에 오류 없이 적절한 SESSION_INITIALIZED 이벤트와 매니페스트 생성이 표시되어야 합니다.

MediaTailor의 CDN 광고 중단 타이밍 및 동기화 문제 해결

AWS Elemental MediaTailor 콘텐츠 전송 네트워크(CDN) 광고 중단 타이밍은 콘텐츠 마커와 정확하게 동기화되어야 합니다. 광고가 잘못된 시간에 나타나거나 광고 중단 시간이 일치하지 않는 경우:

1. 콘텐츠에서 광고 중단 마커를 확인합니다.
 - SCTE-35 마커가 오리지널 콘텐츠에 올바르게 배치되었는지 확인
 - 광고 중단 기간이 실제 광고 콘텐츠 길이와 일치하는지 확인
 - 광고 시간 타이밍이 콘텐츠 경계와 일치하는지 확인
 - 오리지널 매니페스트에서 SCTE-35 마커 형식 및 타이밍 정확도 검증
 - 다양한 콘텐츠 유형(라이브 대 VOD)으로 광고 중단 마커 테스트
2. CDN 캐싱이 타이밍에 미치는 영향을 확인합니다.
 - 타이밍 드리프트를 방지하려면 매니페스트 TTL이 0으로 설정되어 있는지 확인합니다.
 - 시간에 민감한 파라미터가 캐시되지 않는지 확인
 - 콘텐츠 소스, MediaTailor 및 CDN 간의 클럭 동기화 문제 확인
 - 장기 실행 라이브 스트림의 타이밍 드리프트 모니터링
 - NTP와 CDN 엣지 서버 시간 동기화 확인
3. SCTE-35 마커 구현 검증:
 - EXT-X-DATERANGE 태그에 적절한 SCTE35-OUT 및 DURATION 사양이 포함되어 있는지 확인
 - 명시적 큐인 타이밍을 사용할 때 페어링된 SCTE35-OUT 및 SCTE35-IN 마커 확인
 - START-DATE 타임스탬프가 실제 콘텐츠 타이밍과 일치하는지 확인
 - 다양한 SCTE-35 마커 형식 테스트(기간 기반 마커와 페어링된 마커)

4. 다양한 시나리오에서 광고 중단 타이밍을 테스트합니다.

- 광고 중단 타이밍을 직접 MediaTailor 요청과 CDN 요청 비교
- 다양한 CDN 엣지 로케이션에서 타이밍 일관성 테스트
- 다양한 플레이어 유형 및 버퍼링 동작으로 광고 중단 타이밍 확인
- 트래픽이 가장 많은 기간 동안 타이밍 정확도 모니터링

5. 로그 및 모니터링을 사용하여 타이밍 문제를 디버깅합니다.

- 디버그 로깅을 활성화하여 광고 중단 처리 타이밍 추적
- CloudWatch 지표에서 광고 삽입 타이밍 패턴 모니터링
- CDN 로그에서 타이밍 관련 요청 패턴 확인
- 플레이어 디버깅 도구를 사용하여 클라이언트 관점에서 광고 중단 타이밍 확인

예상 타이밍 허용치:

- 광고 중단 타이밍은 콘텐츠의 SCTE-35 마커와 일치해야 합니다.
- 광고 기간은 광고 결정 서버 응답에 지정된 기간과 일치해야 합니다.
- 콘텐츠 소스, MediaTailor 및 CDN 간의 클록 동기화는 1초 이내여야 합니다.
- SCTE-35 마커 타이밍은 실제 콘텐츠 타이밍의 100ms 이내로 정확해야 합니다.

추가 문제 해결 리소스:

- SCTE-35 마커 형식 및 구현은 섹션을 참조하세요. [MediaTailor 광고 삽입을 위한 콘텐츠 소스 통합](#)
- 디버그 로깅 설정 및 타이밍 분석은 섹션을 참조하세요. [AWS Elemental MediaTailor 디버그 로그 생성](#)
- CDN 캐싱 구성 및 타이밍 영향은 섹션을 참조하세요. [CDN 및 MediaTailor 통합을 위한 캐싱 최적화](#)
- 타이밍 검증을 포함한 포괄적인 테스트 절차는 섹션을 참조하세요. [CDN 및 MediaTailor 통합 테스트 및 검증](#)
- 광고 삽입 타이밍 및 성능 모니터링은 섹션을 참조하세요. [MediaTailor CDN 작업 및 성능 모니터링](#)

성공 기준: 해결되면 모든 CDN 엣지 로케이션 및 플레이어 유형에서 일관된 타이밍으로 SCTE-35 마커에 지정된 시간에 광고 시간이 정확하게 표시되어야 합니다. 디버그 로그에는 드리프트 또는 동기화 오류 없이 정확한 광고 중단 처리 타이밍이 표시되어야 합니다.

MediaTailor의 CDN 성능 최적화 및 지연 시간 문제 해결

AWS Elemental MediaTailor 콘텐츠 전송 네트워크(CDN) 통합 성능은 최종 사용자 경험과 광고 전송 품질에 직접적인 영향을 미칩니다. 응답 시간이 느리거나 성능이 저하되는 경우:

성능 측정 기법

성능 문제를 해결하기 전에 기존 측정 및 지속적인 모니터링을 설정합니다.

1. 주요 성능 지표 측정:

- 응답 시간: 매니페스트 요청은 200ms 이내에 완료되고, 세그먼트 요청은 100ms 이내에 완료되어야 합니다.
- 캐시 적중률: 콘텐츠 세그먼트 >95%, 광고 세그먼트 >90%
- 오리진 요청 볼륨: 캐시가 최적화될 때 총 요청의 5% 미만이어야 합니다.
- 첫 번째 프레임까지의 시간: 초기 재생이 2~3초 이내에 시작되어야 합니다.

2. 성능 측정 도구를 사용합니다.

- CDN 분석 대시보드: 캐시 성능, 응답 시간 및 오류율 모니터링
- CloudWatch 지표: GetManifest.Latency를 포함한 MediaTailor 서비스 지표 추적
- 브라우저 개발자 도구: 클라이언트 측 성능 및 네트워크 타이밍 측정
- 명령줄 도구: 타이밍 옵션과 함께 curl을 사용하여 특정 요청 측정

3. 지속적인 모니터링 구현:

- 응답 시간 저하에 대한 자동 성능 알림 설정
- 다양한 지리적 리전에서 성능 모니터링
- 트래픽이 가장 많은 기간 동안 성능 추적
- 구성 변경 전후의 성능 지표 비교

성능 측정 리소스:

- 포괄적인 성능 모니터링 설정은 섹션을 참조하세요. [MediaTailor CDN 작업 및 성능 모니터링](#)
- 성능 테스트 절차는 섹션을 참조하세요. [CDN 및 MediaTailor 통합 테스트 및 검증](#)
- CloudWatch 지표 및 모니터링은 섹션을 참조하세요. [Amazon CloudWatch 지표 AWS Elemental MediaTailor를 사용한 모니터링](#)

CDN 캐시 성능 문제

캐시 성능 문제는 가장 일반적인 CDN 통합 문제 중 하나입니다. 이러한 문제는 모든 MediaTailor 구현에 영향을 미치며 최종 사용자 경험과 비용에 상당한 영향을 미칠 수 있습니다.

낮은 캐시 적중률

증상: 오리진 요청 볼륨 높음, 지연 시간 증가, 대역폭 비용 증가, 시청자 경험 저하

대상 값:

- 콘텐츠 세그먼트: 캐시 적중률 95% 이상
- 광고 세그먼트: 캐시 적중률 90% 이상
- 매니페스트: 구현에 따라 다름(개인화된 매니페스트는 캐시해서는 안 됨)

일반적인 원인:

- 다양한 콘텐츠 유형에 대한 잘못된 TTL 설정
- 캐시 키 구성에 불필요한 쿼리 파라미터 포함
- 오리진의 캐시 제어 헤더가 제대로 구성되지 않음
- 빈번한 캐시 무효화 또는 제거
- 지리적 배포 문제(엣지 로케이션에 캐싱되지 않은 콘텐츠)

솔루션:

1. TTL 설정을 검토하고 최적화합니다.

- 콘텐츠 세그먼트: 세그먼트 기간 이상과 일치하도록 TTL 설정
- 광고 세그먼트: 재사용 가능한 광고의 경우 TTL을 24시간 이상으로 설정
- 정적 자산: TTL을 24시간 이상으로 설정

포괄적인 TTL 권장 사항 및 캐싱 최적화 전략은 [섹션을 참조하세요](#) [CDN 및 MediaTailor 통합을 위한 캐싱 최적화](#).

2. 캐시 키 구성 최적화:

- 캐시 키에서 불필요한 쿼리 파라미터 제거
- 콘텐츠에 영향을 미치는 파라미터만 포함되도록 합니다.
- 파라미터 순서 및 대/소문자 구분 정규화

3. 오리진 캐시 제어 헤더가 올바르게 설정되었는지 확인

4. 트래픽이 많은 구현을 위해 오리진 실드(또는 이에 상응하는 CDN 기능)를 구현합니다. 오리진 실드 기능은 주요 CDN에서 사용할 수 있지만 이름이 다를 수 있습니다(예: CloudFront Origin Shield, Fastly Shield, Cloudflare Argo 계층형 캐시). CDN에서이 기능을 제공하지 않는 경우 [AWS Support](#)에 문의할 때 MediaTailor에서 활성화할 수 있습니다.
5. 캐시 무효화 전략 검토 및 불필요한 제거 감소

검증 단계:

1. CDN 분석 대시보드를 사용하여 캐시 적중률 모니터링
2. curl을 사용하여 특정 URLs을 테스트하여 캐시 헤더 확인
3. 변경 전후의 오리진 요청 볼륨 비교

높은 오리진 요청 볼륨

증상: MediaTailor 오리진에 도달하는 요청 수가 예기치 않게 많음, 오리진 서버 로드 증가, 비용 증가

예상 패턴: 캐시 적중률이 최적일 때 오리진 요청은 총 최종 사용자 요청의 5% 미만이어야 합니다.

일반적인 원인:

- 낮은 TTL 값으로 인한 캐시 누락
- 캐시 키 조각화(고유 캐시 키가 너무 많음)
- 캐싱된 콘텐츠가 없는 리전의 지리적 트래픽 급증
- 빈번한 캐시 무효화

솔루션:

1. 요청 패턴을 분석하여 캐시 누락 원인 식별
2. 콘텐츠 유형 및 업데이트 빈도에 따라 TTL 설정 최적화
3. 새 콘텐츠에 대한 캐시 워밍 전략 구현
4. 오CDNs에서 사용 가능 - [Origin Shield 구현](#) 자세한 내용은 참조)

알림 임계값: 오리진 요청이 총 요청의 10%를 초과하거나 기존보다 50% 증가할 때 알림을 설정합니다.

일반적인 HTTP 오류 해결

CDN 통합의 HTTP 오류는 구성 문제 또는 서비스 문제를 나타내는 경우가 많습니다. 이러한 오류 패턴은 모든 MediaTailor 구현에서 일관됩니다.

404 찾을 수 없음 오류

증상: 매니페스트 또는 세그먼트 요청은 HTTP 404를 반환하고, 플레이어는 콘텐츠를 로드하지 못합니다. 플레이어 로그에서 "MANIFEST_LOAD_ERROR"

일반적인 원인:

- 잘못된 CDN 오리진 구성(잘못된 MediaTailor 엔드포인트 URL)
- 누락되거나 잘못된 캐시 동작 경로 패턴
- CDN 구성의 URL 재작성 문제
- MediaTailor 구성 이름 또는 재생 엔드포인트 오류
- 라이브 콘텐츠의 타이밍 문제(향후 세그먼트 요청)

진단 단계:

1. MediaTailor 오리진에 대해 직접 동일한 URL 테스트(CDN 우회)
2. CDN 오리진 구성이 MediaTailor 재생 엔드포인트와 일치하는지 확인
3. CDN 캐시 동작 경로 패턴 및 우선 순위 확인
4. CDN 액세스 로그에서 요청 라우팅 세부 정보 검토
5. MediaTailor 구성 이름 및 리전 설정 검증

솔루션:

- MediaTailor 재생 엔드포인트와 일치하도록 CDN 오리진 구성 수정
- 캐시 동작 경로 패턴을 업데이트하여 요청을 올바르게 라우팅합니다.
- 해당하는 경우 URL 재작성 규칙 수정
- MediaTailor 구성이 존재하고 활성 상태인지 확인

403 금지 오류

증상: 요청은 HTTP 403, 액세스 거부 메시지, 인증 실패를 반환합니다.

일반적인 원인:

- MediaTailor에 필요한 쿼리 파라미터가 누락되었거나 잘못되었습니다.
- CDN이 필수 헤더 또는 파라미터를 전달하지 않음
- IP 주소 제한 또는 지리적 차단
- 인증 토큰 문제(서명URLs을 사용하는 경우)

솔루션:

- 모든 필수 쿼리 파라미터가 포함되고 전달되었는지 확인
- CDN 구성에서 헤더 및 파라미터 전달 확인
- IP 제한 및 지리적 설정 검토
- 인증 토큰 및 서명 프로세스 검증

400 잘못된 요청 오류

증상: 요청에서 HTTP 400 반환, 잘못된 요청 오류, 파라미터 검증 실패

일반적인 원인:

- 잘못된 쿼리 파라미터 또는 URL 인코딩 문제
- 잘못된 파라미터 값 또는 형식
- 특정 MediaTailor 기능에 필요한 파라미터 누락
- URL 길이 제한 초과

솔루션:

- 쿼리 파라미터 형식 및 URL 인코딩 검증
- MediaTailor API 요구 사항과 비교하여 파라미터 값 확인
- 모든 필수 파라미터가 포함되어 있는지 확인합니다.
- URL 길이 검토 및 파라미터 최적화 고려

5xx 서버 오류

증상: 요청에서 HTTP 500, 502, 503 또는 504 오류, 간헐적인 서비스 장애 반환

일반적인 원인:

- MediaTailor 서비스 문제 또는 용량 제한
- CDN 오리진 연결 문제
- 느린 오리진 응답으로 인한 제한 시간 문제
- 임시 서비스 성능 저하

솔루션:

- AWS 서비스 상태 대시보드에서 MediaTailor 서비스 상태 확인

- CDN 오리진 연결 및 제한 시간 설정 확인
- 지수 백오프를 사용하여 재시도 로직 구현
- MediaTailor CloudWatch 지표의 서비스 상태 모니터링
- 문제가 지속되면 AWS Support에 문의하세요.

1. 기준 성능 측정:

- MediaTailor에 직접 매니페스트 요청 응답 시간 테스트(대상: <200ms)
- 매니페스트 요청에 대한 CDN 응답 시간 측정(대상: 캐시 적중의 경우 <100ms)
- 오리진과 CDN 모두에서 세그먼트 로드 시간 확인

2. CDN 성능 분석:

- 콘텐츠 세그먼트의 캐시 적중률 확인(대상: 인기 콘텐츠의 경우 >80%)
- 오리진 실드(또는 이에 상응하는 CDN 기능)가 오리진과 동일한 AWS 리전에서 활성화되고 구성되어 있는지 확인합니다. 이 기능에는 서로 다른 CDN들이 서로 다른 이름을 사용합니다.
- CDN 엣지 로케이션 성능 및 지리적 분포 모니터링

성능 벤치마크:

- 매니페스트 생성 응답 시간을 모니터링하고 기준 성능과 비교
- CDN 캐시 적중이 오리진 요청보다 훨씬 빠름
- ADS 응답 시간으로 인해 매니페스트 생성이 지연되어서는 안 됩니다.

추가 문제 해결 리소스:

- 포괄적인 성능 최적화 전략은 섹션을 참조하세요. [CDN 및 MediaTailor 통합을 위한 성능 최적화 가이드](#)
- 오리진 실드 구현 세부 정보는 섹션을 참조하세요. [Origin Shield 구현](#)
- CDN 캐싱 최적화는 섹션을 참조하세요. [CDN 및 MediaTailor 통합을 위한 캐싱 최적화](#)
- 성능 모니터링 및 지표는 섹션을 참조하세요. [MediaTailor CDN 작업 및 성능 모니터링](#)
- 성능 테스트 절차는 섹션을 참조하세요. [CDN 및 MediaTailor 통합 테스트 및 검증](#)

성공 기준: 해결되면 응답 시간은 대상 벤치마크(매니페스트 200ms 미만, 세그먼트 100ms 미만)를 충족해야 하고, 캐시 적중률은 대부분의 콘텐츠 유형에 대해 90%를 초과해야 하며, 오리진 요청 볼륨은 총 요청의 5% 미만이어야 합니다. 성능은 모든 지리적 리전 및 디바이스 유형에서 일관되어야 합니다.

MediaTailor의 디바이스 및 플랫폼 간 CDN 불일치 동작 수정

AWS Elemental MediaTailor 콘텐츠 전송 네트워크(CDN) 통합은 모든 디바이스 및 플랫폼에서 일관된 광고 전송을 제공해야 합니다. 디바이스 간에 광고가 다르게 동작하는 경우:

1. 모든 CDN 동작에서 일관된 헤더 전달을 보장합니다.
 - User-Agent, X-Forwarded-For 및 사용자 지정 대상 지정 헤더가 일관되게 전달되는지 확인
 - 헤더 전달 규칙이 모든 관련 캐시 동작에 적용되는지 확인
2. 플레이어와 CDN 구성의 호환성을 확인합니다.
 - 여러 플레이어 유형(HLS.js, Video.js, 네이티브 플레이어)으로 테스트하여 플레이어별 문제 식별
 - 플레이어별 헤더 요구 사항 또는 URL 처리 차이점 확인
3. 여러 디바이스 유형으로 테스트하여 디바이스별 문제를 식별합니다.
 - 테스트에 모바일 디바이스, 태블릿, 스마트 TVs 및 데스크톱 브라우저 포함
 - 다양한 운영 체제 및 브라우저 버전 테스트
 - 디바이스별 광고 타겟팅이 올바르게 작동하는지 확인

이러한 문제 해결 단계를 따랐지만 여전히 도움이 필요한 경우 섹션을 참조하세요 [CDN 통합 지원 받기](#).

문제 해결 준비

CDN 통합 문제가 발생할 때 문제 해결을 간소화하는 도구 및 프로세스를 설정합니다. 사전 대비를 통해 문제가 발생할 때 문제를 더 빠르고 효과적으로 해결할 수 있습니다.

포괄적인 로깅 활성화

CDN 통합 문제를 진단하려면 세부 로그가 필수적입니다. 문제 해결 중에 필요한 정보를 캡처하도록 로깅을 구성합니다.

1. 세부 CDN 액세스 로그 활성화:
 - MediaTailor 요청을 처리하는 모든 캐시 동작에 대한 로깅 구성
 - 로그 항목에 쿼리 문자열 및 사용자 지정 헤더 포함
 - 패턴과 이상을 식별하기 위한 로그 분석 도구 설정
 - 실시간 이벤트 중 즉각적인 문제 감지를 위한 실시간 로그 활성화
 - 기록 문제 해결 데이터를 유지 관리하도록 로그 보존 정책 구성

2. MediaTailor 로깅 구성:

- MediaTailor 구성에 대한 액세스 로그 활성화
- 중앙 집중식 로그 관리를 위한 CloudWatch 로그 그룹 설정
- 오류 패턴을 식별하도록 로그 필터 구성

3. 오리진 서버 로깅 설정:

- 콘텐츠 오리진 서버에서 세부 액세스 로그 활성화
- 로그에 요청 헤더 및 응답 코드 포함
- 오리진 서버 성능 지표 모니터링

진단 요청 헤더 추가

사용자 지정 헤더는 CDN을 통해 요청을 추적하고 라우팅 문제를 식별하는 데 도움이 됩니다.

1. CDN 진단 헤더 구성:

- 각 요청에 고유 식별자 추가(예: X-Request-ID)
- 요청 헤더에 CDN 관련 정보 포함
- 엣지 로케이션 또는 POP(존재 지점) 정보를 추가하여 지리적 라우팅 추적
- 캐시 동작 분석을 위한 캐시 상태 헤더(Hit, Miss, RefreshHit) 포함

2. 디버깅을 위한 응답 헤더 추가:

- 서버 식별 헤더 포함
- 성능 분석을 위한 타이밍 정보 추가
- 매니페스트 요청에 대한 캐시 제어 헤더 포함

기준 성능 지표 설정

정상 성능 범위를 문서화하여 문제 해결 중에 이상을 빠르게 식별합니다.

1. 기준 지표를 기록합니다.

- 다양한 콘텐츠 유형에 대한 캐시 적중률
- 응답 시간 백분위수(P50, P95, P99)
- 상태 코드별 오류 발생률
- 시간대별 요청 볼륨 패턴

2. 성능 기대치 문서화:

- 대상 캐시 적중률(콘텐츠의 경우 95% 이상, 광고의 경우 90% 이상)
 - 허용 가능한 응답 시간(캐싱된 <100ms, 오리진 <500ms)
 - 허용되는 최대 오류율(4xx의 경우 <1%, 5xx의 경우 <0.1%)
3. 성능 대시보드 생성: 기준 값과 비교하여 현재 지표를 표시하는 모니터링 대시보드를 설정합니다.

문제 해결 도구 준비

효과적인 문제 해결에 필요한 도구 및 액세스 권한을 설정합니다.

1. 명령줄 도구:

- curl HTTP 요청 및 응답 테스트용
- dig DNS 문제 해결을 nslookup 위한 또는
- 매니페스트 확인을 위한 HLS/DASH 검증 도구
- 로그 분석 도구(grep, awk 또는 특수 로그 분석기)

2. 액세스 권한:

- 구성 검토를 위한 CDN 관리 콘솔 액세스
- 구성 확인을 위한 MediaTailor 콘솔 액세스
- 지표 및 로그 분석을 위한 CloudWatch 액세스
- 백엔드 문제 해결을 위한 오리진 서버 액세스

3. 설명서:

- 네트워크 아키텍처 다이어그램
- CDN 및 MediaTailor 구성 설명서
- 에스컬레이션 절차를 위한 연락처 정보
- 일반적인 시나리오에 대한 런북 문제 해결

워크플로별 문제 해결 가이드

이 범용 문제 해결 가이드에서는 모든 MediaTailor CDN 통합의 일반적인 문제를 다룹니다. 특정 워크플로 또는 서비스와 관련된 문제는 다음 특수 문제 해결 리소스를 참조하세요.

서버 측 광고 삽입(SSAI) 문제 해결

광고 삽입 실패, 광고 전환 문제 및 개인화 문제를 포함한 SSAI 관련 문제는 워크플로별 SSAI 문제 해결 설명서를 참조하세요.

일반적인 SSAI 관련 문제:

- 광고 삽입 실패 및 빈 광고 중단
- 광고 전환 타이밍 및 동기화 문제
- 개인화 및 대상 지정 문제
- 광고 추적 및 분석 불일치

채널 어셈블리 문제 해결

매니페스트 생성 문제 및 시간 전환 기능을 포함한 채널 어셈블리 관련 문제는 채널 어셈블리 워크폴로 설명서를 참조하세요.

일반적인 채널 어셈블리 문제:

- 매니페스트 생성 및 컴파일 오류
- 시간 전환 기간 및 DVR 기능 문제
- 소스 콘텐츠 가용성 및 장애 조치 문제
- 프로그램 일정 및 메타데이터 동기화

MediaPackage 통합 문제 해결

매니페스트 필터링 및 EMP 엔드포인트 문제를 포함한 MediaPackage 관련 문제는 섹션을 참조하세요 [CDN 통합 문제 해결](#).

일반적인 MediaPackage 통합 문제:

- 매니페스트 필터링 파라미터 오류
- MediaPackage 엔드포인트 연결 문제
- EMP별 캐시 동작 문제
- MediaPackage 오리진 인증 문제

CloudFront 관련 문제 해결

CloudFront별 구성 및 설정 문제는 CloudFront 통합 설명서를 참조하세요.

일반적인 CloudFront 문제:

- 배포 구성 및 캐시 동작 설정
- 오리진 액세스 ID 및 보안 구성
- CloudFront 관련 오류 코드 및 응답

- 지리적 제한 및 엣지 로케이션 문제

추가 리소스:

- 성능 최적화 지침은 섹션을 참조하세요. [CDN 성능 최적화](#)
- 설정 모니터링 및 알림은 섹션을 참조하세요. [CDN 모니터링](#)
- 일반적인 지원 및 지원은 섹션을 참조하세요. [CDN 통합 지원 받기](#)

MediaTailor에 대한 CDN 통합 로그 분석 및 오류 코드 참조

AWS Elemental MediaTailor 콘텐츠 전송 네트워크(CDN) 통합 로그는 성능 및 오류에 대한 중요한 인사이트를 제공합니다. 이 가이드에서는 CDN 통합 문제 해결과 관련된 CDN 로그(콘텐츠 전송 네트워크의 로그)와 MediaTailor 로그 및 오류 코드를 모두 다룹니다. 콘텐츠 전송 네트워크 로그와 오류 코드가 MediaTailor 통합에 대해 알려주는 내용을 이해해야 하는 경우가 많으므로 참조를 사용합니다. 이 가이드는 로그 항목과 오류 메시지를 해석하여 문제의 근본 원인을 식별하는 데 도움이 됩니다.

관련 주제:

- 단계별 문제 해결 절차는 섹션을 step-by-step [CDN 통합 문제 해결](#)
- 사전 모니터링 및 예방 전략은 섹션을 참조하세요. [MediaTailor CDN 작업 및 성능 모니터링](#)
- 에스컬레이션 및 추가 도움말은 섹션을 참조하세요. [CDN 통합 지원 받기](#)
- 포괄적인 MediaTailor 로깅 구성 및 리소스는 섹션을 참조하세요. [MediaTailor 로깅 구성 리소스](#)
- CloudFront 로그 형식 참조는 [CloudFront 액세스 로그 형식](#)을 참조하세요.
- HTTP 상태 코드 참조는 [HTTP 응답 상태 코드](#)를 참조하세요.

MediaTailor에 대한 CDN 로그 해석 및 분석 기법

AWS Elemental MediaTailor 콘텐츠 전송 네트워크(CDN) 통합은 요청 패턴을 이해하고 문제를 식별하는 데 도움이 되는 세부 로그를 생성합니다. 콘텐츠 전송 네트워크 로그를 분석할 때는 다음 주요 지표에 집중하세요.

참고: 다음 예제에서는 Amazon CloudFront 용어를 사용합니다. 다른 CDN 공급자는 유사한 개념에 다른 용어를 사용할 수 있습니다.

HTTP 상태 코드

- 200 - 요청 성공

- 404 - 리소스를 찾을 수 없음(라우팅 규칙 및 MediaTailor 구성 확인)
- 403 - 액세스 거부됨(CDN 보안 설정 및 오리진 권한 확인)
- 502/503/504 - 오리진 서버 오류(MediaTailor 서비스 상태 및 오리진 연결 확인)

캐시 동작 지표

- Hit - CDN 캐시에서 제공되는 콘텐츠
- Miss - 오리진에서 가져온 콘텐츠
- RefreshHit - 오리진으로 검증된 캐시된 콘텐츠
- Error - 요청에 오류가 발생했습니다.

모니터링할 요청 패턴

- 매니페스트 요청은 일반적으로 낮은 TTL 설정RefreshHit으로 인해 Miss 또는를 초래해야 합니다.
- 인기 콘텐츠Hit에 대한 콘텐츠 세그먼트가 표시되어야 합니다.
- 개인 맞춤Miss으로 인해 광고 세그먼트가 표시될 수 있습니다.

서비스 상태 모니터링

CDN 로그를 분석하고 문제를 해결할 때 MediaTailor 서비스 상태를 확인하여 문제가 서비스와 관련이 있는지 확인합니다.

AWS 서비스 상태 대시보드

현재 MediaTailor 서비스 상태 및 진행 중인 서비스 이벤트 확인

액세스: [AWS 서비스 상태 대시보드](#)

이 경우 사용: CDN 로그에 광범위한 5xx 오류 또는 서비스 제한 시간이 표시됩니다.

AWS 개인 상태 대시보드

계정별 서비스 상태 알림 및 유지 관리 이벤트 보기

액세스: [AWS Personal Health Dashboard](#)

이 경우 사용: 계정별 서비스 상태 정보 또는 유지 관리 알림이 필요합니다.

Amazon CloudWatch 지표 모니터링

CloudWatch 지표를 통해 MediaTailor 서비스 상태 지표를 모니터링합니다.

- `GetManifest.Errors` - 매니페스트 생성 오류 추적
- `GetManifest.Latency` - 응답 시간 성능 모니터링
- `AdDecisionServer.Errors` - 광고 서버 연결 문제 모니터링
- `Origin.Errors` - 오리진 서버 연결 문제 추적

액세스: [CloudWatch 콘솔](#)

이 경우 사용: 자세한 서비스 성능 지표와 과거 추세가 필요합니다.

서비스 상태 문제 해결 워크플로:

1. AWS 서비스 상태 대시보드에서 현재 서비스 상태 확인
2. 계정별 알림에 대한 AWS Personal Health Dashboard 검토
3. CloudWatch 지표에서 서비스 성능 지표 모니터링
4. 서비스 상태를 CDN 로그 패턴 및 오류율과 상호 연관시킵니다.
5. 서비스 상태가 정상인 경우 CDN 구성 및 오리진 서버 문제에 집중합니다.

CDN 로그 분석 리소스

- [CloudFront 액세스 로그](#) - CDN 로그 형식 및 필드에 대한 전체 가이드
- [CloudFront 실시간 로그](#) - 실시간 로그 스트리밍 및 분석
- [CloudWatch Logs Insights를 사용한 로그 데이터 분석](#) - 고급 로그 쿼리 및 분석
- [워크로드 리소스 모니터링](#) - Well-Architected 모니터링 패턴

주제

- [MediaTailor 오류 코드 및 CDN 통합 문제 해결](#)
- [MediaTailor에 대한 CDN 로그 분석 도구 및 모니터링 기법](#)
- [MediaTailor 로깅 구성 리소스](#)

MediaTailor 오류 코드 및 CDN 통합 문제 해결

AWS Elemental MediaTailor 오류 코드는 콘텐츠 전송 네트워크(CDN)를 통해 전달될 때 통합 문제에 대한 특정 정보를 제공합니다. MediaTailor가 콘텐츠 전송 네트워크를 통해 오류를 반환하면 이러한 코드는 특정 문제를 나타냅니다.

참고: 이러한 오류 코드는 MediaTailor에서 반환되며 CDN을 통해 전달됩니다. CDN 관련 오류는 CDN 공급자의 설명서를 참조하세요.

400 잘못된 요청

일반적인 원인: 잘못된 쿼리 파라미터, 필수 파라미터 누락, 잘못된 세션 IDs

확인: CDN 쿼리 문자열 전달 구성, 파라미터 인코딩

403 금지됨

일반적인 원인: 존재하지 않거나 액세스 권한이 없는 세그먼트를 요청하는 클라이언트 플레이어입니다. 광고 세그먼트의 경우 MediaTailor가 존재하지 않는 세그먼트를 지정했을 수 있습니다(조사하려면 [AWS Support](#)에 문의). 콘텐츠 세그먼트의 경우 오리진 공급자 액세스 제한 또는 인증 문제

확인: 광고 세그먼트의 경우: MediaTailor가 잘못된 세그먼트 URL을 생성하는 경우 [AWS Support](#)에 문의하세요. URLs 콘텐츠 세그먼트의 경우: 오리진 서버 권한, 인증 자격 증명 및 액세스 정책을 확인합니다. 요청된 URLs에 세그먼트가 있는지 확인

404 Not Found(404 찾을 수 없음)

일반적인 원인: MediaTailor 구성이 존재하지 않거나 비활성 상태임, 잘못된 재생 URL 경로, 존재하지 않는 리소스에 대한 매니페스트 또는 세그먼트 요청

확인: MediaTailor 구성이 존재하고 활성 상태인지 확인하고, 재생 URL이 ManifestEndpointPrefix의와 일치하는지 확인하고GetPlaybackConfiguration, CDN 라우팅 규칙 전달 요청을 확인하여 MediaTailor 엔드포인트 수정

500 Internal Server Error

일반적인 원인: 오리진 서버 문제, ADS 연결 문제, 매니페스트 처리 오류

확인: 오리진 서버 상태, ADS 응답 유효성, MediaTailor 서비스 상태

502 잘못된 게이트웨이

일반적인 원인: HLS 재생 목록 정렬 문제로 UnsupportedManifestException 인해 오리진 서버에 연결할 수 없고 오리진 응답이 잘못되었습니다(재생 목록 간에 SCTE 마커가 정렬되지 않음, 일부 재생 목록에서 SCTE 마커가 누락됨, 비트 전송률 변형 간에 광고 중단 타이밍이 일관되지 않음).

확인: 오리진 서버 연결 및 방화벽 규칙, DNS 확인, 모든 비트 전송률 변형의 HLS 재생 목록 일관성, 모든 재생 목록의 동일한 세그먼트에 대한 SCTE-35 마커 정렬, 모든 재생 목록에 필수 광고 중단 마커가 포함되어 있는지 확인

오류 코드 분석 리소스:

- [MediaTailor 문제 해결](#) - MediaTailor 오류 코드 참조 완료
- [CloudFront 오류 응답 문제 해결](#) - CDN 관련 오류 분석
- [HTTP 응답 상태 코드](#) - 포괄적인 HTTP 상태 코드 참조
- [장애 관리](#) - Well-Architected 오류 처리 패턴

MediaTailor에 대한 CDN 로그 분석 도구 및 모니터링 기법

AWS Elemental MediaTailor 콘텐츠 전송 네트워크(CDN) 통합은 효율적인 분석 도구 및 기술이 필요한 대량의 로그 데이터를 생성합니다. 다음 접근 방식을 사용하여 콘텐츠 전송 네트워크 및 MediaTailor 로그를 효율적으로 분석합니다.

- 명령줄 분석: grep, awk 및와 같은 도구를 사용하여 로그 패턴 sort 필터링 및 분석
- Amazon CloudWatch Logs Insights: 고급 분석을 위해 SQL과 유사한 구문으로 CDN 및 MediaTailor 로그 쿼리
- 타사 도구: 포괄적인 모니터링 및 알림을 위해 로그 분석 플랫폼 고려
- 사용자 지정 대시보드: CDN 지표를 MediaTailor 성능 데이터와 결합하는 시각화 생성

로그 분석 또는 복잡한 오류 패턴 해석에 대한 추가 지원이 필요한 경우 섹션을 참조하세요 [CDN 통합 지원 받기](#).

로그 분석 도구 및 리소스:

- [CloudWatch Logs Insights](#) - 로그 분석을 위한 SQL과 유사한 쿼리
- [Amazon OpenSearch Service](#) - 고급 로그 검색 및 분석
- [CloudWatch 대시보드](#) - 사용자 지정 시각화 및 모니터링
- [운영 가시성을 위한 분산 시스템 계측](#) - 고급 관찰성 패턴
- [워크로드 관찰성 설계](#) - Well-Architected 관찰성 지침

MediaTailor 로깅 구성 리소스

MediaTailor는 CDN 로그 외에도 광고 삽입, 매니페스트 생성 및 서비스 상호 작용을 모니터링하기 위한 포괄적인 로깅 기능을 제공합니다. 다음 리소스를 사용하여 MediaTailor 로그를 구성하고 분석합니다.

MediaTailor 매니페스트 및 서비스 로그

매니페스트 생성, 오리진 상호 작용 및 서비스 이벤트에 대한 로그를 구성하고 분석합니다.

- [AWS Elemental MediaTailor 매니페스트 로그 설명 및 이벤트 유형](#) - MediaTailor 매니페스트 로그 및 이벤트 유형에 대한 전체 참조
- [AWS Elemental MediaTailor 로그 보기](#) - MediaTailor 로그 보기 및 해석 가이드

판매 로그 구성

비용 최적화를 통해 여러 대상으로 유연한 로그 전송을 구성합니다.

- [벤딩된 로그를 사용하여 AWS Elemental MediaTailor 로그 전송](#) - Amazon S3, Firehose 또는 CloudWatch Logs로 로그 전송 구성
- [AWS Elemental MediaTailor 로깅 전략 마이그레이션](#) - 기존 로깅 구성에 대한 마이그레이션 가이드

CloudWatch Logs 통합

모니터링 및 분석을 위해 MediaTailor 로그를 CloudWatch와 통합합니다.

- [AWS Elemental MediaTailor 로그 보기](#) - CloudWatch Logs 구성 및 분석
- [CloudWatchInsights](#) - 고급 로그 쿼리 및 분석

광고별 로깅

광고 삽입 성능을 모니터링하고 광고 관련 문제를 해결합니다.

- ADS 상호 작용 로그 - 광고 결정 서버 통신 및 오류 모니터링

CDN과 MediaTailor 로그의 주요 차이점:

- CDN 로그: 콘텐츠 전송 네트워크의 요청/응답 패턴, 캐시 동작 및 네트워크 수준 오류 표시
- MediaTailor 로그: 광고 삽입 세부 정보, 매니페스트 생성 이벤트, 오리진 상호 작용 및 서비스별 오류 표시
- 통합 분석: 두 로그 유형을 함께 사용하여 스트리밍 워크플로를 완벽하게 파악할 수 있습니다.

CDN 및 MediaTailor 로깅을 모두 결합하는 포괄적인 모니터링은 섹션을 참조하세요 [MediaTailor CDN 작업 및 성능 모니터링](#).

를 사용하여 MediaTailor 및 CDN 자동화 CloudFormation

를 사용하여 콘텐츠 전송 네트워크(CDN) AWS Elemental MediaTailor 를 자동화하면 광고 삽입 워크플로가 AWS CloudFormation 간소화되고 안정성이 향상됩니다.

이 섹션에서는 AWS CloudFormation (코드 서비스형AWS 인프라)를 사용하여 콘텐츠 전송 네트워크(CDN) AWS Elemental MediaTailor 로 자동으로 설정하는 방법을 보여줍니다. 이전 섹션에 설명된 대로 이 통합을 수동으로 구성할 수 있지만 를 사용하면 단일 템플릿으로 전체 프로세스를 자동화하여 시간을 CloudFormation 절약하고 오류를 줄일 수 있습니다.

를 처음 사용하는 경우 필요한 모든 AWS 리소스를 정의하는 템플릿 파일을 생성할 수 있는 서비스 CloudFormation입니다. 이 템플릿을 배포하면 해당 리소스를 CloudFormation 자동으로 생성하고 구성하여 리소스가 올바르게 연동되도록 합니다.

에 대한 자세한 내용은 [CloudFormation 사용 설명서를](#) CloudFormation참조하세요. 의 MediaTailor 리소스 유형에 대한 자세한 CloudFormation내용은 [AWS::MediaTailor 리소스 유형 참조를 참조하세요](#).

주제

- [MediaTailor 및 CDN 통합 CloudFormation 예를 사용하는 이유](#)
- [CDN 및 MediaTailor 통합 CloudFormation 배포 준비](#)
- [CDN 및 MediaTailor 통합을 위한 CloudFormation 템플릿 배포](#)
- [CDN 및 MediaTailor 통합에 CloudFormation 배포된 리소스 사용](#)
- [CDN 및 MediaTailor 통합을 위한 CloudFormation 배포 테스트 및 검증](#)
- [CDN 및 MediaTailor 통합의 일반적인 CloudFormation 배포 문제 해결](#)
- [CDN 및 MediaTailor 통합을 위한 CloudFormation 템플릿 사용자 지정](#)
- [AWS CloudFormation AWS Elemental MediaTailor 및 Amazon CloudFront 통합에 대한 템플릿 참조](#)

MediaTailor 및 CDN 통합 CloudFormation 예를 사용하는 이유

AWS Elemental MediaTailor 를 사용한 자동화 AWS CloudFormation 는 스트리밍 워크플로를 관리하는 브로드캐스트 전문가에게 상당한 이점을 제공합니다. 콘텐츠 전송 네트워크(CDN)를 사용하여 MediaTailor를 수동으로 구성하면 시간이 많이 걸리고 오류가 발생하기 쉽습니다. CloudFormation 자동화를 사용하면 다음과 같은 이점이 있습니다.

- 일관성: 매번 동일한 구성이 배포되도록 하여 인적 오류를 줄입니다.

- 버전 관리: 변경 사항을 추적하기 위해 인프라를 버전 관리 시스템에 코드로 저장합니다.
- 빠른 배포: 수동 구성 시간이 아닌 몇 분 만에 복잡한 구성을 배포합니다.
- 환경 복제: 개발, 테스트 및 프로덕션 환경 전반에서 구성을 쉽게 복제합니다.
- 설명서: 템플릿 자체는 인프라에 대한 설명서 역할을 합니다.

자동화된 워크플로를 수동 구성과 비교하는 방법은 다음과 같습니다.

수동 설정(여러 단계)	자동 설정(단일 템플릿)
MediaTailor 재생 구성 생성	파라미터가 있는 CloudFormation 템플릿 1개 배포
CloudFront 배포 생성	
캐시 동작 구성	
보안 구성 설정	

CloudFront를 사용하여 MediaTailor를 설정하는 자동화된 워크플로는 다음 단계를 따릅니다.

1. 콘텐츠 오리진 및 광고 서버 파라미터와 함께 CloudFormation 템플릿 배포
2. CloudFormation 는 필요한 모든 리소스를 생성하고 구성합니다.
 - 광고 삽입을 위한 MediaTailor 재생 구성
 - 적절한 캐시 동작이 있는 CloudFront 배포
 - 콘텐츠 보호를 위한 보안 구성
3. CloudFormation 출력을 사용하여 광고 지원 스트림 URLs 액세스
4. 동적으로 삽입된 광고로 콘텐츠 스트리밍

CDN 및 MediaTailor 통합 CloudFormation 배포 준비

AWS Elemental MediaTailor 를 사용한 배포에는 특정 사전 조건 및 준비 단계가 AWS CloudFormation 필요합니다. MediaTailor와 Amazon CloudFront를 통합 CloudFormation 하기 위해 로 작업을 시작하기 전에 다음이 있는지 확인합니다.

- MediaTailor, CloudFront 및 CloudFormation 리소스를 생성할 수 있는 권한이 있는 AWS 계정

- 비디오 콘텐츠가 호스팅되는 콘텐츠 오리진(예: AWS Elemental MediaPackage Amazon S3 또는 다른 오리진 서버)
- VAST 요청에 응답할 수 있는 광고 결정 서버(ADS)

CloudFormation 템플릿을 배포하기 전에 다음과 같은 필수 파라미터를 수집합니다.

AdServerUrl

동적 광고 삽입을 위한 VAST 광고 서버의 URL입니다. 테스트를 위해 정적 VAST 엔드포인트가 제공됩니다.

ContentOriginDomainName

프로토콜이 없는 콘텐츠 오리진의 도메인 이름(예: *mediapackage-domain.mediapackagev2.us-west-2.amazonaws.com*, *mybucket.s3.amazonaws.com* 또는 *custom-origin.example.com*). http:// 또는 https:// 접두사 또는 경로를 포함하지 마십시오.

ContentOriginType

콘텐츠 오리진의 유형:

- mediapackagev2: AWS Elemental MediaPackage 오리진의 경우
- s3: Amazon S3 버킷 오리진의 경우
- 사용자 지정: 다른 오리진 유형의 경우

템플릿은 개인화된 광고로 콘텐츠를 제공하기 위해 함께 작동하는 여러 AWS 리소스를 생성합니다. 다음은 각 구성 요소가 수행하는 작업을 설명합니다.

오리진 액세스 제어

오리진 액세스 제어(OAC)는 오리진 서버에서 직접 액세스하지 않고 CloudFront를 통해서만 콘텐츠에 액세스할 수 있도록 하는 보안 기능입니다. 이렇게 하면 무단 액세스로부터 콘텐츠를 보호할 수 있습니다.

MediaPackage 및 Amazon S3 오리진의 경우 템플릿은 콘텐츠에 대한 액세스를 보호하기 위한 오리진 액세스 제어(OAC) 리소스를 생성합니다.

MediaTailor 재생 구성

MediaTailor 재생 구성은 광고 삽입을 처리하는 핵심 구성 요소입니다. 오리진에서 콘텐츠를 수신하고, 광고 서버에서 광고를 요청하고, 이를 각 뷰어에 대한 개인화된 스트림으로 결합합니다.

템플릿은 다음 설정을 사용하여 MediaTailor 재생 구성을 생성합니다.

- CloudFront 배포를 가리키는 비디오 콘텐츠 소스
- 지정된 VAST 엔드포인트에 구성된 광고 결정 서버 URL
- 라이브 스트림 중 광고 삽입을 위한 라이브 프리롤 구성
- 적절한 세그먼트 URL 접두사가 있는 CDN 구성

CloudFront 배포

CloudFront 배포는 짧은 지연 시간으로 전 세계 시청자에게 콘텐츠를 제공합니다. 다양한 유형의 요청(매니페스트, 콘텐츠 세그먼트, 광고 세그먼트)을 처리하고 적절한 오리진으로 라우팅합니다.

CDNs을 처음 사용하는 브로드캐스트 전문가의 경우 몇 가지 주요 용어는 다음과 같습니다.

Origin

원본 콘텐츠가 저장되는 서버(예: MediaPackage 또는 Amazon S3)

캐시 동작

다양한 유형의 콘텐츠를 캐싱하고 전송하는 방법을 결정하는 규칙

캐시 정책

콘텐츠가 캐시되는 기간과 캐싱에 영향을 미치는 요청 구성 요소를 제어하는 설정

템플릿은 다음 구성 요소를 사용하여 CloudFront 배포를 생성합니다.

- 세 가지 오리진:
 - 콘텐츠 오리진(MediaPackage, Amazon S3 또는 사용자 지정)
 - MediaTailor 매니페스트 오리진
 - MediaTailor 세그먼트 오리진
- 적절한 패턴으로 동작을 캐시합니다.
 - 콘텐츠 세그먼트의 기본 동작

- MediaTailor 광고 세그먼트에 대한 동작(/tm/*)
- MediaTailor 중간 미디어(/v1/i-media/*)에 대한 동작
- 개인화된 매니페스트에 대한 동작(/v1/*)
- 세그먼트 리디렉션 요청에 대한 동작(/segment/*)
- 각 동작에 최적화된 캐시 정책:
 - CachingOptimized 캐시 가능한 콘텐츠의 경우
 - CachingDisabled 개인화된 매니페스트용
- 적절한 헤더 전달을 보장하기 위한 오리진 요청 정책
- CORS 지원을 위한 응답 헤더 정책

CDN 및 MediaTailor 통합을 위한 CloudFormation 템플릿 배포

AWS Elemental MediaTailor AWS CloudFormation 템플릿을 사용한 배포는 템플릿이 생성할 내용을 이해하면 간단합니다. 이 프로세스는 약 15~30분이 소요되며, CloudFront 배포 배포를 기다리는 데 대부분의 시간이 소요됩니다.

CloudFormation 템플릿을 배포하고 자동 광고 삽입 워크플로를 설정하려면:

MediaTailor CloudFormation 템플릿을 배포하려면

1. AWS Elemental MediaTailor GitHub 리포지토리에서 CloudFormation 템플릿을 다운로드하거나에서 복사합니다 [AWS CloudFormation AWS Elemental MediaTailor 및 Amazon CloudFront 통합에 대한 템플릿 참조](#).
2. [CloudFormation 콘솔](#)을 엽니다.
3. 스택 생성 > 새 리소스 사용(표준)을 선택합니다.
4. 템플릿 지정에서 템플릿 파일 업로드를 선택하고 템플릿을 업로드합니다.
5. 스택 이름을 입력하고 필요한 파라미터의 값을 입력합니다.
 - AdServerUrl: VAST 광고 서버의 URL(예: <https://your-ad-server.com/vast>)
 - ContentOriginDomainName: 프로토콜이 없는 콘텐츠 오리진의 도메인 이름(예: mediapackage-domain.mediapackagev2.us-west-2.amazonaws.com)
 - ContentOriginType: 콘텐츠 오리진 유형을 선택합니다.
 - mediapackagev2: 오리진의 경우 AWS Elemental MediaPackage
 - s3: Amazon S3 버킷 오리진의 경우

- 사용자 지정: 다른 오리진 유형의 경우
6. 구성을 검토하고 스택 생성을 선택합니다.
 7. 스택 생성이 완료될 때까지 기다립니다. 일반적으로 5~10분이 걸립니다. 이벤트 탭에서 진행 상황을 모니터링할 수 있습니다.
 8. 완료되면 출력 탭으로 이동하여 HLS 및 DASH 매니페스트의 URLs을 찾습니다.

Note

를 콘텐츠 오리진 AWS Elemental MediaPackage 으로 사용하는 경우 MediaPackage 엔드 포인트가 올바르게 구성되고 액세스할 수 있는지 확인합니다. 자세한 내용은 [MediaPackage CDN 통합](#) 단원을 참조하십시오.

CDN 및 MediaTailor 통합에 CloudFormation 배포된 리소스 사용

AWS Elemental MediaTailor AWS CloudFormation 스택에서 배포한 리소스는 광고 삽입을 통해 콘텐츠에 액세스하는 데 사용할 몇 가지 중요한 출력을 제공합니다. CloudFormation 스택이 성공적으로 생성되면 출력을 사용하여 광고가 삽입된 콘텐츠에 액세스하는 방법을 이해해야 합니다. 이는 수동 설정에서 MediaTailor URLs을 사용하는 방법과 비슷하지만 CloudFormation 배포는 이러한 URLs 자동으로 제공합니다.

배포에 성공하면 CloudFormation 스택은 광고 삽입을 통해 콘텐츠에 액세스하는 데 사용할 몇 가지 중요한 출력을 제공합니다.

CloudFrontDomainName

CloudFront 배포의 도메인 이름(예: *d1234abcdef.CloudFront.net*)

HlsManifestUrl

광고 삽입이 포함된 HLS 매니페스트의 기본 URL(예: *https://d1234abcdef.CloudFront.net/v1/master/12345/my-playback-config/*)

DashManifestUrl

광고 삽입이 포함된 DASH 매니페스트의 기본 URL(예: *https://d1234abcdef.CloudFront.net/v1/dash/12345/my-playback-config/*)

MediaTailorPlaybackConfigName

생성된 MediaTailor 재생 구성의 이름(예: *my-stack-PlaybackConfig*)

재생 URLs 구성

광고가 포함된 콘텐츠의 전체 재생 URL을 생성하려면 CloudFormation 출력의 기본 URL을 특정 매니페스트 경로와 결합해야 합니다. 이는 브로드캐스트 전문가가 이해해야 할 중요한 단계입니다. 기존 콘텐츠를 광고 삽입 시스템과 연결하기 때문입니다.

1. 출력에서 적절한 매니페스트 URL로 시작합니다.

```
HlsManifestUrl: https://d1234abcdef.CloudFront.net/v1/master/12345/my-playback-config/
```

2. 특정 매니페스트 경로를 추가합니다.

```
Your manifest path: channel/index.m3u8
```

3. 전체 재생 URL은 다음과 같습니다.

```
https://d1234abcdef.CloudFront.net/v1/master/12345/my-playback-config/channel/index.m3u8
```

비디오 플레이어에서이 URL을 사용하여 동적으로 삽입된 광고로 콘텐츠를 재생합니다.

Tip

매니페스트 경로가 무엇인지 잘 모르는 경우 오리진 서버를 확인합니다. MediaPackage 오리진의 경우 엔드포인트의 HLS 또는 DASH 매니페스트 경로입니다. Amazon S3 오리진의 경우 버킷 내 매니페스트 파일의 경로입니다.

MediaTailor URL 구조에 대한 자세한 내용은 섹션을 참조하세요 [MediaTailor와 CDN 통합 설정](#).

비디오 플레이어 구성

재생 URL이 있으면 이를 사용하도록 비디오 플레이어를 구성해야 합니다. 브로드캐스트 전문가의 경우 이는 모든 HLS 또는 DASH 스트림에 대해 플레이어를 구성하는 것과 비슷하지만 이제 스트림에 개인화된 광고가 포함됩니다. 다음은 인기 HLS.js 플레이어를 사용하는 간단한 예입니다.

```
<!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>

```

다음과 같은 전문 브로드캐스트 플레이어 사용할 수도 있습니다.

- JW 플레이어
- Bitmovin 플레이어
- THEOplayer
- Video.js

MediaTailor와의 플레이어 통합에 대한 자세한 내용은 섹션을 참조하세요 [MediaTailor 광고 서버 통합 요구 사항](#).

CDN 및 MediaTailor 통합을 위한 CloudFormation 배포 테스트 및 검증

AWS Elemental MediaTailor 배포 검증은 라이브로 전환하기 전에 브로드캐스트 전문가에게 중요한 단계입니다. 이 섹션에서는 배포를 테스트하여 광고가 제대로 삽입되고 콘텐츠가 원활하게 전달되는지 확인하는 방법을 안내합니다.

CloudFormation 템플릿을 배포한 후 다음 단계에 따라 설정이 올바르게 작동하는지 확인합니다.

MediaTailor 및 CloudFront 통합을 테스트하려면

1. 콘솔에서 모든 리소스가 CloudFormation 성공적으로 생성되었는지 확인합니다.
2. MediaTailor [콘솔에서 MediaTailor](#) 재생 구성이 활성화 상태인지 확인합니다.
3. CloudFront [콘솔에서 CloudFront](#) 배포가 배포되고 활성화되어 있는지 확인합니다.
4. 샘플 매니페스트를 사용하여 재생을 테스트합니다.
 1. 에 설명된 대로 전체 재생 URL을 구성합니다 [재생 URLs 구성](#).
 2. HLS 또는 DASH를 지원하는 비디오 플레이어(예: VLC, JW 플레이어 또는 AWS 콘솔 플레이어)를 사용합니다.
 3. 콘텐츠 재생 및 광고가 예상 중단점에 삽입되었는지 확인합니다.
5. CloudWatch의 MediaTailor 로그에서 광고 삽입 오류가 있는지 확인합니다.

광고 삽입을 테스트할 때 성공 지표를 찾아보세요.

- 콘텐츠와 광고 간의 원활한 전환
- 광고는 예상 브레이크 포인트(프리롤, 미드롤, 포스트롤)에 표시됩니다.
- 광고 품질이 콘텐츠 품질과 일치
- 광고 전환 중 버퍼링 또는 재생 오류 없음

자세한 테스트 절차는 섹션을 참조하세요 [AWS Elemental MediaTailor 광고 삽입 동작 이해](#). 포괄적인 CDN 통합 테스트 및 검증은 섹션을 참조하세요 [CDN 및 MediaTailor 통합 테스트 및 검증](#).

CDN 및 MediaTailor 통합의 일반적인 CloudFormation 배포 문제 해결

AWS Elemental MediaTailor 배포 또는 재생 중에 자동화를 사용해도 배포 문제가 발생할 수 있습니다. 방송 전문가는 이러한 문제를 해결하는 방법을 이해하면 광고 삽입을 통해 안정적인 스트리밍 서비스를 유지하는 데 도움이 됩니다.

CloudFormation 배포 또는 결과 MediaTailor 및 CloudFront 통합에 문제가 발생하는 경우 다음과 같은 일반적인 문제 및 솔루션을 참조하세요.

CloudFormation 배포 문제

"리소스 생성 실패" 오류와 함께 스택 생성 실패

가능한 원인:

- 잘못된 콘텐츠 오리진 도메인 이름 형식
- 리소스를 생성할 수 있는 권한 부족

해결 방법: CloudFormation 이벤트 탭에서 특정 리소스 오류를 확인합니다. 콘텐츠 오리진 도메인 이름의 형식이 프로토콜 접두어나 경로 없이 올바른지 확인합니다. IAM 역할에 필요한 모든 리소스를 생성할 수 있는 충분한 권한이 있는지 확인합니다.

CloudFront 배포를 배포하는 데 시간이 오래 걸립니다.

원인: CloudFront 배포를 완전히 배포하는 데 일반적으로 15~30분이 걸립니다.

해결책: 이것은 정상적 동작입니다. 테스트하기 전에 배포가 "배포" 상태에 도달할 때까지 기다립니다.

재생 및 광고 삽입 문제

콘텐츠가 재생되지만 광고가 삽입되지 않음

가능한 원인:

- 광고 결정 서버가 응답하지 않거나 빈 VAST를 반환합니다.
- 콘텐츠에 광고 마커가 포함되지 않음

해결 방법: 광고 서버에 액세스할 수 있고 유효한 VAST 응답을 반환하는지 확인합니다. 콘텐츠에 적절한 광고 마커(라이브 콘텐츠의 경우 SCTE-35 마커 또는 VOD의 경우 광고 중단 태그)가 있는지 확인합니다.

403 콘텐츠에 액세스할 때 금지되는 오류

가능한 원인:

- 오리진 액세스 제어가 올바르게 구성되지 않음
- 오리진 버킷 또는 엔드포인트 권한 문제

해결 방법: Amazon S3 오리진의 경우 버킷 정책이 CloudFront 배포에서 액세스를 허용하는지 확인합니다. MediaPackage 오리진의 경우 오리진 액세스 제어가 올바르게 구성되어 있고 엔드포인트에 액세스할 수 있는지 확인합니다.

재생 오류 또는 버퍼링

가능한 원인:

- 캐시 동작 경로 패턴이 콘텐츠 경로와 일치하지 않음
- 잘못된 오리진 도메인 구성

해결 방법: 캐시 동작에 요청을 적절한 오리진으로 라우팅하는 올바른 경로 패턴이 있는지 확인합니다. CloudFront 로그를 확인하여 요청을 처리하는 오리진을 확인하고 예상되는 오리진인지 확인합니다.

브로드캐스트 전문가의 경우 다음과 같은 추가 문제 해결 팁이 도움이 될 수 있습니다.

- Amazon CloudWatch Logs Insights를 사용하여 MediaTailor 로그에서 특정 오류 패턴 검색
- 프로덕션 광고 서버를 사용하기 전에 간단한 VAST 광고 서버(예: 템플릿에 제공된 기본 서버)로 먼저 테스트합니다.
- 콘솔의 MediaTailor 매니페스트 검사기 도구를 사용하여 콘텐츠의 광고 마커 확인
- 브라우저의 개발자 도구에서 네트워크 트래픽을 확인하여 광고 서버에 대한 요청이 올바르게 수행되고 있는지 확인합니다.

추가 문제 해결을 위해 MediaTailor 및 CloudFront 모두에 대한 CloudWatch CloudWatch 로그를 확인하여 특정 오류를 식별합니다.

CDN 및 MediaTailor 통합을 위한 CloudFormation 템플릿 사용자 지정

AWS Elemental MediaTailor 템플릿 사용자 지정을 사용하면 브로드캐스트 전문가가 특정 워크플로 요구 사항에 맞게 AWS CloudFormation 템플릿을 조정할 수 있습니다. 기본 템플릿은 많은 시나리오에서 작동하지만 이러한 사용자 지정은 더 복잡한 요구 사항을 해결하는 데 도움이 될 수 있습니다.

아래 예제에서는 템플릿에 추가할 수 있는 YAML 코드 조각을 보여줍니다. YAML 또는 CloudFormation 구문에 익숙하지 않은 경우 개발자 또는 AWS 솔루션 아키텍트와 협력하여 이러한 변경을 수행하는 것이 좋습니다.

특정 워크플로 요구 사항에 맞게 CloudFormation 템플릿을 사용자 지정할 수 있습니다.

오리진 추가 또는 수정

여러 콘텐츠 소스(예: 기본 및 백업 소스 또는 다른 콘텐츠 라이브러리)를 사용하는 브로드캐스트 워크플로의 경우 CloudFront 배포에 오리진을 추가할 수 있습니다.

```
Origins:
  # Add a new origin for additional content
  - Id: SecondaryContentOrigin
    DomainName: secondary-content.example.com
    CustomOriginConfig:
      OriginProtocolPolicy: 'https-only'
```

```
OriginSSLProtocols:
  - TLSv1.2
```

그런 다음 해당 캐시 동작을 추가하여 특정 패턴을 이 오리진으로 라우팅합니다.

```
CacheBehaviors:
  - PathPattern: '/secondary-content/*'
    TargetOriginId: SecondaryContentOrigin
    ViewerProtocolPolicy: 'https-only'
    CachePolicyId: 658327ea-f89d-4fab-a63d-7e88639e58f6 # Managed-CachingOptimized
```

사용자 지정 캐시 정책 생성

특정 캐싱 요구 사항(예: 품질 선택 파라미터 또는 뷰어 인증)이 있는 브로드캐스트 워크플로의 경우 관리형 캐시 정책을 사용하는 대신 사용자 지정 캐시 정책을 생성할 수 있습니다. TTL 값 및 캐싱 전략에 대한 자세한 지침은 섹션을 참조하세요 [CDN 및 MediaTailor 통합을 위한 캐싱 최적화](#).

```
# Define a custom cache policy
CustomCachePolicy:
  Type: AWS::CloudFront::CachePolicy
  Properties:
    CachePolicyConfig:
      Name: !Sub '${AWS::StackName}-CustomCachePolicy'
      DefaultTTL: 86400 # 24 hours
      MaxTTL: 31536000 # 1 year
      MinTTL: 1 # 1 second
      ParametersInCacheKeyAndForwardedToOrigin:
        CookiesConfig:
          CookieBehavior: none
        HeadersConfig:
          HeaderBehavior: none
        QueryStringsConfig:
          QueryStringBehavior: whitelist
          QueryStrings:
            - quality
            - format

# Reference the custom policy in a cache behavior
CacheBehaviors:
  - PathPattern: '/custom-path/*'
    TargetOriginId: ContentOrigin
    ViewerProtocolPolicy: 'https-only'
```



```
CachePolicyId: !Ref CustomCachePolicy
```

MediaTailor 구성 개선

고급 광고 삽입 기능이 필요한 브로드캐스트 워크플로의 경우 광고 미리 가져오기(지연 시간 단축), 개인화 임계값 및 범퍼 광고와 같은 옵션으로 MediaTailor 구성을 개선할 수 있습니다.

```
MediaTailorPlaybackConfig:
  Type: AWS::MediaTailor::PlaybackConfiguration
  Properties:
    # Add ad prefetching for improved performance
    AvailSuppression:
      Mode: BEHIND_LIVE_EDGE
      Value: 00:00:00
    # Add personalization parameters
    PersonalizationThresholdSeconds: 2
    # Add bumper ads
    Bumper:
      StartUrl: https://example.com/bumper-start.mp4
      EndUrl: https://example.com/bumper-end.mp4
    # Other existing properties...
```

MediaTailor 구성 옵션에 대한 자세한 내용은 섹션을 참조하세요 [AWS Elemental MediaTailor 를 사용하여 광고 삽입](#).

보안 기능 추가

특정 보안 요구 사항(지리적 제한 또는 DDoS 공격에 대한 보호 등)이 있는 브로드캐스트 워크플로의 경우 AWS WAF 통합 및 지리적 제한을 추가할 수 있습니다.

```
# Create a AWS WAF Web ACL
WebACL:
  Type: AWS::WAFv2::WebACL
  Properties:
    Name: !Sub '${AWS::StackName}-WebACL'
    Scope: CloudFront
    DefaultAction:
      Allow: {}
    VisibilityConfig:
      SampledRequestsEnabled: true
      CloudWatchMetricsEnabled: true
      MetricName: !Sub '${AWS::StackName}-WebACL'
```

```

Rules:
  - Name: RateLimitRule
    Priority: 0
    Action:
      Block: {}
    VisibilityConfig:
      SampledRequestsEnabled: true
      CloudWatchMetricsEnabled: true
      MetricName: RateLimitRule
    Statement:
      RateBasedStatement:
        Limit: 1000
        AggregateKeyType: IP

# Reference the AWS WAF Web ACL in the CloudFront distribution
CloudFrontDistribution:
  Type: AWS::CloudFront::Distribution
  Properties:
    DistributionConfig:
      WebACLId: !GetAtt WebACL.Arn
      # Add geo-restriction
      Restrictions:
        GeoRestriction:
          RestrictionType: whitelist
          Locations:
            - US
            - CA
            - GB
      # Other existing properties...

```

CloudFormation 템플릿에 대한 자세한 내용은 [AWS CloudFormation 사용 설명서를](#) 참조하세요.

브로드캐스트별 CloudFormation 템플릿 및 예제는 [AWS Media Services 도구 GitHub 리포지토리](#)를 참조하세요.

AWS CloudFormation AWS Elemental MediaTailor 및 Amazon CloudFront 통합에 대한 템플릿 참조

AWS Elemental MediaTailor Amazon CloudFront와의 통합은 다음 전체 AWS CloudFormation 템플릿을 사용하여 자동화할 수 있습니다.

```
AWSTemplateFormatVersion: '2010-09-09'
```

Description: | CloudFormation template that sets up AWS Elemental MediaTailor integration with CloudFront Distribution for server-side ad insertion. This template supports various content origins including MediaPackage, Amazon S3, and custom origins, making it versatile for different streaming architectures.

Parameters:

AdServerUrl:

Type: String

Default: 'https://d1kbnmkziz9rksx.CloudFront.net/VASTEndpoint.xml'

Description: URL of the VAST ad server for dynamic ad insertion. Static VAST endpoint provided for testing.

ContentOriginDomainName:

Type: String

Description: |

Domain name of your content origin without protocol (e.g., *mediapackage-domain.mediapackagev2.us-west-2.amazonaws.com*, *mybucket.s3.amazonaws.com*, or *custom-origin.example.com*). Do not include http:// or https:// prefixes or any paths.

AllowedPattern: `"^(([a-zA-Z0-9]|[a-zA-Z0-9][a-zA-Z0-9\\-]*[a-zA-Z0-9])\\.)*([A-Za-z0-9]|[A-Za-z0-9][A-Za-z0-9\\-]*[A-Za-z0-9])$"`

ConstraintDescription: Must be a valid domain name (e.g., *example.com*) without protocol or path components. IP addresses are not allowed.

ContentOriginType:

Type: String

AllowedValues:

- mediapackagev2
- s3
- custom

Default: mediapackagev2

Description: |

The type of content origin:

- mediapackagev2: AWS Elemental MediaPackage V2
- s3: Amazon S3 bucket
- custom: Any other custom origin

Resources:

```
#-----
# Origin Access Control (for securing MediaPackage V2 and Amazon S3 origins)
#-----
```

CloudFrontOriginAccessControl:

Type: AWS::CloudFront::OriginAccessControl

Condition: IsNotCustomOrigin

Properties:

OriginAccessControlConfig:

Name: !Sub '\${AWS::StackName}-OAC'

OriginAccessControlOriginType: !Ref ContentOriginType

SigningBehavior: always

SigningProtocol: sigv4

Description: Origin Access Control for content origin

#-----

MediaTailor Playback Configuration

#-----

MediaTailorPlaybackConfig:

Type: AWS::MediaTailor::PlaybackConfiguration

Properties:

Name: !Sub '\${AWS::StackName}-PlaybackConfig'

The video content source should point to your CloudFront distribution

VideoContentSourceUrl: !Sub 'https://\${CloudFrontDistribution.DomainName}/'

The Ad Decision Server URL is where MediaTailor will request ads

AdDecisionServerUrl: !Ref AdServerUrl

Configuration for pre-roll ads during live streams

LivePreRollConfiguration:

AdDecisionServerUrl: !Ref AdServerUrl

MaxDurationSeconds: 30

CDN configuration for integrating with CloudFront

CdnConfiguration:

AdSegmentUrlPrefix: '/'

ContentSegmentUrlPrefix: '/'

Set a reasonable manifest segment timeout

ManifestProcessingRules:

AdMarkerPassthrough:

Enabled: false

#-----

CloudFront Distribution

#-----

CloudFrontDistribution:

Type: AWS::CloudFront::Distribution

Properties:

DistributionConfig:

Enabled: true

HttpVersion: http2and3

IPV6Enabled: true

```

    Comment: !Sub 'Distribution for MediaTailor ad insertion with
    ${ContentOriginType} origin'

    # Default cache behavior points to the content origin
    DefaultCacheBehavior:
      TargetOriginId: ContentOrigin
      ViewerProtocolPolicy: 'https-only'
      # Using managed policies for optimal performance and simplicity
      CachePolicyId: 658327ea-f89d-4fab-a63d-7e88639e58f6 # Managed-
CachingOptimized
      OriginRequestPolicyId: 88a5eaf4-2fd4-4709-b370-b4c650ea3fcf # Managed-
HostHeaderOnly
      ResponseHeadersPolicyId: eaab4381-ed33-4a86-88ca-d9558dc6cd63 # Managed-
CORS-with-preflight-and-SecurityHeadersPolicy
      Compress: true

    # Define all the origins needed for the workflow
    Origins:
      # Main content origin (MediaPackage, Amazon S3, or Custom)
      - Id: ContentOrigin
        DomainName: !Ref ContentOriginDomainName
        # Apply Origin Access Control for secure origins
        OriginAccessControlId: !If [IsNotCustomOrigin, !GetAtt
CloudFrontOriginAccessControl.Id, !Ref "AWS::NoValue"]
        # For custom origins, we need a CustomOriginConfig
        CustomOriginConfig:
          OriginProtocolPolicy: 'https-only'
          OriginSSLProtocols:
            - TLSv1.2
          OriginKeepaliveTimeout: 5
          OriginReadTimeout: 30
          HTTPPort: 80
          HTTPSPort: 443

      # MediaTailor Manifests Origin - handles manifest manipulation for ad
insertion
      - Id: MediaTailorManifests
        DomainName: !Sub 'manifests.mediatailor.${AWS::Region}.amazonaws.com'
        CustomOriginConfig:
          OriginProtocolPolicy: 'https-only'
          OriginSSLProtocols:
            - TLSv1.2
          OriginKeepaliveTimeout: 5
          OriginReadTimeout: 30

```

```

    # Origin Shield improves caching efficiency
    OriginShield:
      Enabled: true
      OriginShieldRegion: !Ref AWS::Region

# MediaTailor Segments Origin - handles personalized ads
- Id: MediaTailorSegments
  DomainName: !Sub 'segments.mediatailor.${AWS::Region}.amazonaws.com'
  CustomOriginConfig:
    OriginProtocolPolicy: 'https-only'
    OriginSSLProtocols:
      - TLSv1.2
    OriginKeepaliveTimeout: 5
    OriginReadTimeout: 30

# Cache behaviors to route specific request patterns to the right origin
CacheBehaviors:
  # Handle MediaTailor segment requests for ad content which are cache-able
  - PathPattern: '/tm/*'
    TargetOriginId: MediaTailorSegments
    ViewerProtocolPolicy: 'https-only'
    CachePolicyId: 658327ea-f89d-4fab-a63d-7e88639e58f6 # Managed-
CachingOptimized
    OriginRequestPolicyId: 88a5eaf4-2fd4-4709-b370-b4c650ea3fcf # Managed-
HostHeaderOnly
    ResponseHeadersPolicyId: eaab4381-ed33-4a86-88ca-d9558dc6cd63 # Managed-
CORS-with-preflight-and-SecurityHeadersPolicy
    Compress: true

  # Handle MediaTailor interstitial (SGAI) media requests which are cache-able
  - PathPattern: '/v1/i-media/*'
    TargetOriginId: MediaTailorManifests
    ViewerProtocolPolicy: 'https-only'
    CachePolicyId: 658327ea-f89d-4fab-a63d-7e88639e58f6 # Managed-
CachingOptimized
    OriginRequestPolicyId: 88a5eaf4-2fd4-4709-b370-b4c650ea3fcf # Managed-
HostHeaderOnly
    ResponseHeadersPolicyId: eaab4381-ed33-4a86-88ca-d9558dc6cd63 # Managed-
CORS-with-preflight-and-SecurityHeadersPolicy
    Compress: true

  # Handle MediaTailor Personalized manifests which are not cache-able
  - PathPattern: '/v1/*'
    TargetOriginId: MediaTailorManifests

```

```

ViewerProtocolPolicy: 'https-only'
CachePolicyId: 4135ea2d-6df8-44a3-9df3-4b5a84be39ad # Managed-
CachingDisabled
OriginRequestPolicyId: 59781a5b-3903-41f3-afcb-af62929ccde1 # Managed-
AllViewer
ResponseHeadersPolicyId: eaab4381-ed33-4a86-88ca-d9558dc6cd63 # Managed-
CORS-with-preflight-and-SecurityHeadersPolicy
Compress: true

# Handle MediaTailor segment *redirect* requests which are not cache-able
(used for server side reporting)
- PathPattern: '/segment/*'
  TargetOriginId: MediaTailorManifests
  ViewerProtocolPolicy: 'https-only'
  CachePolicyId: 4135ea2d-6df8-44a3-9df3-4b5a84be39ad # Managed-
CachingDisabled
OriginRequestPolicyId: 59781a5b-3903-41f3-afcb-af62929ccde1 # Managed-
AllViewer
ResponseHeadersPolicyId: eaab4381-ed33-4a86-88ca-d9558dc6cd63 # Managed-
CORS-with-preflight-and-SecurityHeadersPolicy
Compress: true

Conditions:
  IsNotCustomOrigin: !Not [!Equals [!Ref ContentOriginType, 'custom']]

Outputs:
  CloudFrontDomainName:
    Description: Domain name of the CloudFront distribution
    Value: !GetAtt CloudFrontDistribution.DomainName

  HlsManifestUrl:
    Description: URL for HLS manifest with ads inserted (append your manifest path)
    Value: !Sub 'https://
${CloudFrontDistribution.DomainName}${MediaTailorPlaybackConfig.HlsConfiguration.ManifestEndpoint

  DashManifestUrl:
    Description: URL for DASH manifest with ads inserted (append your manifest path)
    Value: !Sub 'https://
${CloudFrontDistribution.DomainName}${MediaTailorPlaybackConfig.DashConfiguration.ManifestEndpoint

  MediaTailorPlaybackConfigName:
    Description: Name of the MediaTailor playback configuration
    Value: !Ref MediaTailorPlaybackConfig

```

MediaTailor에 대한 프로덕션 지원 CloudFront 구성

이 CloudFront 배포 구성은 서버 측 광고 삽입을 통해 MediaTailor 콘텐츠를 대규모로 제공하는 데 필요한 모든 것을 제공합니다. 이 구성을 복사하고 특정 오리진 및 요구 사항으로 사용자 지정합니다.

이 구성이 수행하는 작업

이 구성은 최적의 캐싱 및 성능으로 모든 MediaTailor 요청 유형을 처리하는 프로덕션 지원 CloudFront 배포를 생성합니다. 여기에는 요청을 올바르게 라우팅하고 콘텐츠를 적절하게 캐싱하는 네 가지 캐시 동작이 있는 세 가지 오리진(콘텐츠, MediaTailor 세그먼트 및 MediaTailor 매니페스트)이 포함됩니다.

이 구성을 사용해야 하는 경우

이 설정은 서버 측 광고 삽입이 필요한 라이브 스트리밍, video-on-demand 및 하이브리드 워크플로에 적합합니다.

3원 아키텍처

MediaTailor는 3원 아키텍처 패턴을 사용하여 콘텐츠 전송 및 광고 삽입 성능을 최적화합니다. 각 오리진은 광고 삽입 워크플로에서 특정 목적을 수행합니다.

콘텐츠 오리진

MediaTailor에 공급하는 실제 콘텐츠 오리진입니다. 예를 들어 AWS Elemental MediaPackage V2 또는 다른 콘텐츠 전송 서비스일 수 있습니다. 이 오리진은 광고 삽입 전에 원래 콘텐츠를 제공합니다. 일반적인 예는 다음과 같습니다.

- MediaPackage V2 패키징 구성
- 타사 콘텐츠 전송 네트워크
- 온프레미스 스트리밍 서버
- 정적 콘텐츠가 있는 Amazon S3 버킷

MediaTailor 세그먼트 오리진

이 오리진은 호스트 이름을 사용하고 MediaTailor가 호스트 이름을 인코딩한 후 실제 광고 세그먼트를 `segments.mediatailor.region.amazonaws.com` 제공합니다. 광고가 포함된 비디오 세그먼트입니다. 이 오리진은 다음을 처리합니다.

- 콘텐츠와 동일한 형식의 트랜스코딩된 광고 세그먼트
- `/segment/*` 경로 패턴에서 리디렉션된 요청

- 서버 측 광고 삽입을 위해 처리된 광고 세그먼트

MediaTailor 매니페스트 오리진

이 오리진은 호스트 이름을 사용 `manifests.mediatailor.region.amazonaws.com` 하며 지정된 리전의 재생 구성을 위한 AWS 리전 호스트 이름으로 사용할 수 있습니다. MediaTailor는 요청의 경로를 기반으로 올바른 재생 구성을 선택합니다. 이 오리진은 다음을 제공합니다.

- 최종 사용자별 광고 삽입이 포함된 개인화된 HLS 및 DASH 매니페스트
- 캐시 가능한 콘텐츠에 대한 서버 가이드 광고 삽입(SGAI) 매니페스트
- 서버 측 보고를 위한 광고 추적 및 비컨 처리

오리진 호스트 이름을 사용하면 지정된 리전에 있고 요청 경로에 재생 구성 이름을 포함하는 경우 동일한 CloudFront 배포에서 작동하는 여러 재생 구성을 가질 `manifests.mediatailor.region.amazonaws.com` 수 있습니다. 예제:

- `https://your-distribution.cloudfront.net/v1/master/playback-config-1/manifest.m3u8`
- `https://your-distribution.cloudfront.net/v1/master/playback-config-2/manifest.m3u8`

콘텐츠 유형을 기반으로 오리진 요청 정책을 선택하여 적절한 기능을 보장하면서 캐시 중복을 방지합니다. 주요 차이점은 캐시 가능한 콘텐츠와 캐시 불가능한 콘텐츠입니다.

- 매니페스트(캐싱 불가): 동적 콘텐츠에 필요한 모든 헤더를 전달하는 AllViewer 데 사용합니다. 매니페스트는 캐시되지 않으므로 캐시 중복 위험이 없습니다.
- 세그먼트(캐싱 가능): 최적의 성능을 None 위해를 사용합니다.
- S3 오리진: Amazon S3 버킷 CORS-S3origin에 사용
- MediaPackage 오리진: MediaPackage V2 엔드포인트 CORS-S3origin에 사용

Origin name	Origin domain	Origin path	Origin type	Origin Shield region	Origin access
ContentOrigin	zrljic.egress.kv9ydh.mediapackagev2.us-west-2.amazonaws.com		MediaPackage V2	us-west-2	E2SRJY3NEMC428
MediaTailorSegments	segments.mediatailor.us-west-2.amazonaws.com		MediaTailor	-	-
MediaTailorManifests	manifests.mediatailor.us-west-2.amazonaws.com		MediaTailor	us-west-2	-

캐시 동작 우선 순위 및 구성

MediaTailor는 다양한 유형의 요청을 올바르게 처리하기 위해 특정 캐시 동작 구성이 필요합니다. CDNs은 캐시 동작을 순서대로 처리하고(가장 작음에서 가장 큼) 일치하는 첫 번째 경로 패턴에 동작을 사용하기 때문에 캐시 동작의 우선 순위가 중요합니다. 이 우선 순위를 이해하는 것은 문제 해결에 필수적입니다.

- 우선 순위 0: 가장 구체적인 패턴(예: /tm/*)이 먼저 평가됩니다.
- 우선 순위가 높은 숫자: 덜 구체적인 패턴은 순서대로 평가됩니다.
- 기본 동작: 다른 패턴과 일치하지 않는 모든 요청을 캐싱합니다.

요청이 예상대로 동작하지 않는 경우 경로 패턴이 의도하지 않은 방식으로 겹치지 않는지 확인합니다.

General	Security	Origins	Behaviors	Error pages	Invalidations	Tags	Logging
Behaviors							
Filter behaviors by property or value Save Move up Move down Edit Delete Create behavior							
Preced...	Path patt...	Origin or origin group	Viewer pr...	Cache policy name	Origin request policy name	Response headers policy name	
0	/tm/*	MediaTailorSegments	HTTPS only	Managed-CachingOptimized	Managed-CORS-S3Origin	Managed-CORS-with-preflight-and-SecurityHeadersPolicy	
1	/i-media/*	MediaTailorManifests	HTTPS only	Managed-CachingOptimized	Managed-CORS-S3Origin	Managed-CORS-with-preflight-and-SecurityHeadersPolicy	
2	/v1/*	MediaTailorManifests	HTTPS only	Managed-CachingDisabled	Managed-AllViewerAndCloudFrontHeaders-2022-06	Managed-CORS-with-preflight-and-SecurityHeadersPolicy	
3	/segment/*	MediaTailorManifests	HTTPS only	Managed-CachingDisabled	Managed-AllViewerAndCloudFrontHeaders-2022-06	Managed-CORS-with-preflight-and-SecurityHeadersPolicy	
4	Default (*)	ContentOrigin	HTTPS only	Managed-CachingOptimized	CustomOriginRequestPolicyWithWithLLHLSDirectives	Managed-CORS-with-preflight-and-SecurityHeadersPolicy	

우선순위 0: 광고 세그먼트 경로 동작

이 동작은 세그먼트 경로 동작에서 리디렉션된 요청을 처리하여 실제 광고 세그먼트를 처리합니다. CloudFront는 /tm/* 경로 패턴이 있는 모든 요청에 다음 동작을 적용합니다. 광고 세그먼트 전송은 중단 없는 재생에 중요하므로 이 동작이 우선순위가 가장 높습니다.

- 경로 패턴: /tm/*

이 패턴과 일치하는 URLs에:

- `https://your-distribution.cloudfront.net/tm/ad-segment-001.ts`
- `https://your-distribution.cloudfront.net/tm/transcoded-ad.m4s`
- 오리진: `segments.mediataylor.region.amazonaws.com` 도메인으로 생성한 오리진입니다.

오리진에 대한 이전 섹션의 예제에서 MediaTailorSegments입니다.

- 캐시 정책: Managed-CachingOptimized

CloudFront 관리형 캐시 정책에 포함된 항목에 대한 자세한 내용은 CloudFront 사용 설명서의 [CachingOptimized](#)를 참조하세요. 타사 CDN의 관리형 정책에서 이러한 동일한 설정을 사용할 수도 있습니다.

- 오리진 요청 정책: None
- 응답 헤더 정책: Managed-CORS-with-preflight-and-SecurityHeadersPolicy

응답 헤더 정책에 포함된 항목에 대한 자세한 내용은 CloudFront 사용 설명서의 [CORS-with-preflight-and-SecurityHeadersPolicy](#)를 참조하세요.

이러한 설정을 다른 CDN에 적용

CloudFront 이외의 CDN을 사용하는 경우 다음을 수행하는 동등한 설정을 찾습니다.

경로 패턴 일치

MediaTailor 광고 세그먼트를 처리하도록 /tm/* 경로 패턴에 대한 특정 동작 구성

캐시 키 구성

캐시 키에 Origin 헤더를 포함하여 응답이 서로 다른 오리진에 대해 별도로 캐시되도록 합니다.

헤더 전달

Origin 헤더 및 기타 CORS 관련 헤더를 오리진에 전달

응답 헤더 관리

응답에 Access-Control-Allow-Origin 헤더가 표시되도록 CDN 구성

특정 용어 및 구성 옵션은 CDN 공급자에 따라 다르지만 기본 원칙은 동일하게 유지됩니다.

우선순위 1: 서버 가이드 광고 삽입 동작

이 동작은 고객이 캐시 가능한 매니페스트를 제공하는 안내 모드를 구성할 때 [MediaTailor 서버 가이드 광고 삽입 개요 및 구현](#) (SGAI) 요청을 처리합니다. CloudFront는 /v1/i-media/* 경로 패턴이 있는 모든 요청에 다음 동작을 적용합니다. 매니페스트는 최종 사용자별로 고유하지 않으므로 SGAI를 사용하면 캐싱 성능이 향상됩니다.

- 경로 패턴: /v1/i-media/* (SGAI용 iMedia 경로)

이 패턴과 일치하는 URLs에:

- `https://your-distribution.cloudfront.net/v1/i-media/your-config/manifest.m3u8`
- `https://your-distribution.cloudfront.net/v1/i-media/your-config/playlist.mpd`
- 오리진: `manifests.mediatailor.region.amazonaws.com` 도메인으로 생성한 오리진입니다.

오리진에 대한 이전 섹션의 예제에서 MediaTailorManifests입니다.

- 캐시 정책: `Managed-CachingOptimized`

CloudFront 관리형 캐시 정책에 포함된 항목에 대한 자세한 내용은 CloudFront 사용 설명서의 [CachingOptimized](#)를 참조하세요. 타사 CDN의 관리형 정책에서 이러한 동일한 설정을 사용할 수도 있습니다.

- 오리진 요청 정책: `None`
- 응답 헤더 정책: `Managed-CORS-with-preflight-and-SecurityHeadersPolicy`

응답 헤더 정책에 포함된 항목에 대한 자세한 내용은 CloudFront 사용 설명서의 [CORS-with-preflight-and-SecurityHeadersPolicy](#)를 참조하세요.

우선순위 2: 개인화된 매니페스트 동작

이 동작은 개인화된 매니페스트 요청을 처리합니다. CloudFront는 `/v1/*` 경로 패턴이 있는 모든 요청에 다음 동작을 적용합니다. CloudFront는 다음 동작을 적용하고 개인화된 매니페스트는 최종 사용자별 광고 콘텐츠 URLs 포함하므로 캐싱하지 않습니다. 이러한 동작은 `/v1/*` 경로 패턴이 있는 모든 요청에 적용됩니다. 이는 각 최종 사용자가 맞춤형 광고 삽입이 포함된 고유한 매니페스트를 수신하는 핵심 MediaTailor 기능입니다.

- 경로 패턴: `/v1/*` (표준 V1 MediaTailor 요청)

이 패턴과 일치하는 URLs에:

- `https://your-distribution.cloudfront.net/v1/master/your-config/manifest.m3u8`
- `https://your-distribution.cloudfront.net/v1/dash/your-config/manifest.mpd`
- 오리진: `manifests.mediatailor.region.amazonaws.com` 도메인으로 생성한 오리진입니다.

오리진에 대한 이전 섹션의 예제에서 MediaTailorManifests입니다.

- 캐시 정책: Managed-CachingDisabled

캐시 정책에 포함된 항목에 대한 자세한 내용은 CloudFront 사용 설명서의 [CachingDisabled](#)를 참조하세요.

- 오리진 요청 정책: AllViewer

개인화된 매니페스트의 경우 AllViewer 정책을 사용하여 동적 콘텐츠에 필요한 모든 헤더를 전달합니다.

- 응답 헤더 정책: Managed-CORS-with-preflight-and-SecurityHeadersPolicy

응답 헤더 정책에 포함된 항목에 대한 자세한 내용은 CloudFront 사용 설명서의 [CORS-with-preflight-and-SecurityHeadersPolicy](#)를 참조하세요.

수락-인코딩 헤더

CDN은 최종 사용자로부터 Accept-Encoding 헤더를 보존하는 것이 좋습니다. 이 헤더는 MediaTailor에 개인화된 매니페스트 압축에 대한 지침을 제공합니다.

CloudFront에서 AllViewerAndCloudFrontHeaders 오리진 요청 정책에는 최종 사용자의 Accept-Encoding 헤더 전달이 포함됩니다. 다른 CDN을 사용하는 경우 이 헤더를 보존해야 합니다.

다음은 MediaTailor가 Accept-Encoding 헤더를 처리하는 방법입니다.

- 레거시 디바이스: gzip을 지원하지 않는 이전 스마트 TVs는 Accept-Encoding 헤더를 보내지 않으므로 MediaTailor는 압축되지 않은 매니페스트를 반환합니다.
- 최신 디바이스: iPhones, Chrome 브라우저 및 기타 최신 클라이언트가 Accept-Encoding 헤더를 전송하여 MediaTailor가 전송 전에 매니페스트를 압축하도록 허용

우선순위 3: 서버 측 비컨 경로 동작

이 동작은 MediaTailor에 대한 요청을 처리하여에 대한 리디렉션을 생성합니다 [서버 측 추적](#). 이러한 요청은 비컨을 추적하는 데 필수적이므로 모든 요청은 MediaTailor에서 처리해야 합니다. CloudFront는 /segment/* 경로 패턴이 있는 모든 요청에 다음 동작을 적용합니다.

- 경로 패턴: /segment/*

이 패턴과 일치하는 URLs에:

- <https://your-distribution.cloudfront.net/segment/tracking-beacon-123>

- `https://your-distribution.cloudfront.net/segment/ad-request-456.ts`
- 오리진: `manifests.mediatailor.region.amazonaws.com` 도메인으로 생성한 오리진입니다.

오리진에 대한 이전 섹션의 예제에서 MediaTailorManifests입니다.

- 캐시 정책: Managed-CachingDisabled

캐시 정책에 포함된 항목에 대한 자세한 내용은 CloudFront 사용 설명서의 [CachingDisabled](#)를 참조하세요.

- 오리진 요청 정책: AllViewer

서버 측 비컨 요청의 경우 AllViewer 정책을 사용하여 추적에 필요한 모든 헤더를 전달합니다. 이러한 요청은 캐시되지 않으므로 캐시 중독 위험이 없습니다.

- 응답 헤더 정책: Managed-CORS-with-preflight-and-SecurityHeadersPolicy

응답 헤더 정책에 포함된 항목에 대한 자세한 내용은 CloudFront 사용 설명서의 [CORS-with-preflight-and-SecurityHeadersPolicy](#)를 참조하세요.

MediaTailor는 이러한 요청을 처리할 때 실제 세그먼트 위치를 가리키는 경로와 함께 302 리디렉션 응답을 반환합니다. 예를 들어에 대한 요청은 세그먼트 오리진/`tm/encoded-ad-segment.ts`의 로 리디렉션/`segment/ad123.ts`될 수 있습니다.

우선순위 4: 콘텐츠 오리진 경로 동작

요청 경로가 다른 패턴과 일치하지 않는 경우 CloudFront는 기본 동작을 적용합니다. 이 동작은 MediaTailor에서 처리하지 않고 요청을 콘텐츠 오리진으로 직접 전송합니다. 이렇게 하면 필요한 경우 콘텐츠 오리진(예: MediaPackage V2)에 직접 액세스할 수 있습니다. CloudFront는 이전 경로 패턴을 포함하지 않는 모든 요청에 다음 동작을 적용합니다.

- 경로 패턴: (*)
- 오리진: 콘텐츠 오리진의 도메인으로 생성한 오리진입니다.

오리진에 대한 이전 섹션의 예제에서 ContentOrigin입니다.

- 캐시 정책: Managed-CachingOptimized

CloudFront 관리형 캐시 정책에 포함된 항목에 대한 자세한 내용은 CloudFront 사용 설명서의 [CachingOptimized](#)를 참조하세요. 타사 CDN의 관리형 정책에서 이러한 동일한 설정을 사용할 수도 있습니다.

Note

지연 시간이 짧은 HLS 구현의 경우 표준 정책 대신 지연 시간이 짧은 HLS(LLH) 명령과 함께 사용자 지정 캐싱 `CachingOptimized` 정책을 사용하는 것이 좋습니다.

- 오리진 요청 정책: `None`
- 응답 헤더 정책: `Managed-CORS-with-preflight-and-SecurityHeadersPolicy`

기본 콘텐츠 오리진 동작은 일반적으로 광고 세그먼트 동작과 동일한 CORS 캐시 중독 위험에 직면하지 않지만 응답 `Managed-CORS-with-preflight-and-SecurityHeadersPolicy` 헤더 정책을 사용하고 캐시 키에 `Origin` 헤더를 포함하는 것이 좋습니다. 이렇게 하면 모든 콘텐츠 유형에서 일관된 CORS 처리를 보장하고 웹 기반 플레이어의 잠재적 재생 문제를 방지할 수 있습니다.

콘텐츠 세그먼트의 경우 `Managed-CachingOptimized` 캐시 정책은 우수한 성능을 제공하는 반면 `Managed-CORS-with-preflight-and-SecurityHeadersPolicy` 응답 헤더 정책은 적절한 CORS 처리를 보장합니다. 이 조합을 통해 CORS 헤더가 필요한 웹 기반 플레이어와의 호환성을 유지하면서 효율적인 캐싱이 가능합니다.

광고 세그먼트와 콘텐츠 세그먼트 모두에 일관된 CORS 처리를 적용하면 보다 안정적인 재생 환경이 생성되고 문제 해결이 간소화됩니다. 적절한 CORS 구성이 없으면 콘텐츠와 광고 간에 전환할 때 플레이어가 일관되지 않은 동작을 경험할 수 있습니다.

응답 헤더 정책에 포함된 항목에 대한 자세한 내용은 CloudFront 사용 설명서의 [CORS-with-preflight-and-SecurityHeadersPolicy](#)를 참조하세요.

CDN 및 MediaTailor 통합에 대한 지원 및 문제 해결 도움받기

AWS Elemental MediaTailor CDN 통합 문제는 진단 및 해결이 복잡할 수 있습니다. 셀프 서비스 문제 해결을 통해 해결할 수 없는 CDN 및 MediaTailor 통합 문제에 대한 추가 도움이 필요한 경우 이 가이드를 사용하세요.

에스컬레이션하기 전에 다음 셀프 서비스 옵션을 시도해 보세요.

- 의 문제 해결 단계를 따릅니다. [CDN 통합 문제 해결](#)
- 를 사용하여 로그 및 오류 코드 분석 [CDN 통합 로그 분석 참조](#)
- 를 사용하여 모니터링 설정 검토 [모니터링 도구 설정](#)
- [MediaTailor 문제 해결 가이드](#)에서 서비스별 문제를 확인하세요.

- [AWS re:Post](#)에서 유사한 문제 및 커뮤니티 솔루션 검색
- [AWS 지식 센터](#)에서 일반적인 통합 패턴 검토

주제

- [MediaTailor CDN 문제를 AWS Support로 에스컬레이션해야 하는 경우](#)
- [지원 센터에 문의하기 전에 MediaTailor CDN 정보를 수집합니다.](#)
- [MediaTailor CDN 통합 지원 리소스 및 채널](#)
- [MediaTailor CDN 통합 지원 사례 모범 사례](#)

MediaTailor CDN 문제를 AWS Support로 에스컬레이션해야 하는 경우

AWS Elemental MediaTailor 셀프 서비스 문제 해결로 문제가 해결되지 않으면 CDN 통합 문제를 AWS Support에 에스컬레이션해야 합니다. 다음과 같은 경우 Support로 AWS 에스컬레이션하는 것이 좋습니다.

참고: 기술 지원에는 유료 AWS 지원 플랜이 필요합니다. 지원 플랜 세부 정보 및 응답 시간은 [AWS 지원 플랜](#)을 참조하세요.

- 문제는 프로덕션 트래픽 또는 수익에 영향을 미칩니다.
- 해결하지 않고 모든 문제 해결 단계를 따랐습니다.
- 오류 패턴은 서비스 수준 문제를 시사합니다.
- 복잡한 구성 시나리오에 대한 지원이 필요한 경우
- 최적화 노력에도 불구하고 성능 문제가 지속됨

지원 센터에 문의하기 전에 MediaTailor CDN 정보를 수집합니다.

AWS Elemental MediaTailor CDN 통합 지원 사례에는 효과적인 문제 해결을 위한 특정 구성 및 진단 정보가 필요합니다. AWS Support에 문의하기 전에이 정보를 수집하여 문제를 신속하게 해결할 수 있습니다.

팁: 지원 사례를 생성하기 전에이 정보를 준비하면 해결 시간을 크게 줄이고 지원 엔지니어가 특정 구성을 이해하는 데 도움이 됩니다.

- MediaTailor 구성 세부 정보:
 - 구성 이름 및 AWS 리전

- 재생 구성 ARN
- ADS URL 및 통합 유형
- CDN 구성 세부 정보:
 - CDN 배포 ID 또는 구성 이름
 - 매니페스트 및 세그먼트에 대한 캐시 동작 구성
 - 오리진 구성 및 라우팅 규칙
- 오류 정보:
 - 특정 오류 메시지 및 HTTP 상태 코드
 - 문제 발생 시 타임스탬프
 - 문제를 보여주는 샘플 URLs
 - 문제와 관련된 CDN 및 MediaTailor 로그 항목
- 테스트 정보:
 - 문제 해결을 위해 이미 수행한 단계
 - 문제가 발생하는 디바이스 및 플레이어
 - 문제가 모든 콘텐츠 또는 특정 스트림에 영향을 미치는지 여부
 - 문제의 빈도 및 패턴(간접, 일관성, 시간 기반)

MediaTailor CDN 통합 지원 리소스 및 채널

AWS Elemental MediaTailor CDN 통합 지원은 여러 채널을 통해 사용할 수 있으므로 문제를 해결하고 구현을 최적화하는 데 도움이 됩니다.

- AWS 지원: 기술 지원을 위해 AWS Management Console을 통해 지원 사례 생성

액세스: [AWS 지원 센터](#)

- AWS re:Post: AWS관련 질문 및 커뮤니티 지원을 위한 커뮤니티 기반 Q&A 플랫폼

액세스: [AWS re:Post](#)

- AWS 설명서: MediaTailor 및 CDN 서비스에 대한 포괄적인 가이드

액세스: [MediaTailor 설명서](#) 및 [CloudFront 설명서](#)

- AWS 교육: 미디어 서비스 및 CDN 최적화를 위한 과정 및 인증

액세스: [AWS 훈련 및 인증](#)

- AWS 지식 센터: 일반적인 AWS 문제 및 모범 사례에 대해 큐레이션된 문서

액세스: [AWS 지식 센터](#)

- AWS Trusted Advisor: 최적화 및 모범 사례를 위한 자동화된 권장 사항

액세스: [AWS Trusted Advisor](#)

- AWS Personal Health Dashboard: 서비스 상태 및 유지 관리 알림

액세스: [AWS Personal Health Dashboard](#)

MediaTailor CDN 통합 지원 사례 모범 사례

AWS Elemental MediaTailor 이러한 모범 사례를 따르면 CDN 통합 지원 사례가 더 효율적으로 해결됩니다. 가장 빠른 해상도를 얻으려면:

- 비즈니스 영향에 따라 적절한 심각도 수준을 선택합니다.
- 초기 사례 제출 시 모든 관련 정보를 제공합니다.
- 특정 예제 및 복제 단계 포함
- 관련 로그 파일 및 구성 스크린샷 첨부
- 추가 정보에 대한 엔지니어 요청 지원에 즉시 대응

추가 지원 리소스:

- [AWS 지원 사례 관리](#) - 지원 사례 생성 및 관리 가이드
- [AWS 지원 플랜](#) - 지원 플랜 기능 및 응답 시간 비교
- [워크로드 지원 준비](#) - 운영 준비에 대한 Well-Architected 지침
- [AWS Well-Architected Framework](#) -에서 워크로드를 구축하고 운영하는 모범 사례 AWS

의 보안 AWS Elemental MediaTailor

의 클라우드 보안 AWS 이 최우선 순위입니다. AWS 고객은 보안에 가장 민감한 조직의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처의 이점을 누릴 수 있습니다.

보안은 AWS와 사용자 간의 공동 책임입니다. [공동 책임 모델](#)에서는 이 사항을 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드 보안 - AWS 는에서 AWS 서비스를 실행하는 인프라를 보호할 책임이 있습니다 AWS 클라우드. AWS 또한 안전하게 사용할 수 있는 서비스를 제공합니다. 타사 감사자는 [AWS 규정 준수 프로그램](#) 일환으로 보안의 효과를 정기적으로 테스트하고 확인합니다. 에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 규정 준수 프로그램 [AWS 제공 범위 내 서비스규정 준수 프로그램](#) 제공 범위 내 서비스를 AWS Elemental MediaTailor참조하세요.
- 클라우드의 보안 - 사용자의 책임은 사용하는 AWS 서비스에 따라 결정됩니다. 또한 귀하는 귀사의 데이터 민감도, 귀사의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 MediaTailor를 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 다음 주제에서는 보안 및 규정 준수 목표에 맞게 MediaTailor를 구성하는 방법을 보여줍니다. 또한 MediaTailor 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법을 알아봅니다.

주제

- [의 데이터 보호 AWS Elemental MediaTailor](#)
- [에 대한 자격 증명 및 액세스 관리 AWS Elemental MediaTailor](#)
- [에 대한 규정 준수 검증 AWS Elemental MediaTailor](#)
- [의 복원력 AWS Elemental MediaTailor](#)
- [MediaTailor의 인프라 보안](#)
- [교차 서비스 혼동된 대리자 방지](#)
- [MediaTailor에서 로깅 및 모니터링](#)

의 데이터 보호 AWS Elemental MediaTailor

AWS [공동 책임 모델](#) 은 AWS Elemental MediaTailor의 데이터 보호에 적용됩니다. 이 모델에 설명된 대로 AWS 는 모든를 실행하는 글로벌 인프라를 보호할 책임이 있습니다 AWS 클라우드. 사용자는 이

인프라에 호스팅되는 콘텐츠에 대한 통제 권한을 유지할 책임이 있습니다. 사용하는 AWS 서비스의 보안 구성과 관리 태스크에 대한 책임도 사용자에게 있습니다. 데이터 프라이버시에 대한 자세한 내용은 [데이터 프라이버시 FAQ](#) 참조하세요. 유럽의 데이터 보호에 대한 자세한 내용은 [일반 데이터 보호 규정\(GDPR\) 센터](#)를 참조하세요.

데이터 보호를 위해 자격 증명을 보호하고 AWS 계정 AWS IAM Identity Center 또는 AWS Identity and Access Management (IAM)를 사용하여 개별 사용자를 설정하는 것이 좋습니다. 이렇게 하면 개별 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 다중 인증(MFA)을 사용합니다.
- SSL/TLS를 사용하여 AWS 리소스와 통신합니다. TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- 를 사용하여 API 및 사용자 활동 로깅을 설정합니다 AWS CloudTrail. CloudTrail 추적을 사용하여 AWS 활동을 캡처하는 방법에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 [CloudTrail 추적 작업을 참조하세요](#).
- 내부의 모든 기본 보안 제어와 함께 AWS 암호화 솔루션을 사용합니다 AWS 서비스.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용합니다.
- 명령줄 인터페이스 또는 API를 AWS 통화에 액세스할 때 FIPS 140-3 검증 암호화 모듈이 필요한 경우 FIPS 엔드포인트를 사용합니다. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 [연방 정보 처리 표준\(FIPS\) 140-3](#)을 참조하세요.

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 형식 텍스트 필드에 입력하지 않는 것이 좋습니다. 여기에는 MediaTailor 또는 기타 AWS 서비스에서 콘솔 AWS CLI, API 또는 AWS SDKs를 사용하여 작업하는 경우가 포함됩니다. 이름에 사용되는 태그 또는 자유 형식 텍스트 필드에 입력하는 모든 데이터는 청구 또는 진단 로그에 사용될 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 자격 증명을 URL에 포함해서는 안 됩니다.

데이터 암호화

AWS Elemental MediaTailor는 콘텐츠 매니페스트 관리 또는 서버, CDNs 또는 플레이어와의 통신에서 데이터를 암호화하거나 해독하지 않습니다. MediaTailor는 고객 데이터 또는 기타 민감한 정보를 제공할 필요가 없습니다.

고객 계정 번호, 신용 카드 정보 또는 로그인 자격 증명과 같은 민감한 정보를 자유 형식 필드 또는 쿼리 파라미터에 입력하지 마세요. 이는 콘솔 AWS Elemental MediaTailor, API, SDKs 및 AWS Command

Line Interface ()를 포함한 모든 사용에 적용됩니다AWS CLI. 서비스에 입력하는 모든 데이터는 진단 로그에 포함하기 위해 선택될 수 있습니다.

외부 서버에 URL을 제공할 때 그 서버에 요청을 검증하기 위해 암호화되지 않은 자격 증명 정보를 URL에 포함시키지 마십시오.

에 대한 자격 증명 및 액세스 관리 AWS Elemental MediaTailor

AWS Identity and Access Management (IAM)는 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어하는 데 도움이 되는 AWS 서비스입니다. IAM 관리자는 MediaTailor 리소스를 사용할 수 있는 인증(로그인) 및 권한(권한 있음)을 받을 수 있는 사용자를 제어합니다. IAM은 추가 비용 없이 사용할 수 있는 AWS 서비스입니다.

주제

- [대상](#)
- [ID를 통한 인증](#)
- [정책을 사용하여 액세스 관리](#)
- [AWS Elemental MediaTailor 에서 IAM을 사용하는 방법](#)
- [에 대한 자격 증명 기반 정책 예제 AWS Elemental MediaTailor](#)
- [에 대한 리소스 기반 정책 예제 AWS Elemental MediaTailor](#)
- [AWS 에 대한 관리형 정책 AWS Elemental MediaTailor](#)
- [MediaTailor에 서비스 연결 역할 사용](#)
- [AWS Elemental MediaTailor 자격 증명 및 액세스 문제 해결](#)

대상

AWS Identity and Access Management (IAM)를 사용하는 방법은 역할에 따라 다릅니다.

- 서비스 사용자 - 기능에 액세스할 수 없는 경우 관리자에게 권한 요청([참조AWS Elemental MediaTailor 자격 증명 및 액세스 문제 해결](#))
- 서비스 관리자 - 사용자 액세스 결정 및 권한 요청 제출([AWS Elemental MediaTailor 에서 IAM을 사용하는 방법](#) 참조)
- IAM 관리자 - 액세스를 관리하기 위한 정책 작성([에 대한 자격 증명 기반 정책 예제 AWS Elemental MediaTailor](#) 참조)

ID를 통한 인증

인증은 자격 증명 자격 증명을 AWS 사용하여 로그인하는 방법입니다. AWS 계정 루트 사용자, IAM 사용자 또는 IAM 역할을 수임하여 인증되어야 합니다.

AWS IAM Identity Center (IAM Identity Center), Single Sign-On 인증 또는 Google/Facebook 자격 증명과 같은 자격 증명 소스의 자격 증명을 사용하여 페더레이션 자격 증명으로 로그인할 수 있습니다. 로그인하는 방법에 대한 자세한 내용은 AWS 로그인 사용 설명서의 [AWS 계정에 로그인하는 방법](#) 섹션을 참조하세요.

프로그래밍 방식 액세스를 위해서는 요청에 암호화 방식으로 서명할 수 있는 SDK 및 CLI를 AWS 제공합니다. 자세한 내용은 IAM 사용 설명서의 [API 요청용AWS Signature Version 4](#) 섹션을 참조하세요.

AWS 계정 루트 사용자

를 생성할 때 모든 AWS 서비스 및 리소스에 대한 완전한 액세스 권한이 있는 AWS 계정 theroot 사용자라는 하나의 로그인 자격 증명으로 AWS 계정시작합니다. 일상적인 태스크에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자 자격 증명이 필요한 작업은 IAM 사용 설명서의 [루트 사용자 자격 증명이 필요한 작업](#) 섹션을 참조하세요.

페더레이션 ID

가장 좋은 방법은 인간 사용자에게 자격 증명 공급자와의 페더레이션을 사용하여 임시 자격 증명을 AWS 서비스 사용하여 액세스하도록 요구하는 것입니다.

페더레이션 자격 증명은 엔터프라이즈 디렉터리, 웹 자격 증명 공급자 또는 자격 증명 소스의 자격 증명을 AWS 서비스 사용하여 Directory Service 에 액세스하는 사용자입니다. 페더레이션 ID는 임시 자격 증명을 제공하는 역할을 수임합니다.

중앙 집중식 액세스 관리를 위해 AWS IAM Identity Center를 추천합니다. 자세한 정보는 AWS IAM Identity Center 사용 설명서의 [What is IAM Identity Center?](#)를 참조하세요.

IAM 사용자 및 그룹

[IAM 사용자](#)는 단일 개인 또는 애플리케이션에 대한 특정 권한을 가진 ID입니다. 장기 자격 증명에 있는 IAM 사용자 대신 임시 자격 증명을 사용하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 [자격 증명 공급자와의 페더레이션을 사용하여 임시 자격 증명을 AWS 사용하여 액세스하도록 인간 사용자에게 요구하기](#)를 참조하세요.

[IAM 그룹](#)은 IAM 사용자 모음을 지정하고 대규모 사용자 집합에 대한 관리 권한을 더 쉽게 만듭니다. 자세한 내용은 IAM 사용 설명서의 [IAM 사용자 사용 사례](#) 섹션을 참조하세요.

IAM 역할

[IAM 역할](#)은 임시 자격 증명을 제공하는 특정 권한이 있는 자격 증명입니다. [사용자에서 IAM 역할\(콘솔\)로 전환하거나 또는 API 작업을 호출하여 역할을](#) 수입할 수 있습니다. AWS CLI AWS 자세한 내용은 IAM 사용 설명서의 [역할 수입 방법](#)을 참조하세요.

IAM 역할은 페더레이션 사용자 액세스, 임시 IAM 사용자 권한, 교차 계정 액세스, 교차 서비스 액세스 및 Amazon EC2에서 실행되는 애플리케이션에 유용합니다. 자세한 내용은 IAM 사용 설명서의 [교차 계정 리소스 액세스](#)를 참조하세요.

정책을 사용하여 액세스 관리

정책을 AWS 생성하고 자격 증명 또는 리소스에 연결하여 AWS 에서 액세스를 제어합니다. 정책은 자격 증명 또는 리소스와 연결될 때 권한을 정의합니다.는 보안 주체가 요청할 때 이러한 정책을 AWS 평가합니다. 대부분의 정책은 JSON 문서 AWS 로 저장됩니다. JSON 정책 문서에 대한 자세한 내용은 IAM 사용 설명서의 [JSON 정책 개요](#) 섹션을 참조하세요.

정책을 사용하여 관리자는 어떤 보안 주체가 어떤 리소스에 대해 어떤 조건에서 작업을 수행할 수 있는지 정의하여 누가 무엇을 액세스할 수 있는지 지정합니다.

기본적으로 사용자 및 역할에는 어떠한 권한도 없습니다. IAM 관리자는 IAM 정책을 생성하고 사용자가 수입할 수 있는 역할에 추가합니다. IAM 정책은 작업을 수행하기 위해 사용하는 방법과 관계없이 작업에 대한 권한을 정의합니다.

ID 기반 정책

ID 기반 정책은 ID(사용자, 사용자 그룹 또는 역할)에 연결하는 JSON 권한 정책 문서입니다. 이러한 정책은 자격 증명이 수행할 수 있는 작업, 대상 리소스 및 이에 관한 조건을 제어합니다. ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서에서 [고객 관리형 정책으로 사용자 지정 IAM 권한 정의](#)를 참조하세요.

ID 기반 정책은 인라인 정책(단일 ID에 직접 포함) 또는 관리형 정책(여러 ID에 연결된 독립 실행형 정책)일 수 있습니다. 관리형 정책 또는 인라인 정책을 선택하는 방법을 알아보려면 IAM 사용 설명서의 [관리형 정책 및 인라인 정책 중에서 선택](#) 섹션을 참조하세요.

리소스 기반 정책

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 예를 들어 IAM 역할 신뢰 정책 및 Amazon S3 버킷 정책이 있습니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한

정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 리소스 기반 정책에서 [보안 주체를 지정](#)해야 합니다.

리소스 기반 정책은 해당 서비스에 있는 인라인 정책입니다. 리소스 기반 정책에서는 IAM의 AWS 관리형 정책을 사용할 수 없습니다.

기타 정책 유형

AWS 는 보다 일반적인 정책 유형에서 부여한 최대 권한을 설정할 수 있는 추가 정책 유형을 지원합니다.

- 권한 경계 - ID 기반 정책에서 IAM 엔터티에 부여할 수 있는 최대 권한을 설정합니다. 자세한 정보는 IAM 사용 설명서의 [IAM 엔터티의 권한 범위](#)를 참조하세요.
- 서비스 제어 정책(SCP) - AWS Organizations내 조직 또는 조직 단위에 대한 최대 권한을 지정합니다. 자세한 내용은 AWS Organizations 사용 설명서의 [서비스 제어 정책](#)을 참조하세요.
- 리소스 제어 정책(RCP) - 계정의 리소스에 사용할 수 있는 최대 권한을 설정합니다. 자세한 내용은 AWS Organizations 사용 설명서의 [리소스 제어 정책\(RCP\)](#)을 참조하세요.
- 세션 정책 - 역할 또는 페더레이션 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 자세한 내용은 IAM 사용 설명서의 [세션 정책](#)을 참조하세요.

여러 정책 유형

여러 정책 유형이 요청에 적용되는 경우, 결과 권한은 이해하기가 더 복잡합니다. 에서 여러 정책 유형이 관련될 때 요청을 허용할지 여부를 AWS 결정하는 방법을 알아보려면 IAM 사용 설명서의 [정책 평가 로직](#)을 참조하세요.

AWS Elemental MediaTailor 에서 IAM을 사용하는 방법

IAM을 사용하여 MediaTailor에 대한 액세스를 관리하기 전에 MediaTailor에서 사용할 수 있는 IAM 기능을 알아봅니다.

에서 사용할 수 있는 IAM 기능 AWS Elemental MediaTailor

IAM 특성	MediaTailor 지원
자격 증명 기반 정책	예
리소스 기반 정책	예

IAM 특성	MediaTailor 지원
정책 작업	예
정책 리소스	아니요
정책 조건 키(서비스별)	예
ACL	아니요
ABAC(정책 내 태그)	부분적
임시 자격 증명	예
엔터티 권한	예
서비스 역할	아니요
서비스 연결 역할	예

MediaTailor 및 기타 AWS 서비스에서 대부분의 IAM 기능을 사용하는 방법을 전체적으로 알아보려면 IAM 사용 설명서의 [AWS IAM으로 작업하는 서비스](#)를 참조하세요.

MediaTailor에 대한 자격 증명 기반 정책

ID 기반 정책 지원: 예

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 ID에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자 및 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서에서 [고객 관리형 정책으로 사용자 지정 IAM 권한 정의](#)를 참조하세요.

IAM ID 기반 정책을 사용하면 허용되거나 거부되는 작업과 리소스뿐 아니라 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. JSON 정책에서 사용할 수 있는 모든 요소에 대해 알아보려면 IAM 사용 설명서의 [IAM JSON 정책 요소 참조](#)를 참조하세요.

MediaTailor의 자격 증명 기반 정책 예제

MediaTailor 자격 증명 기반 정책의 예를 보려면 [섹션을 참조하세요에 대한 자격 증명 기반 정책 예제 AWS Elemental MediaTailor](#).

MediaTailor 내의 리소스 기반 정책

리소스 기반 정책 지원: 예

MediaTailor 서비스는 한 가지 유형의 리소스 기반 정책만 지원합니다. 채널에 연결되어 있으므로 채널 정책이라고 합니다. 이 정책은 채널에서 작업을 수행할 수 있는 보안 주체를 정의합니다.

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 리소스 기반 정책의 예제는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 [보안 주체를 지정](#)해야 합니다. 보안 주체에는 계정, 사용자, 역할, 페더레이션 사용자 또는가 포함될 수 있습니다 AWS 서비스.

교차 계정 액세스를 활성화하려는 경우, 전체 계정이나 다른 계정의 IAM 개체를 리소스 기반 정책의 보안 주체로 지정할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [IAM에서 교차 계정 리소스 액세스](#)를 참조하세요.

리소스 기반 정책을 채널에 연결하는 방법은 섹션을 참조하세요 [MediaTailor 콘솔을 사용하여 채널 생성](#).

MediaTailor 내의 리소스 기반 정책 예제

MediaTailor 리소스 기반 정책의 예를 보려면 섹션을 참조하세요 [에 대한 리소스 기반 정책 예제 AWS Elemental MediaTailor](#).

MediaTailor에 대한 정책 작업

정책 작업 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

JSON 정책의 Action요소는 정책에서 액세스를 허용하거나 거부하는 데 사용할 수 있는 작업을 설명합니다. 연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함하세요.

MediaTailor 작업 목록을 보려면 서비스 승인 참조의에서 [정의한 작업을 AWS Elemental MediaTailor](#) 참조하세요.

MediaTailor의 정책 작업은 작업 앞에 다음 접두사를 사용합니다.

```
mediatailor
```

단일 문에서 여러 작업을 지정하려면 쉼표로 구분합니다.

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

MediaTailor 자격 증명 기반 정책의 예를 보려면 섹션을 참조하세요 [에 대한 자격 증명 기반 정책 예제 AWS Elemental MediaTailor](#).

MediaTailor에 대한 정책 리소스

정책 리소스 지원: 아니요

AWS Elemental MediaTailor 는 정책에서 리소스 ARNs 지정을 지원하지 않습니다.

MediaTailor에 사용되는 정책 조건 키

서비스별 정책 조건 키 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

Condition 요소는 정의된 기준에 따라 문이 실행되는 시기를 지정합니다. 같음(equals) 또는 미만(less than)과 같은 [조건 연산자](#)를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수 있습니다. 모든 AWS 전역 조건 키를 보려면 IAM 사용 설명서의 [AWS 전역 조건 컨텍스트 키를](#) 참조하세요.

MediaTailor 조건 키 목록은 서비스 승인 참조의 [에 대한 조건 키를 AWS Elemental MediaTailor](#) 참조하세요. 조건 키를 사용할 수 있는 작업 및 리소스를 알아보려면 [에서 정의한 작업을 AWS Elemental MediaTailor](#) 참조하세요.

AWS Elemental MediaTailor 는 서비스별 조건 키를 제공하지 않지만 일부 전역 조건 키 사용을 지원합니다. 모든 AWS 전역 조건 키를 보려면 AWS Identity and Access Management 사용 설명서의 [AWS 전역 조건 컨텍스트 키를](#) 참조하세요.

MediaTailorACLs

ACL 지원: 아니요

액세스 제어 목록(ACL)은 어떤 보안 주체(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는지를 제어합니다. ACL은 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

MediaTailor를 사용한 ABAC

ABAC 지원(정책의 태그): 부분적

속성 기반 액세스 제어(ABAC)는 태그라고 불리는 속성을 기반으로 권한을 정의하는 권한 부여 전략입니다. IAM 엔터티 및 AWS 리소스에 태그를 연결한 다음 보안 주체의 태그가 리소스의 태그와 일치할 때 작업을 허용하는 ABAC 정책을 설계할 수 있습니다.

태그에 근거하여 액세스를 제어하려면 `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` 또는 `aws:TagKeys` 조건 키를 사용하여 정책의 [조건 요소](#)에 태그 정보를 제공합니다.

서비스가 모든 리소스 유형에 대해 세 가지 조건 키를 모두 지원하는 경우, 값은 서비스에 대해 예입니다. 서비스가 일부 리소스 유형에 대해서만 세 가지 조건 키를 모두 지원하는 경우, 값은 부분적입니다.

ABAC에 대한 자세한 내용은 IAM 사용 설명서의 [ABAC 권한 부여를 통한 권한 정의](#)를 참조하세요. ABAC 설정 단계가 포함된 자습서를 보려면 IAM 사용 설명서의 [속성 기반 액세스 제어\(ABAC\) 사용](#)을 참조하세요.

MediaTailor의 경우 부분 값을 사용합니다.

MediaTailor에서 임시 자격 증명 사용

임시 자격 증명 지원: 예

임시 자격 증명은 AWS 리소스에 대한 단기 액세스를 제공하며 페더레이션 또는 전환 역할을 사용할 때 자동으로 생성됩니다. 장기 액세스 키를 사용하는 대신 임시 자격 증명을 동적으로 생성하는 것이 AWS 좋습니다. 자세한 내용은 IAM 사용 설명서의 [IAM의 임시 보안 자격 증명](#) 및 [IAM으로 작업하는 AWS 서비스](#) 섹션을 참조하세요.

MediaTailor에 대한 교차 서비스 보안 주체 권한

전달 액세스 세션(FAS) 지원: 예

전달 액세스 세션(FAS)은를 호출하는 보안 주체의 권한을 다운스트림 서비스에 AWS 서비스 대한 요청과 AWS 서비스함께 사용합니다. FAS 요청 시 정책 세부 정보는 [전달 액세스 세션](#)을 참조하세요.

MediaTailor의 서비스 역할

서비스 역할 지원: 아니요

AWS Elemental MediaTailor 는 서비스 역할을 지원하지 않습니다.

MediaTailor에 대한 서비스 연결 역할

서비스 연결 역할 지원: 예

서비스 연결 역할은에 연결된 서비스 역할의 한 유형입니다 AWS 서비스. 서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은에 나타나 AWS 계정 며 서비스가 소유합니다. IAM 관리자는 서비스 연결 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.

MediaTailor 서비스 연결 역할 생성 또는 관리에 대한 자세한 내용은 섹션을 참조하세요 [MediaTailor에 서비스 연결 역할 사용](#).

에 대한 자격 증명 기반 정책 예제 AWS Elemental MediaTailor

기본적으로 사용자 및 역할에는 MediaTailor 리소스를 생성하거나 수정할 수 있는 권한이 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다.

이러한 예제 JSON 정책 문서를 사용하여 IAM ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 [IAM 정책 생성\(콘솔\)](#)을 참조하세요.

각 리소스 유형에 대한 ARNs 형식을 포함하여 MediaTailor에서 정의한 작업 및 리소스 유형에 대한 자세한 내용은 서비스 승인 참조의 [에 대한 작업, 리소스 및 조건 키를 AWS Elemental MediaTailor](#) 참조하세요.

주제

- [정책 모범 사례](#)
- [MediaTailor 콘솔 사용](#)
- [사용자가 자신의 고유한 권한을 볼 수 있도록 허용](#)

정책 모범 사례

자격 증명 기반 정책에 따라 계정에서 사용자가 MediaTailor 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부가 결정됩니다. 이 작업으로 인해 AWS 계정에 비용이 발생할 수 있습니다. ID 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따르세요.

- AWS 관리형 정책을 시작하고 최소 권한으로 전환 - 사용자 및 워크로드에 권한 부여를 시작하려면 많은 일반적인 사용 사례에 대한 권한을 부여하는 AWS 관리형 정책을 사용합니다. 에서 사용할 수 있습니다 AWS 계정. 사용 사례에 맞는 AWS 고객 관리형 정책을 정의하여 권한을 추가로 줄이는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 [AWS 관리형 정책](#) 또는 [AWS 직무에 대한 관리형 정책](#)을 참조하세요.
- 최소 권한 적용 - IAM 정책을 사용하여 권한을 설정하는 경우, 작업을 수행하는 데 필요한 권한만 부여합니다. 이렇게 하려면 최소 권한으로 알려진 특정 조건에서 특정 리소스에 대해 수행할 수 있는 작업을 정의합니다. IAM을 사용하여 권한을 적용하는 방법에 대한 자세한 정보는 IAM 사용 설명서에 있는 [IAM의 정책 및 권한](#)을 참조하세요.
- IAM 정책의 조건을 사용하여 액세스 추가 제한 - 정책에 조건을 추가하여 작업 및 리소스에 대한 액세스를 제한할 수 있습니다. 예를 들어, SSL을 사용하여 모든 요청을 전송해야 한다고 지정하는 정책 조건을 작성할 수 있습니다. AWS 서비스와 같은 특정을 통해 사용되는 경우 조건을 사용하여 서비스 작업에 대한 액세스 권한을 부여할 수도 있습니다 CloudFormation. 자세한 내용은 IAM 사용 설명서의 [IAM JSON 정책 요소: 조건](#)을 참조하세요.
- IAM Access Analyzer를 통해 IAM 정책을 확인하여 안전하고 기능적인 권한 보장 - IAM Access Analyzer에서는 IAM 정책 언어(JSON)와 모범 사례가 정책에서 준수되도록 새로운 및 기존 정책을 확인합니다. IAM Access Analyzer는 100개 이상의 정책 확인 항목과 실행 가능한 추천을 제공하여 안전하고 기능적인 정책을 작성하도록 돕습니다. 자세한 내용은 IAM 사용 설명서의 [IAM Access Analyzer에서 정책 검증](#)을 참조하세요.
- 다중 인증(MFA) 필요 -에서 IAM 사용자 또는 루트 사용자가 필요한 시나리오가 있는 경우 추가 보안을 위해 MFA를 AWS 계정킵니다. API 작업을 직접적으로 호출할 때 MFA가 필요하다면 정책에 MFA 조건을 추가합니다. 자세한 내용은 IAM 사용 설명서의 [MFA를 통한 보안 API 액세스](#)를 참조하세요.

IAM의 모범 사례에 대한 자세한 내용은 IAM 사용 설명서의 [IAM의 보안 모범 사례](#)를 참조하세요.

MediaTailor 콘솔 사용

AWS Elemental MediaTailor 콘솔에 액세스하려면 최소 권한 집합이 있어야 합니다. 이러한 권한은에서 MediaTailor 리소스에 대한 세부 정보를 나열하고 볼 수 있도록 허용해야 합니다 AWS 계정. 최소 필수 권한보다 더 제한적인 ID 기반 정책을 생성하는 경우, 콘솔이 해당 정책에 연결된 엔티티(사용자 또는 역할)에 대해 의도대로 작동하지 않습니다.

AWS CLI 또는 AWS API만 호출하는 사용자에게는 최소 콘솔 권한을 허용할 필요가 없습니다. 대신, 수행하려는 API 작업과 일치하는 작업에만 액세스할 수 있도록 합니다.

사용자와 역할이 여전히 MediaTailor 콘솔을 사용할 수 있도록 하려면 MediaTailor *ConsoleAccess* 또는 *ReadOnly* AWS 관리형 정책도 엔터티에 연결합니다. 자세한 내용은 IAM 사용 설명서의 [사용자에게 권한 추가](#)를 참조하세요.

사용자가 자신의 고유한 권한을 볼 수 있도록 허용

이 예제는 IAM 사용자가 자신의 사용자 ID에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 생성하는 방법을 보여줍니다. 이 정책에는 콘솔에서 또는 AWS CLI 또는 AWS API를 사용하여 프로그래밍 방식으로 이 작업을 완료할 수 있는 권한이 포함됩니다.

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

에 대한 리소스 기반 정책 예제 AWS Elemental MediaTailor

리소스 기반 정책을 채널에 연결하는 방법은 섹션을 참조하세요 [MediaTailor 콘솔을 사용하여 채널 생성](#).

주제

- [익명 액세스](#)
- [크로스 계정 액세스](#)

익명 액세스

다음 Allow 정책을 고려하세요. 이 정책이 적용되면 MediaTailor는 정책의 채널 리소스에 대한 `mediatailor:GetManifest` 작업에 익명으로 액세스할 수 있습니다. 이는 `region`이 AWS 리전이 고, `accountID`가 AWS 계정 ID이고, `channelName`이 채널 리소스의 이름인 경우에 발생합니다.

크로스 계정 액세스

다음 Allow 정책을 고려하세요. 이 정책이 적용되면 MediaTailor는 계정 간에 정책의 채널 리소스에 대한 `mediatailor:GetManifest` 작업을 허용합니다. 이는 `##`이 AWS 리전이고, `accountID`가 AWS 계정 ID이고, `channelName`이 채널 리소스의 이름인 경우에 발생합니다.

AWS 에 대한 관리형 정책 AWS Elemental MediaTailor

사용자, 그룹 및 역할에 권한을 추가하려면 직접 정책을 작성하는 것보다 AWS 관리형 정책을 사용하는 것이 더 쉽습니다. 팀에 필요한 권한만 제공하는 [IAM 고객 관리형 정책을 생성](#)하기 위해서는 시간과 전문 지식이 필요합니다. 빠르게 시작하려면 AWS 관리형 정책을 사용할 수 있습니다. 이 정책은 일반적인 사용 사례를 다루며 사용자의 AWS 계정에서 사용할 수 있습니다. AWS 관리형 정책에 대한 자세한 내용은 IAM 사용 설명서의 [AWS 관리형 정책을](#) 참조하세요.

AWS 서비스는 AWS 관리형 정책을 유지 관리하고 업데이트합니다. AWS 관리형 정책에서는 권한을 변경할 수 없습니다. 서비스에서 때때로 추가 권한을 AWS 관리형 정책에 추가하여 새로운 기능을 지원합니다. 이 유형의 업데이트는 정책이 연결된 모든 ID(사용자, 그룹 및 역할)에 적용됩니다. 서비스는

새로운 기능이 시작되거나 새 작업을 사용할 수 있을 때 AWS 관리형 정책에 업데이트됩니다. 서비스는 AWS 관리형 정책에서 권한을 제거하지 않으므로 정책 업데이트로 인해 기존 권한이 손상되지 않습니다.

또한 여러 서비스에 걸쳐 있는 직무에 대한 관리형 정책을 AWS 지원합니다. 예를 들어 ReadOnlyAccess AWS 관리형 정책은 모든 AWS 서비스 및 리소스에 대한 읽기 전용 액세스를 제공합니다. 서비스가 새 기능을 시작하면 새 작업 및 리소스에 대한 읽기 전용 권한을 AWS 추가합니다. 직무 정책의 목록과 설명은 IAM 사용 설명서의 [직무에 관한 AWS 관리형 정책](#)을 참조하세요.

AWS 관리형 정책: AWSElementalMediaTailorFullAccess

AWSElementalMediaTailorFullAccess 정책을 IAM ID에 연결할 수 있습니다. 재생 구성과 프로그램 및 채널과 같은 채널 어셈블리 리소스를 생성하고 관리해야 하는 사용자에게 유용합니다. 이 정책은에 대한 전체 액세스를 허용하는 권한을 부여합니다 AWS Elemental MediaTailor. 이러한 사용자는 MediaTailor 리소스를 생성, 업데이트 및 삭제할 수 있습니다.

이 정책의 권한을 보려면 AWS 관리형 정책 참조의 [AWSElementalMediaTailorFullAccess](#)를 참조하세요.

AWS 관리형 정책: AWSElementalMediaTailorReadOnly

AWSElementalMediaTailorReadOnly 정책을 IAM ID에 연결할 수 있습니다. 프로그램 및 채널과 같은 재생 구성 및 채널 어셈블리 리소스를 확인해야 하는 사용자에게 유용합니다. 이 정책은에 대한 읽기 전용 액세스를 허용하는 권한을 부여합니다 AWS Elemental MediaTailor. 이러한 사용자는 MediaTailor 리소스를 생성, 업데이트 또는 삭제할 수 없습니다.

이 정책의 권한을 보려면 AWS 관리형 정책 참조의 [AWSElementalMediaTailorReadOnly](#)를 참조하세요.

AWS 관리형 정책에 대한 MediaTailor 업데이트

이 서비스가 이러한 변경 사항을 추적하기 시작한 이후부터 MediaTailor의 AWS 관리형 정책 업데이트에 대한 세부 정보를 봅니다. 이 페이지의 변경 사항에 대한 자동 알림을 받으려면 MediaTailor에서 RSS 피드를 구독하세요 [에 대한 문서 기록 AWS Elemental MediaTailor](#).

변경	설명	Date
MediaTailor에 새 관리형 정책 추가	MediaTailor는 다음과 같은 관리형 정책을 추가했습니다.	2021년 11월 24일

변경	설명	Date
	• AWSElementalMediaTailorReadOnly • AWSElementalMediaTailorFullAccess	
MediaTailor에서 변경 사항 추적 시작	MediaTailor가 AWS 관리형 정책에 대한 변경 사항 추적을 시작했습니다.	2021년 11월 24일

MediaTailor에 서비스 연결 역할 사용

AWS Elemental MediaTailor 는 AWS Identity and Access Management (IAM) [서비스 연결 역할](#)을 사용합니다. 서비스 연결 역할은 MediaTailor에 직접 연결된 고유한 유형의 IAM 역할입니다. 서비스 연결 역할은 MediaTailor에서 사전 정의하며 서비스가 사용자를 대신하여 다른 AWS 서비스를 호출하는 데 필요한 모든 권한을 포함합니다.

필요한 권한을 수동으로 추가할 필요가 없으므로 서비스 연결 역할을 사용하면 MediaTailor를 더 쉽게 설정할 수 있습니다. MediaTailor는 서비스 연결 역할의 권한을 정의하며, 달리 정의되지 않은 한 MediaTailor만 해당 역할을 수임할 수 있습니다. 정의된 권한에는 신뢰 정책과 권한 정책이 포함되며 이 권한 정책은 다른 IAM 엔터티에 연결할 수 없습니다.

먼저 관련 리소스를 삭제한 후에만 서비스 연결 역할을 삭제할 수 있습니다. 이렇게 하면 리소스에 대한 액세스 권한을 실수로 제거할 수 없으므로 MediaTailor 리소스가 보호됩니다.

서비스 연결 역할을 지원하는 기타 서비스에 대한 자세한 내용은 [IAM으로 작업하는AWS 서비스](#)를 참조해 서비스 연결 역할 열이 예(Yes)인 서비스를 찾으세요. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 예(Yes) 링크를 선택합니다.

MediaTailor 서비스 연결 역할에 지원되는 리전

MediaTailor는 서비스를 사용할 수 있는 모든 리전에서 서비스 연결 역할 사용을 지원합니다. 자세한 설명은 [AWS 리전 및 엔드포인트](#)를 참조하세요.

주제

- [MediaTailor에 대한 서비스 연결 역할 권한](#)
- [MediaTailor에 대한 서비스 연결 역할 생성](#)

- [MediaTailor에 대한 서비스 연결 역할 편집](#)
- [MediaTailor에 대한 서비스 연결 역할 삭제](#)

MediaTailor에 대한 서비스 연결 역할 권한

MediaTailor는 AWSServiceRoleForMediaTailor라는 서비스 연결 역할을 사용합니다. MediaTailor는 이 서비스 연결 역할을 사용하여 CloudWatch를 호출하여 로그 그룹, 로그 스트림 및 로그 이벤트를 생성하고 관리합니다. 이 서비스 연결 역할은 관리형 정책 AWSMediaTailorServiceRolePolicy에 연결됩니다.

AWSServiceRoleForMediaTailor 서비스 연결 역할은 역할을 수임하기 위해 다음 서비스를 신뢰합니다.

- `mediatailor.amazonaws.com`

역할 권한 정책은 MediaTailor가 지정된 리소스에서 다음 작업을 완료하도록 허용합니다.

- 작업: `arn:aws:logs:*:*:log-group:/aws/MediaTailor/*:log-stream:*`에 대한 `logs:PutLogEvents`
- 작업: `arn:aws:logs:*:*:log-group:/aws/MediaTailor/*`에 대한 `logs:CreateLogStream`, `logs:CreateLogGroup`, `logs:DescribeLogGroups`, `logs:DescribeLogStreams`

IAM 엔터티(사용자, 그룹, 역할 등)가 서비스 링크 역할을 생성하고 편집하거나 삭제할 수 있도록 권한을 구성할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [서비스 연결 역할 권한](#) 섹션을 참조하세요.

MediaTailor에 대한 서비스 연결 역할 생성

서비스 연결 역할은 수동으로 생성할 필요가 없습니다. AWS Management Console, AWS Command Line Interface (AWS CLI) 또는 AWS API에서 세션 로깅을 활성화하면 MediaTailor가 서비스 연결 역할을 생성합니다.

Important

이러한 서비스 연결 역할은 해당 역할이 지원하는 기능을 사용하는 다른 서비스에서 작업을 완료했을 경우 계정에 나타날 수 있습니다. 또한 서비스 연결 역할을 지원하기 시작한 2021년 9월 15일 이전에 MediaTailor 서비스를 사용 중이었다면 MediaTailor는 계정에

AWSServiceRoleForMediaTailor 역할을 생성했습니다. 자세한 내용은 [내 IAM 계정에 표시되는 새 역할](#)을 참조하세요.

이 서비스 연결 역할을 삭제했다가 다시 생성해야 하는 경우 동일한 프로세스를 사용하여 계정에서 역할을 다시 생성할 수 있습니다. 세션 로깅을 활성화하면 MediaTailor가 서비스 연결 역할을 다시 생성합니다.

IAM 콘솔을 사용하여 MediaTailor 사용 사례로 서비스 연결 역할을 생성할 수도 있습니다. AWS CLI 또는 AWS API에서 서비스 이름을 사용하여 `mediatailor.amazonaws.com` 서비스 연결 역할을 생성합니다. 자세한 내용은 IAM 사용 설명서의 [서비스 연결 역할 생성](#) 섹션을 참조하세요. 이 서비스 연결 역할을 삭제하면 동일한 프로세스를 사용하여 역할을 다시 생성할 수 있습니다.

MediaTailor에 대한 서비스 연결 역할 편집

MediaTailor에서는 AWSServiceRoleForMediaTailor 서비스 연결 역할을 편집할 수 없습니다. 서비스 연결 역할을 생성한 후에는 다양한 개체가 역할을 참조할 수 있기 때문에 역할 이름을 변경할 수 없습니다. 하지만 IAM을 사용하여 역할의 설명을 편집할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [서비스 연결 역할 편집](#)을 참조하세요.

MediaTailor에 대한 서비스 연결 역할 삭제

서비스 연결 역할이 필요한 기능 또는 서비스가 더 이상 필요 없는 경우에는 해당 역할을 삭제하는 것이 좋습니다. 따라서 적극적으로 모니터링하거나 유지하지 않는 미사용 엔터티가 없도록 합니다. 단, 서비스 연결 역할에 대한 리소스를 먼저 정리해야 수동으로 삭제할 수 있습니다.

Note

리소스를 정리하려고 할 때 MediaTailor 서비스가 역할을 사용하는 경우 삭제에 실패할 수 있습니다. 이 문제가 발생하면 몇 분 기다렸다가 작업을 다시 시도하세요.

MediaTailor 리소스를 정리하려면 AWSServiceRoleForMediaTailor

- 로그 구성을 위해 MediaTailor에서 생성한 서비스 연결 역할을 삭제하려면 먼저 계정의 모든 로그 구성을 비활성화해야 합니다. 로그 구성을 비활성화하려면 활성화된 백분율 값을 0으로 설정합니다. 그러면 해당 재생 구성을 로깅하는 모든 세션이 꺼집니다. 자세한 내용은 [로그 구성 비활성화](#) 단원을 참조하십시오.

IAM을 사용하여 수동으로 서비스 연결 역할을 삭제하려면 다음을 수행하세요.

IAM 콘솔, AWS Command Line Interface (AWS CLI) 또는 AWS API를 사용하여 AWSServiceRoleForMediaTailor 서비스 연결 역할을 삭제합니다. 자세한 내용은 [IAM 사용 설명서](#)의 서비스 연결 역할 삭제를 참조하세요.

AWS Elemental MediaTailor 자격 증명 및 액세스 문제 해결

다음 정보를 사용하여 MediaTailor 및 IAM 작업 시 발생할 수 있는 일반적인 문제를 진단하고 수정할 수 있습니다.

주제

- [MediaTailor에서 작업을 수행할 권한이 없음](#)
- [iam:PassRole을 수행하도록 인증되지 않음](#)
- [내 외부의 사람이 내 MediaTailor 리소스 AWS 계정 에 액세스하도록 허용하고 싶습니다.](#)
- [MediaTailor가 내 Amazon S3 버킷에 액세스할 수 없음](#)

MediaTailor에서 작업을 수행할 권한이 없음

작업을 수행할 권한이 없다는 오류가 표시되면 작업을 수행할 수 있도록 정책을 업데이트해야 합니다.

다음의 예제 오류는 mateojackson IAM 사용자가 콘솔을 사용하여 가상 *my-example-widget* 리소스에 대한 세부 정보를 보려고 하지만 가상 mediatailor:*GetWidget* 권한이 없을 때 발생합니다.

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

이 경우, mediatailor:*GetWidget* 작업을 사용하여 *my-example-widget* 리소스에 액세스할 수 있도록 mateojackson 사용자 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

iam:PassRole을 수행하도록 인증되지 않음

iam:PassRole 작업을 수행할 권한이 없다는 오류가 수신되면 MediaTailor에 역할을 전달할 수 있도록 정책을 업데이트해야 합니다.

일부 AWS 서비스에서는 새 서비스 역할 또는 서비스 연결 역할을 생성하는 대신 기존 역할을 해당 서비스에 전달할 수 있습니다. 이렇게 하려면 역할을 서비스에 전달할 권한이 있어야 합니다.

다음 예제 오류는 라는 IAM 사용자가 콘솔을 사용하여 MediaTailor에서 작업을 수행하려고 marymajor 할 때 발생합니다. 하지만 작업을 수행하려면 서비스 역할이 부여한 권한이 서비스에 있어야 합니다. Mary는 서비스에 역할을 전달할 권한이 없습니다.

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

이 경우, Mary가 iam:PassRole작업을 수행할 수 있도록 Mary의 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

내 외부의 사람이 내 MediaTailor 리소스 AWS 계정 에 액세스하도록 허용하고 싶습니다.

다른 계정의 사용자 또는 조직 외부의 사람이 리소스에 액세스할 때 사용할 수 있는 역할을 생성할 수 있습니다. 역할을 수임할 신뢰할 수 있는 사람을 지정할 수 있습니다. 리소스 기반 정책 또는 액세스 제어 목록(ACL)을 지원하는 서비스의 경우, 이러한 정책을 사용하여 다른 사람에게 리소스에 대한 액세스 권한을 부여할 수 있습니다.

자세한 내용은 다음을 참조하세요.

- MediaTailor가 이러한 기능을 지원하는지 여부를 알아보려면 섹션을 참조하세요 [AWS Elemental MediaTailor 에서 IAM을 사용하는 방법](#).
- 소유 AWS 계정 한의 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 [IAM 사용 설명서의 소유한 다른의 IAM 사용자에게 액세스 권한 제공을 참조 AWS 계정 하세요](#).
- 타사에 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 [타사가 AWS 계정 소유한에 대한 액세스 권한 제공을](#) AWS 계정참조하세요.
- ID 페더레이션을 통해 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 [외부에서 인증된 사용자에게 액세스 권한 제공\(ID 페더레이션\)](#)을 참조하세요.
- 크로스 계정 액세스에 대한 역할과 리소스 기반 정책 사용의 차이점을 알아보려면 IAM 사용 설명서의 [IAM의 크로스 계정 리소스 액세스](#)를 참조하세요.

MediaTailor가 내 Amazon S3 버킷에 액세스할 수 없음

MediaTailor가 Amazon S3 버킷의 콘텐츠에 액세스하려고 할 때 오류를 반환하는 경우 다음과 같은 일반적인 원인을 확인합니다.

Amazon S3 권한 문제

오류 증상:

- 502 BadGatewayException Amazon S3 관련 메시지 사용
- 403 Forbidden MediaTailor 로그의 오류
- Amazon S3 오리진에서 콘텐츠 로드 실패

해결 방법:

1. MediaTailor 서비스 역할에 필요한 Amazon S3 권한이 있는지 확인합니다.
2. Amazon S3 버킷 정책이 MediaTailor 액세스를 허용하는지 확인합니다.

HTTPS 액세스 문제

오류 증상:

- SSL/TLS 핸드셰이크 실패
- 502 Bad Gateway HTTPS Amazon S3 URLs 액세스하는 경우

해결 방법:

1. Amazon S3 버킷 정책에 HTTPS가 필요한지 확인합니다.
2. MediaTailor 구성이 Amazon S3 콘텐츠에 HTTPS URLs을 사용하는지 확인합니다.

교차 계정 Amazon S3 액세스

오류 증상:

- Access Denied Amazon S3 버킷이 다른 계정에 있는 경우의 오류
- MediaTailor는 교차 계정 역할을 수임할 수 없습니다.

해결 방법:

1. 콘텐츠 계정의 Amazon S3 버킷 정책을 업데이트하여 MediaTailor의 서비스 역할을 신뢰합니다.
2. MediaTailor의 서비스 역할에 교차 계정 권한 추가

3. 두 계정 모두에 적절한 신뢰 관계가 구성되어 있는지 확인합니다.

에 대한 규정 준수 검증 AWS Elemental MediaTailor

새로운 AWS 가입 환경은 규제 워크로드용으로 설계되지 않았습니다. 새로운 AWS 가입 환경을 사용하고 있지만 규제 대상 워크로드 AWS 에를 사용하려는 경우 AWS 환경에 [가입 AWS \(고급\)](#)하거나 [고급 기능을 활성화](#)할 수 있습니다.

AWS 서비스 가 특정 규정 준수 프로그램의 범위 내에 있는지 알아보려면 [AWS 서비스 규정 준수 프로그램 범위 내](#)를 참조하고 관심 있는 규정 준수 프로그램을 선택합니다. 일반 정보는 [AWS 규정 준수 프로그램](#).

를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다 AWS Artifact. 자세한 내용은 [Downloading Reports inDownloading AWS Artifact](#)을 참조하세요.

사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표, 관련 법률 및 규정에 따라 AWS 서비스 결정됩니다. 사용 시 규정 준수 책임에 대한 자세한 내용은 [AWS 보안 설명서를](#) AWS 서비스참조하세요.

의 복원력 AWS Elemental MediaTailor

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 중심으로 구축됩니다.는 물리적으로 분리되고 격리된 여러 가용 영역을 AWS 리전 제공하며,이 가용 영역은 짧은 지연 시간, 높은 처리량 및 높은 중복성을 갖춘 네트워킹과 연결됩니다. 가용 영역을 사용하면 중단 없이 영역 간에 자동으로 장애 극복 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 다중 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

AWS 리전 및 가용 영역에 대한 자세한 내용은 [AWS 글로벌 인프라를](#) 참조하세요.

AWS 글로벌 인프라 외에도 MediaTailor는 데이터 복원력 및 백업 요구 사항을 지원하는 몇 가지 기능을 제공합니다.

MediaTailor의 인프라 보안

관리형 서비스인 AWS 글로벌 네트워크 보안으로 보호 AWS Elemental MediaTailor 됩니다. AWS 보안 서비스 및가 인프라를 AWS 보호하는 방법에 대한 자세한 내용은 [AWS 클라우드 보안을](#) 참조하세요. 인프라 보안 모범 사례를 사용하여 환경을 설계하려면 보안 원칙 AWS Well-Architected Framework의 [인프라 보호를](#) 참조하세요 AWS .

AWS 게시된 API 호출을 사용하여 네트워크를 통해 MediaTailor에 액세스합니다. 클라이언트는 다음을 지원해야 합니다.

- Transport Layer Security(TLS). TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- DHE(Ephemeral Diffie-Hellman) 또는 ECDHE(Elliptic Curve Ephemeral Diffie-Hellman)와 같은 완전 전송 보안(PFS)이 포함된 암호 제품군. Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

모든 네트워크 위치에서 이러한 API 작업을 호출할 수 있지만 MediaTailor는 소스 IP 주소에 따른 제한을 포함할 수 있는 리소스 기반 액세스 정책을 지원합니다. MediaTailor 정책을 사용하여 특정 Amazon Virtual Private Cloud(VPC) 엔드포인트 또는 특정 VPCs. 이렇게 하면 네트워크 내의 특정 VPC에서만 지정된 MediaTailor 리소스에 대한 AWS 네트워크 액세스가 효과적으로 격리됩니다.

교차 서비스 혼동된 대리자 방지

혼동된 대리자 문제는 작업을 수행할 권한이 없는 엔터티가 권한이 더 많은 엔터티에게 작업을 수행하도록 강요할 수 있는 보안 문제입니다. 에서 AWS교차 서비스 가장은 혼동된 대리자 문제를 초래할 수 있습니다. 교차 서비스 가장은 한 서비스(직접 호출하는 서비스)가 다른 서비스(직접 호출되는 서비스)를 직접 호출할 때 발생할 수 있습니다. 직접 호출하는 서비스는 다른 고객의 리소스에 대해 액세스 권한이 없는 방식으로 작동하게 권한을 사용하도록 조작될 수 있습니다. 이를 방지하기 위해 AWS에서는 계정의 리소스에 대한 액세스 권한이 부여된 서비스 위탁자를 사용하여 모든 서비스에 대한 데이터를 보호하는 데 도움이 되는 도구를 제공합니다.

리소스 정책에서 [aws:SourceArn](#) 및 [aws:SourceAccount](#) 전역 조건 컨텍스트 키를 사용하여 리소스에 다른 서비스를 AWS Elemental MediaTailor 제공하는 권한을 제한하는 것이 좋습니다. 두 전역 조건 컨텍스트 키를 모두 사용하는 경우 `aws:SourceAccount` 값과 `aws:SourceArn` 값의 계정은 동일한 정책 문에서 사용할 경우 동일한 계정 ID를 사용해야 합니다.

의 값은 MediaTailor가 리전 및 계정에 CloudWatch 로그를 게시하는 재생 구성 `aws:SourceArn`이어야 합니다. 그러나 MediaTailor가 Amazon CloudWatch logs를 계정에 게시할 수 있는 [MediaTailorLogger](#) 역할을 사용하는 경우에만 적용됩니다. MediaTailor가 CloudWatch 로그를 게시하도록 [서비스 연결 역할](#)을 사용하는 경우에는 적용되지 않습니다.

혼동된 대리인 문제로부터 보호하는 가장 효과적인 방법은 리소스의 전체 ARN이 포함된 `aws:SourceArn`글로벌 조건 컨텍스트 키를 사용하는 것입니다. 리소스의 전체 ARN을 모를 경우 또는 여러 리소스를 지정하는 경우, ARN의 알 수 없는 부분에 대해 와일드카드(*)를 포함한 `aws:SourceArn`전역 조건 컨텍스트 키를 사용합니다. 예제: `arn:aws:servicename::123456789012:*`.

다음 예제에서는 MediaTailor에서 `aws:SourceArn` 및 `aws:SourceAccount` 전역 조건 컨텍스트 키를 사용하여 혼동된 대리자 문제를 방지하는 방법을 보여줍니다.

MediaTailor에서 로깅 및 모니터링

이 단원에는 보안을 위한 AWS Elemental MediaTailor의 로깅 및 모니터링 옵션에 대한 개요가 나와 있습니다. MediaTailor의 로깅 및 모니터링에 대한 자세한 내용은 섹션을 참조하세요 [AWS Elemental MediaTailor 리소스 모니터링 및 태그 지정](#).

모니터링은 AWS Elemental MediaTailor 및 AWS 솔루션의 안정성, 가용성 및 성능을 유지하는 데 중요한 부분입니다. 다중 지점 장애가 발생할 경우 보다 쉽게 디버깅할 수 있도록 AWS 솔루션의 모든 부분으로부터 모니터링 데이터를 수집해야 합니다. AWS는 MediaTailor 리소스를 모니터링하고 잠재적 인시던트에 대응하기 위한 여러 도구를 제공합니다.

Amazon CloudWatch 경보

CloudWatch 경보를 사용하면 지정한 기간 동안 단일 지표를 감시할 수 있습니다. 지표가 지정한 임계값을 초과하면 Amazon SNS 주제 또는 AWS Auto Scaling 정책으로 알림이 전송됩니다. CloudWatch 경보는 특정 상태에 있다고 해서 작업을 호출하지 않습니다. 대신, 상태가 변경되어 지정한 기간 동안 유지되어야 합니다. 자세한 내용은 [the section called “CloudWatch 지표를 사용한 모니터링”](#) 단원을 참조하십시오.

AWS CloudTrail 로그

CloudTrail은에서 사용자, 역할 또는 AWS 서비스가 수행한 작업에 대한 레코드를 제공합니다 AWS Elemental MediaTailor. CloudTrail에서 수집한 정보를 사용하여 MediaTailor에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다. 자세한 내용은 [AWS Elemental MediaTailor API 호출 기록](#) 단원을 참조하십시오.

AWS Trusted Advisor

Trusted Advisor는 수십만 명의 AWS 고객에게 서비스를 제공하여 학습한 모범 사례를 활용합니다.는 AWS 환경을 Trusted Advisor 검사한 다음 비용 절감, 시스템 가용성 및 성능 개선 또는 보안 격차 해소를 위한 기회가 존재할 때 권장 사항을 제시합니다. 모든 AWS 고객은 5개의 Trusted Advisor 점검 항목에 액세스할 수 있습니다. Business 또는 Enterprise Support 플랜을 보유한 고객은 모든 Trusted Advisor 점검 항목을 볼 수 있습니다.

자세한 내용은 [AWS Trusted Advisor](#) 단원을 참조하십시오.

AWS Elemental MediaTailor 리소스 모니터링 및 태그 지정

모니터링은 AWS Elemental MediaTailor 및 다른 AWS 솔루션의 안정성, 가용성 및 성능을 유지하는데 중요한 부분입니다. MediaTailor를 모니터링하고, 이상이 있을 때 이를 보고하고, 필요한 경우 자동 조치를 취할 수 있도록 다음과 같은 모니터링 도구를 AWS 제공합니다.

- Amazon CloudWatch는 AWS 리소스와 AWS 실행 중인 애플리케이션을 실시간으로 모니터링합니다. 지표를 수집 및 추적하고, 사용자 지정 대시보드를 생성할 수 있으며, 지정된 지표가 지정한 임계값에 도달하면 사용자에게 알리거나 조치를 취하도록 경보를 설정할 수 있습니다. 예를 들어 CloudWatch에서 Amazon EC2 인스턴스의 CPU 사용량 또는 기타 지표를 추적하고 필요할 때 자동으로 새 인스턴스를 시작할 수 있습니다. 자세한 내용은 [Amazon CloudWatch 사용자 안내서](#)를 참조하세요.
- Amazon CloudWatch Logs를 사용하면 광고 결정 서버(ADS)와의 모든 상호 작용에서 로그 파일을 모니터링, 저장 및 액세스할 수 있습니다. AWS Elemental MediaTailor 는 광고 요청, 리디렉션, 응답, 보고 요청 및 응답에 대한 로그를 내보냅니다. ADS 및 오리진 서버의 오류도 Amazon CloudWatch의 로그 그룹으로 내보내집니다. 또한 MediaTailor는 건너뛴 광고와 해당 광고를 건너뛴 이유에 대한 정보도 제공합니다. 자세한 내용은 [광고 건너뛰기 문제 해결](#) 단원을 참조하십시오. 또한 매우 내구성이 뛰어난 스토리지에 로그 데이터를 저장할 수 있습니다. 일반 정보는 [Amazon CloudWatch Logs 사용 설명서](#)를 참조하세요. ADS 로그 및 Amazon CloudWatch Logs Insights를 통한 분석을 위해 ADS 로그에 액세스하는 방법에 대한 자세한 내용은 섹션을 참조하세요 [AWS Elemental MediaTailor Amazon CloudWatch Logs Insights의 ADS 로그 분석](#).

주제

- [AWS Elemental MediaTailor 로그 보기](#)
- [Amazon CloudWatch 지표 AWS Elemental MediaTailor 를 사용한 모니터링](#)
- [AWS Elemental MediaTailor API 호출 기록](#)
- [AWS Elemental MediaTailor 채널 어셈블리 알림 수신](#)
- [AWS Elemental MediaTailor 리소스 태그 지정](#)
- [분석 대시보드를 사용하여 광고 삽입 성능 모니터링](#)
- [워크플로 모니터를 사용하여 AWS 미디어 서비스 모니터링](#)

AWS Elemental MediaTailor 로그 보기

MediaTailor는 채널 및 재생 구성의 다양한 마일스톤 및 활동을 설명하는 로그를 내보냅니다. 이러한 로그를 사용하여 워크플로를 파악하고 서비스 문제를 해결할 수 있습니다. 다음 주제에서는 로그 및 로깅 옵션에 대해 설명합니다.

주제

- [AWS Elemental MediaTailor ADS 로그 설명 및 이벤트 유형](#)
- [AWS Elemental MediaTailor 매니페스트 로그 설명 및 이벤트 유형](#)
- [AWS Elemental MediaTailor 트랜스코딩 로그 설명 및 이벤트 유형](#)
- [벤딩된 로그를 사용하여 AWS Elemental MediaTailor 로그 전송](#)
- [Amazon CloudWatch Logs에 직접 AWS Elemental MediaTailor 로그 작성](#)
- [AWS Elemental MediaTailor 로그 볼륨 제어](#)
- [AWS Elemental MediaTailor 로그 및 이벤트 필터링](#)
- [AWS Elemental MediaTailor 디버그 로그 생성](#)

AWS Elemental MediaTailor ADS 로그 설명 및 이벤트 유형

다음 섹션에서는 MediaTailor가 광고 결정 서버(ADS)의 이벤트를 설명하기 위해 내보내는 로그를 설명합니다. 로그입니다AdDecisionServerInteractions.

주제

- [AdDecisionServerInteractions 이벤트](#)
- [ADS 로그 설명](#)
- [ADS 로그 JSON 스키마](#)

AdDecisionServerInteractions 이벤트

광고 결정 서버(ADS)와의 MediaTailor 상호 작용 중에 다음 이벤트가 발생합니다.

Log	설명
AD_MARKER_FOUND	MediaTailor가 매니페스트에서 광고 마커를 찾았습니다.

Log	설명
BEACON_FIRED	추적 비컨이 실행되어 광고 이벤트를 보고했습니다. 서버 측 보고 모드(기본값)에서 MediaTailor는 비컨을 실행합니다. 클라이언트 측 보고 모드에서 재생 디바이스는 비컨을 실행합니다.
EMPTY_VAST_RESPONSE	ADS는 0개의 광고가 포함된 빈 VAST 응답을 반환했습니다.
EMPTY_VMAP_RESPONSE	ADS가 빈 VMAP 응답을 반환했습니다.
ERROR_ADS_INVALID_RESPONSE	ADS가 200이 아닌 상태 코드를 반환했습니다.
ERROR_ADS_IO	MediaTailor가 ADS와 통신을 시도하는 동안 오류가 발생했습니다.
ERROR_ADS_RESPONSE_PARSE	ADS 응답을 구문 분석하는 동안 MediaTailor에서 오류가 발생했습니다.
ERROR_ADS_RESPONSE_UNKNOWN_ROOT_ELEMENT	ADS 응답에 잘못된 루트 요소가 포함되어 있습니다.
ERROR_ADS_TIMEOUT	ADS에 대한 MediaTailor 요청 시간이 초과되었습니다.
ERROR_DISALLOWED_HOST	ADS 호스트는 허용되지 않습니다.
ERROR_FIRING_BEACON_FAILED	MediaTailor가 추적 비컨을 실행하는 데 실패했습니다.
ERROR_PERSONALIZATION_DISABLED	이 세션에서는 광고 삽입이 비활성화됩니다.
ERROR_UNKNOWN	ADS 요청 중에 MediaTailor에서 알 수 없는 오류가 발생했습니다.
ERROR_UNKNOWN_HOST	ADS 호스트를 알 수 없습니다.
ERROR_VAST_INVALID_MEDIA_FILE	VAST에 유효하지 않거나 누락된 MediaFile 요소가 Ad 있습니다.

Log	설명
ERROR_VAST_INVALID_VAST_AD_TAG_URI	VAST 응답에 잘못된 것이 포함되어 있습니다 VASTAdTagURI .
ERROR_VAST_MISSING_CREATIVES	VAST에는 0개 또는 여러 Creatives 요소가 Ad 포함되어 있습니다. 정확히 하나의 Creatives 요소가 필요합니다.
ERROR_VAST_MISSING_IMPRESSION	VAST에는 0개의 Impression 요소가 Ad 포함되어 있습니다. 하나 이상의 Impression 요소가 필요합니다.
ERROR_VAST_MISSING_MEDIAFILES	VAST에는 0개 또는 여러 MediaFiles 요소가 Ad 포함되어 있습니다. 정확히 하나의 MediaFiles 요소가 필요합니다.
ERROR_VAST_MISSING_OVERLAYS	MediaTailor가 광고 서버로부터 비선형 크리에이티브를 수신하지 않았습니다.
ERROR_VAST_MULTIPLE_LINEAR	VAST에는 여러 Linear 요소가 Ad 포함되어 있습니다.
ERROR_VAST_MULTIPLE_TRACKING_EVENTS	VAST에는 여러 TrackingEvents 요소가 Ad 포함되어 있습니다.
ERROR_VAST_REDIRECT_EMPTY_RESPONSE	VAST 리디렉션 요청이 빈 응답을 반환했습니다.
ERROR_VAST_REDIRECT_FAILED	VAST 리디렉션 요청에 오류가 발생했습니다.
ERROR_VAST_REDIRECT_MULTIPLE_VAST	VAST 리디렉션 요청이 여러 광고를 반환했습니다.
FILLED_AVAIL	MediaTailor가 사용 가능 구간을 성공적으로 채웠습니다.
FILLED_OVERLAY_AVAIL	MediaTailor가 오버레이 가능 구간을 성공적으로 채웠습니다.

Log	설명
INTERSTITIAL_VOD_FAILURE	ADS 요청 또는 응답에서 VOD 재생 목록에 대한 중간 간격을 채우는 동안 문제가 발생했습니다. 광고는 삽입되지 않습니다.
INTERSTITIAL_VOD_SUCCESS	MediaTailor가 VOD 재생 목록에 대한 중간 광고 시간을 성공적으로 채웠습니다.
MAKING_ADS_REQUEST	MediaTailor가 ADS에서 광고를 요청하고 있습니다.
MODIFIED_TARGET_URL	MediaTailor가 아웃바운드 대상 URL을 수정했습니다.
NON_AD_MARKER_FOUND	MediaTailor가 매니페스트에서 실행 불가능한 광고 마커를 찾았습니다.
POST_ADS_RESPONSE_FUNCTION_COMPLETED	사후 광고 응답 후크의 개별 함수가 완료되었습니다. 이는 옵트인 이벤트 유형입니다.
POST_ADS_RESPONSE_FUNCTION_ERROR	사후 광고 응답 후크의 개별 함수가 실패했습니다.
POST_ADS_RESPONSE_HOOK_ERROR	사후 광고 응답 후크 실행에 실패했습니다.
POST_ADS_RESPONSE_HOOK_SUMMARY	성공 또는 오류 상태를 포함한 사후 광고 응답 후크 실행 요약입니다. 이는 옵트인 이벤트 유형입니다.
PRE_ADS_REQUEST_FUNCTION_COMPLETED	사전 광고 요청 후크의 개별 함수가 완료되었습니다. 이는 옵트인 이벤트 유형입니다.
PRE_ADS_REQUEST_FUNCTION_ERROR	사전 광고 요청 후크의 개별 함수가 실패했습니다.
PRE_ADS_REQUEST_HOOK_ERROR	사전 광고 요청 후크 실행에 실패했습니다.

Log	설명
PRE_ADS_REQUEST_HOOK_SUMMARY	성공 또는 오류 상태를 포함한 사전 광고 요청 후크 실행 요약입니다. 이는 옵트인 이벤트 유형입니다.
PRE_MANIFEST_INSERTION_FUNCTION_COMPLETED	사전 매니페스트 삽입 후크의 개별 함수가 완료되었습니다. 이는 옵트인 이벤트 유형입니다.
PRE_MANIFEST_INSERTION_FUNCTION_ERROR	사전 매니페스트 삽입 후크의 개별 함수가 실패했습니다.
PRE_MANIFEST_INSERTION_HOOK_ERROR	사전 매니페스트 삽입 후크 실행이 실패했습니다.
PRE_MANIFEST_INSERTION_HOOK_SUMMARY	성공 또는 오류 상태를 포함한 사전 매니페스트 삽입 후크 실행 요약입니다. 이는 옵트인 이벤트 유형입니다.
RAW_ADS_RESPONSE	MediaTailor가 원시 ADS 응답을 받았습니다.
REDIRECTED_VAST_RESPONSE	MediaTailor는 VAST 리디렉션 후 VAST 응답을 받았습니다.
VAST_REDIRECT	VAST 광고 응답에는 리디렉션이 포함됩니다.
VAST_RESPONSE	MediaTailor가 VAST 응답을 받았습니다.
VOD_TIME_BASED_AVAIL_PLAN_SUCCESS	MediaTailor는 VOD 템플릿에 대한 시간 기반 가용 계획을 성공적으로 생성했습니다.
VOD_TIME_BASED_AVAIL_PLAN_VAST_RESPONSE_FOR_OFFSET	MediaTailor는 VOD 템플릿에 대한 시간 기반 가용 계획을 생성하고 있습니다. MediaTailor가 시간 오프셋에 대한 VAST 응답을 받았습니다.
VOD_TIME_BASED_AVAIL_PLAN_WARNING_NO_ADVERTISEMENTS	VOD 템플릿에 대한 시간 기반 가용 계획을 생성하는 동안 ADS 요청 또는 응답에 문제가 발생했습니다. 광고는 삽입되지 않습니다.

Log	설명
WARNING_NO_ADVERTISEMENTS	MediaTailor에서 가용 영역을 채우는 동안 문제가 발생했습니다. 광고가 삽입되지 않습니다.
WARNING_URL_VARIABLE_SUBSTITUTION_FAILED	MediaTailor는 ADS URL에서 동적 변수를 대체할 수 없습니다. URL 구성을 확인합니다.
WARNING_VPAID_AD_DROPPED	누락된 슬레이트로 인해 VPAID 광고가 삭제되었거나 세션에서 서버 측 보고를 사용합니다.

ADS 로그 설명

이 섹션에서는 ADS 로그 설명의 구조 및 콘텐츠에 관해 설명합니다. JSON 편집기에서 직접 탐색하려면 [the section called “ADS 로그 JSON 스키마”](#)에 있는 목록을 사용하십시오.

ADS 로그의 각 이벤트에는 CloudWatch Logs에서 생성되는 표준 필드가 포함되어 있습니다. 자세한 내용은 [CloudWatch Logs 인사이트를 사용하여 로그 데이터 분석](#)을 참조하세요.

ADS 로그 속성

이 섹션에서는 ADS 로그의 속성에 관해 설명합니다.

속성	유형	필수	설명
adsRequestUrl	문자열	false	MediaTailor가 수행하는 ADS 요청의 전체 URL
avail	avail 형식의 객체	false	MediaTailor가 광고로 채우는 가능 구간에 관한 정보. 현재 FILLED_AVAIL 이벤트 유형의 경우 이것은 MediaTailor가 가능 구간을 처음 만날 때 생성하는 계획입니다. 가능 구간이 결국 어떻

속성	유형	필수	설명
			게 채워지느냐는 이 계획과 콘텐츠가 전개되는 방식에 따라 달라집니다.
awsAccountId	문자열	true	세션에 사용된 MediaTailor 구성용 AWS 계정 ID
customerId	문자열	true	여러 로그 항목을 상호 연관시키는 데 사용할 수 있는 해시된 버전의 AWS 계정 ID
eventDescription	문자열	true	이 로그 메시지를 트리거한 이벤트에 대한 짧은 설명으로, MediaTailor 서비스에서 제공합니다. 기본적으로 이 문자열은 비어 있습니다. 예시: Got VAST response.
eventTimestamp	문자열	true	이벤트의 날짜 및 시간
eventType	문자열	true	이 로그 메시지를 트리거한 이벤트의 코드. 예시: VAST_RESPONSE .

속성	유형	필수	설명
errorType	문자열	false	구조화된 오류 분류입니다. PRE_ADS_REQUEST_HO OK_ERROR 및 PRE_ADS_REQUEST_FU NCTION_ERROR 이벤트에 표시됩니다. 값: INTERNAL_ERROR , SYNTAX_ERROR , RESOURCE_LIMIT_ERROR , RESTRICTION_ERROR , TIMEOUT_ERROR , VALIDATION_ERROR .
cause	문자열	false	특정 실패를 설명하는 오류 세부 정보 문자열입니다. PRE_ADS_REQUEST_HO OK_ERROR 및 PRE_ADS_REQUEST_FU NCTION_ERROR 이 이벤트에 표시됩니다.

속성	유형	필수	설명
eventId	문자열	false	후크 호출당 고유 식별자입니다. 이 필드를 사용하여 동일한 실행에 대한 후크 수준 이벤트와 함수 수준 이벤트를 상호 연관시킵니다. 함수 이벤트에만 표시됩니다.
executionTimeMs	정수	false	밀리초 단위의 실행 시간입니다. 함수 이벤트에만 표시됩니다.
functionId	문자열	false	함수의 식별자입니다. 후크 이벤트에서 이는 후크에 구성된 함수입니다. 함수 이벤트에서 실행기 내의 개별 함수입니다. 함수 이벤트에만 표시됩니다.
functionType	문자열	false	함수 유형: CUSTOM_OUTPUT , HTTP_REQUEST SEQUENTIAL_EXECUTOR , 또는 CONCURRENT_EXECUTOR . 함수 수준 이벤트에만 표시됩니다.

속성	유형	필수	설명
http	객체	false	HTTP 요청 및 응답 세부 정보입니다. 함수에 대한 HTTP_REQUEST 함수 수준 이벤트에만 표시됩니다. request (url, 메서드, 헤더, 본문), responseTimeMs , response (statusCode, 헤더, 본문) 및 truncated (로그 이벤트 크기 제한에 맞게 요청 또는 응답 본문이 잘린 경우 true)를 포함합니다.
input	객체	false	출력 표현식이 대상으로 하는 변수의 이전 값입니다. 함수 수준 이벤트에만 표시됩니다.
output	객체	false	함수에서 생성된 값입니다. PRE_ADS_REQUEST_FUNCTION_COMPLETED 이벤트에만 표시됩니다.
status	문자열	false	후크 실행 결과: SUCCESS 또는 ERROR. PRE_ADS_REQUEST_HOOK_SUMMARY 이벤트에만 표시됩니다.

속성	유형	필수	설명
originId	문자열	true	MediaTailor 구성의 구성 이름. 이것은 구성의 일부이기도 한 동영상 콘텐츠 원본과 다릅니다.
prefetchScheduleName	문자열	false	이 광고 이벤트와 연결된 미리 가져오기 일정의 이름입니다.
requestHeaders	requestheaders 형식의 배열	false	MediaTailor가 ADS 요청과 함께 포함한 헤더. 일반적으로 ADS에 대한 요청이 실패한 경우 문제 해결을 돕기 위해 로그에 이 헤더가 포함됩니다.
requestId	문자열	true	동일 요청에 대해 여러 로그 항목을 상호 연관시키는 데 사용할 수 있는 MediaTailor 요청 ID
sessionId	문자열	true	MediaTailor가 플레이어 세션에 지정한 고유한 숫자 식별자. 세션에 대해 플레이어가 하는 모든 요청은 세션 ID가 동일합니다. 예시: e039fd39-09f0-46b2-aca9-9871cc116cde .

속성	유형	필수	설명
sessionType	문자열(법정 값: [DASH, HLS])	true	플레이어의 스트림 유형
vastAd	vastAd 형식의 객체	false	VAST 응답에서 구문 분석된 단일 광고에 관한 정보
vastResponse	vastResponse 형식의 객체	false	MediaTailor가 ADS에서 받은 VAST 응답에 관한 정보
vodCreativeOffsets	vodCreativeOffsets 형식의 객체	false	VMAP 응답에 근거하여 MediaTailor가 가능 구간을 삽입할 매니페스트의 시간 오프셋을 나타내는 맵
vodVastResponseTimeOffset	number	false	VOD 광고 삽입을 위한 VMAP별 시간 오프셋

adContent

이 섹션에서는 ADS 로그 adContent의 속성에 대해 설명합니다.

ADS 로그 adContent의 속성

속성	유형	필수	설명
adPlaylistUri	adPlaylistUri 형식의 객체	false	변형의 원본 매니페스트에서 변형의 광고 매니페스트로 매핑. DASH의 경우 여기에는 단일 항목이 포함되어 있는데, 그 이유는 모든 변형이 단일

속성	유형	필수	설명
			DASH 매니페스트에 표시되기 때문입니다.

adPlaylistUri

이 섹션에서는 ADS 로그 adPlaylistUri의 속성에 관해 설명합니다.

ADS 로그 adPlaylistUri의 속성

속성	유형	필수	설명
<any string>	문자열	false	특정 변형에 대한 ad 매니페스트의 URL

avail

이 섹션에서는 ADS 로그 가능 구간의 속성에 관해 설명합니다.

ADS 로그 가능 구간의 속성

속성	유형	필수	설명
availId	문자열	true	이 가능 구간의 고유 식별자. HLS의 경우 이것은 가능 구간이 시작되는 미디어 시퀀스 번호입니다. DASH의 경우 이것은 기간 ID입니다.
creativeAds	creativeAd 형식의 배열	true	MediaTailor가 가능 구간에 삽입한 광고
fillRate	number	true	광고가 가능 구간 지속 시간을 채우는 속도로서 0.0(0%인 경우)에

속성	유형	필수	설명
			서 1.0(100%인 경우) 까지입니다.
filledDuration	number	true	가능 구간에 삽입된 모든 광고의 지속시간을 합한 것
numAds	number	true	MediaTailor가 가능 구간에 삽입한 광고의 수
originAvailableDuration	number	true	원본의 콘텐츠 스트림에 지정된 가능 구간의 지속시간(CUE_OUT 또는 SCTE)
skippedAds	skippedAd 형식의 배열	false	TRANSCODE_IN_PROGRESS 또는 TRANSCODE_ERROR 와 같은 이유로 MediaTailor가 삽입하지 않은 광고 건너뛴 광고 이유 목록은 섹션을 참조하세요 광고 건너뛰기 문제 해결 .
slateAd	slateAd 형식의 객체	true	슬레이트 광고에 관한 정보로서 MediaTailor 는 이 정보를 사용해 가능 구간 내에 채워지지 않은 세그먼트를 모두 채웁니다.

creativeAd

이 섹션에서는 ADS 로그 creativeAd의 속성에 대해 설명합니다.

ADS 로그 creativeAd의 속성

속성	유형	필수	설명
adContent	adContent 형식의 객체	true	삽입된 광고의 콘텐츠에 관한 정보
creativeUniqueId	문자열	true	트랜스코딩용 키로 사용되는 광고용 고유 식별자. 이것은 VAST 응답의 크리에이티브에 대한 ID 필드입니다(제공되는 경우). 또는 광고의 메자닌 URL입니다.
trackingEvents	trackingEvents 형식의 객체	true	광고의 다양한 추적 이벤트에 대한 추적 비콘 URL. 키는 이벤트 이름이고, 값은 비콘 URL의 목록입니다.
transcodeAdDuration	number	true	트랜스코딩된 자산에서 계산된 광고의 지속 시간
uri	문자열	true	메자닌 버전 광고의 URL로서, 트랜스코더에 입력되는 값
vastDuration	number	true	VAST 응답에서 구문 분석되는 광고의 지속 시간

requestheaders

이 섹션에서는 ADS 로그 requestheaders의 속성에 대해 설명합니다.

ADS 로그 requestheaders의 속성

속성	유형	필수	설명
name	문자열	true	헤더의 이름
value	문자열	true	헤더의 값

skippedAd

이 섹션에서는 ADS 로그 skippedAd의 속성에 대해 설명합니다.

ADS 로그 skippedAd의 속성

속성	유형	필수	설명
adMezzanineUrl	문자열	true	건너뛴 광고의 메자닌 URL
creativeUniqueId	문자열	true	트랜스코딩용 키로 사용되는 광고용 고유 식별자. 이것은 VAST 응답의 크리에이티브에 대한 ID 필드입니다(제공되는 경우). 또는 광고의 메자닌 URL입니다.
skippedReason	문자열	true	광고가 삽입되지 않은 이유를 나타내는 코드. 예시: TRANSCODE_IN_PROGRESS . 건너뛴 광고 이유 목록은 섹션을 참조하세요

속성	유형	필수	설명
			요광고 건너뛰기 문제 해결 .
transcode dAdDuration	number	false	트랜스코딩된 자산에서 계산된 광고의 지속 시간
vastDuration	number	true	VAST 응답에서 구문 분석되는 광고의 지속 시간

slateAd

이 섹션에서는 ADS 로그 slateAd의 속성에 대해 설명합니다.

ADS 로그 slateAd의 속성

속성	유형	필수	설명
adContent	adContent 형식의 객체	true	삽입된 광고의 콘텐츠에 관한 정보
creativeU niqueId	문자열	true	트랜스코딩용 키로 사용되는 광고용 고유 식별자. 이것은 VAST 응답의 크리에이티브에 대한 ID 필드입니다(제공되는 경우). 또는 광고의 메자닌 URL입니다.
transcode dAdDuration	number	true	트랜스코딩된 자산에서 계산된 광고의 지속 시간

속성	유형	필수	설명
uri	문자열	true	메자닌 버전 광고의 URL로서, 트랜스코더에 입력되는 값

trackingEvents

이 섹션에서는 ADS 로그 trackingEvents의 속성에 관해 설명합니다.

ADS 로그 trackingEvents의 속성

속성	유형	필수	설명
<any string>	문자열 형식의 배열	false	지정된 추적 이벤트의 비콘 URL 목록(노출, 완료 등)

vastAd

이 섹션에서는 ADS 로그 vastAd의 속성에 관해 설명합니다.

ADS 로그 vastAd의 속성

속성	유형	필수	설명
adSystem	문자열	true	VAST 응답에 있는 AdSystem 태그의 값
adTitle	문자열	true	VAST 응답에서 광고에 사용할 수 있는 미디어 파일
creativeAdId	문자열	true	VAST 응답에 있는 Creative 태그의 adId 속성에 대한 값

속성	유형	필수	설명
creativeId	문자열	true	VAST 응답에 있는 Creative 태그의 id 속성에 대한 값
duration	number	true	VAST 응답의 linear 요소에 있는 duration 태그에 근거하여 산출한 대략적인 광고 지속시간
trackingEvents	trackingEvents 형식의 객체	true	광고의 다양한 추적 이벤트에 대한 추적 비콘 URL. 키는 이벤트 이름이고, 값은 비콘 URL의 목록입니다.
vastAdId	문자열	true	VAST 응답에 있는 Ad 태그의 id 속성에 대한 값
vastAdTagUri	문자열	false	광고에 대한 VMAP별 리디렉션 URI
vastMediaFiles	vastMediaFile 형식의 배열	true	VAST 응답에서 광고에 사용할 수 있는 미디어 파일의 목록

vastMediaFile

이 섹션에서는 ADS 로그 vastMediaFile의 속성에 관해 설명합니다.

ADS 로그 vastMediaFile의 속성

속성	유형	필수	설명
apiFramework	문자열	true	미디어 파일 관리에 필요한 API 프레임워크 예시: VPAID.
bitrate	number	true	미디어 파일의 비트레이트
delivery	문자열	true	미디어 파일에 사용되는 프로토콜로서 프로그램메시브 또는 스트리밍으로 설정됨
height	number	true	미디어 파일의 픽셀 높이
id	문자열	true	MediaFile 태그의 id 속성에 대한 값
type	문자열	true	MediaFile 태그의 형식 속성에서 얻은 MIME 형식의 미디어 파일
uri	문자열	true	메자닌 버전 광고의 URL로서, 트랜스코더에 입력되는 값
width	number	true	미디어 파일의 픽셀 너비

vastResponse

이 섹션에서는 ADS 로그 vastResponse의 속성에 관해 설명합니다.

ADS 로그 vastResponse 속성

속성	유형	필수	설명
errors	문자열 형식의 배열	true	VAST 응답의 Error 태그에서 구문 분석된 오류 URL
vastAds	vastAd 형식의 배열	true	VAST 응답에서 구문 분석된 광고
version	문자열	true	응답에 있는 VAST 태그의 version 속성에서 구문 분석된 VAST 사양 버전

vodCreativeOffsets

이 섹션에서는 ADS 로그 vodCreativeOffsets의 속성에 관해 설명합니다.

ADS 로그 vodCreativeOffsets의 속성

속성	유형	필수	설명
<any string>	vodCreativeOffset 형식의 배열	false	매니페스트의 시간 오프셋에서 이 시점에 삽입할 광고 목록으로 매핑

vodCreativeOffset

이 섹션에서는 ADS 로그 vodCreativeOffset의 속성에 관해 설명합니다.

ADS 로그 vodCreativeOffset의 속성

속성	유형	필수	설명
adContent	adContent 형식의 객체	true	삽입된 광고의 콘텐츠에 관한 정보

속성	유형	필수	설명
creativeUniqueId	문자열	true	트랜스코딩용 키로 사용되는 광고용 고유 식별자. 이것은 VAST 응답의 크리에이티브에 대한 ID 필드입니다(제공되는 경우). 또는 광고의 메자닌 URL입니다.
trackingEvents	trackingEvents 형식의 객체	true	광고의 다양한 추적 이벤트에 대한 추적 비콘 URL. 키는 이벤트 이름이고, 값은 비콘 URL의 목록입니다.
transcodeAdDuration	number	true	트랜스코딩된 자산에서 계산된 광고의 지속 시간
uri	문자열	true	메자닌 버전 광고의 URL로서, 트랜스코더에 입력되는 값
vastDuration	number	true	VAST 응답에서 구문 분석되는 광고의 지속 시간

ADS 로그 JSON 스키마

다음은 AWS Elemental MediaTailor ADS 로그의 JSON 스키마 목록입니다.

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$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: `TRANSCODE_IN_PROGRESS`.

```

    },
    "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": {
    "$id": "#/properties/adContent",
    "type": ["object", "null"],
    "title": "adContent",
    "description": "Information about the content of the inserted ad.",
    "additionalProperties": false,
    "properties": {
      "adPlaylistUri": {
        "$id": "#/properties/adContent/adPlaylistUri",
        "type": "object",
        "title": "adPlaylistUri",
        "description": "The mapping from the origin manifest for a variant to the ad
manifest for the variant. For DASH, this contains a single entry, because all variants
are represented in a single DASH manifest. ",
        "additionalProperties": {
          "$id": "#/properties/adContent/adPlaylistUri/adPlaylistUri",
          "type": "string",
          "description": "The URL of the ad manifest for the specific variant."
        }
      }
    }
  },
},

```

```
"adMezzanineUri": {
  "type": "string",
  "description": "The URL of the mezzanine version of the ad, which is the input to
the transcoder."
},

"bitrate": {
  "type": "integer",
  "examples": [
    533
  ],
  "description": "The bitrate."
},

"width": {
  "type": "integer",
  "examples": [
    1280
  ],
  "description": "Width in pixels."
},

"height": {
  "type": "integer",
  "examples": [
    720
  ],
  "description": "Height in pixels."
},

"trackingEvents": {
  "type": "object",
  "title": "trackingEvents",
  "description": "The tracking beacon URLs for the various tracking events for the
ad. The keys are the event names, and the values are a list of beacon URLs.",

  "additionalProperties": {
    "type": "array",
    "description": "The list of beacon URLs for the specified tracking event
(impression, complete, and so on)",
    "items": {
      "type": "string",
      "description": "The beacon URLs for this tracking event."
    }
  }
},
```

```

"nonLinearAds": {
  "$id": "#/properties/nonLinearAds",
  "type": "object",
  "title": "nonLinearAds",
  "description": "A NonLinearAds as it appears in the VAST response.",
  "required": [
    "nonLinearAdList",
    "nonLinearTrackingEvents"
  ],
  "properties": {
    "nonLinearAdList": {
      "type": "array",
      "description": "List of non linear ads as they exist within one
NonLinearAds.",
      "items": {
        "type": "object",
        "title": "nonLinearAdList",
        "description": "List of NonLinearAd as they are parsed from its parent
NonLinearAds.",
        "properties": {
          "nonLinearAdId": {
            "type": "string",
            "description": "Ad ID of this non linear ad."
          },
          "nonLinearAdSystem": {
            "type": "string",
            "description": "Ad system of this non linear ad's parent Inline ad."
          },
          "nonLinearAdTitle": {
            "type": "string",
            "description": "Ad title of this non linear ad's parent Inline ad."
          },
          "nonLinearCreativeId": {
            "type": "string",
            "description": "Creative ID of this non linear ad's parent Creative
ad."
          },
          "nonLinearCreativeAdId": {
            "type": "string",
            "description": "Creative ad ID of this non linear ad."
          },
          "nonLinearCreativeSequence": {
            "type": "string",

```

```

    "description": "Creative sequence of this non linear ad."
  },
  "nonLinearWidth": {
    "type": "string",
    "description": "Width of this non linear ad."
  },
  "nonLinearHeight": {
    "type": "string",
    "description": "Height of this non linear ad."
  },
  "nonLinearExpandedWidth": {
    "type": "string",
    "description": "Expanded width of this non linear ad."
  },
  "nonLinearExpandedHeight": {
    "type": "string",
    "description": "Expanded height of this non linear ad."
  },
  "nonLinearScalable": {
    "type": "boolean",
    "description": "Boolean denoting if this non linear ad is scalable."
  },
  "nonLinearMaintainAspectRatio": {
    "type": "boolean",
    "description": "Boolean denoting if aspect ratio should be maintained
for this non linear ad."
  },
  "nonLinearMinSuggestedDuration": {
    "type": "string",
    "description": "Min suggested duration for this non linear ad."
  },
  "nonLinearApiFramework": {
    "type": "string",
    "description": "API framework for this non linear ad's parent Inline
ad."
  },
  "nonLinearStaticResource": {
    "type": "string",
    "description": "Static resource for this non linear ad."
  },
  "nonLinearStaticResourceCreativeType": {
    "type": "string",
    "description": "Static Resource creative type for this non linear ad."
  },

```

```

    "nonLinearIFrameResource": {
      "type": "string",
      "description": "I-Frame resource for this non linear ad."
    },
    "nonLinearHtmlResource": {
      "type": "string",
      "description": "HTML resource for this non linear ad."
    },
    "nonLinearAdParameters": {
      "type": "string",
      "description": "Ad parameters for this non linear ad."
    },
    "nonLinearClickThrough": {
      "type": "string",
      "description": "Click Through data for this non linear ad."
    },
    "nonLinearClickTracking": {
      "type": "string",
      "description": "Click Tracking data for this non linear ad."
    },
    "nonLinearClickTrackingId": {
      "type": "string",
      "description": "Click Tracking ID for this non linear ad."
    }
  }
},
"nonLinearTrackingEvents": { "$ref": "#/definitions/trackingEvents" },
"extensions": {
  "$id": "#/properties/nonLinearAds/extensions",
  "type": "array",
  "description": "Extensions that exist for this NonLinearAds.",
  "items": {
    "$id": "#/properties/nonLinearAds/extensions/items",
    "type": "object",
    "title": "Extensions",
    "description": "Extensions found in non linear ads",
    "additionalProperties": false,
    "properties": {
      "extensionType": {
        "$id": "#/properties/nonLinearAds/extensions/extensionType",
        "type": "string",
        "description": "The value of the extension type attribute of the
<code>Extensions</code> tag.",

```

```

        "examples": [
            "FreeWheel"
        ]
    },
    "extensionContent": {
        "$id": "#/properties/nonLinearAds/extensions/extensionContent",
        "type": "string",
        "description": "The extension content attribute of the
<code>Extensions</code> tag.",
        "examples": [
            "progressive"
        ]
    }
}
}
}
},
"adBreakTrackingEvents": {
    "$id": "#/properties/adBreakTrackingEvents",
    "type": "object",
    "title": "adBreakTrackingEvents",
    "description": "These are all VMAP ad break tracking events.",
    "additionalProperties": {
        "type": "array",
        "description": "VMAP/ad break tracking events and corresponding URL",
        "items": {
            "type": "string",
            "description": "The beacon URLs for this tracking event."
        }
    }
},
"vastAd": {
    "$id": "#/properties/vastAd",
    "type": "object",
    "title": "vastAd",
    "description": "Information about a single ad parsed from the VAST response.",
    "required": [
        "vastAdId",
        "adSystem",
        "adTitle",
        "creativeId",
        "creativeAdId",
        "duration",

```

```

    "vastMediaFiles",
    "trackingEvents"
  ],
  "additionalProperties": false,
  "properties": {
    "vastAdId": {
      "$id": "#/properties/vastAd/vastAdId",
      "type": "string",
      "description": "The value of the id attribute of the <code>Ad</code> tag in
the VAST response",
      "examples": [
        "ad1"
      ]
    },
    "adSystem": { "$ref": "#/definitions/adSystem" },
    "adTitle": {
      "$id": "#/properties/vastAd/adTitle",
      "type": "string",
      "description": "The media files that are available for the ad in the VAST
response.",
      "examples": [
        "External NCA1C1L1 LinearInlineSkippable"
      ]
    },
    "creativeId": {
      "$id": "#/properties/vastAd/creativeId",
      "type": "string",
      "description": "The value of the id attribute of the <code>Creative</code>
tag in the VAST response.",
      "examples": [
        "creative1"
      ]
    },
    "creativeAdId": {
      "$id": "#/properties/vastAd/creativeAdId",
      "type": "string",
      "description": "The value of the adId attribute of the <code>Creative</code>
tag in the VAST response."
    },
    "duration": {
      "$id": "#/properties/vastAd/duration",
      "type": "number",
      "description": "The approximate duration of the ad, based on the
<code>duration</code> tag in the <code>linear</code> element of the VAST response.",

```



```

        "examples": [
            30,
            30.0
        ]
    },
    "vastMediaFiles": {
        "$id": "#/properties/vastAd/vastMediaFiles",
        "type": "array",
        "description": "The list of available media files for the ad in the VAST
response.",
        "items": {
            "$id": "#/properties/vastAd/vastMediaFiles/items",
            "type": "object",
            "title": "vastMediaFile",
            "description": "Information about a media file for the ad.",
            "required": [
                "uri",
                "id",
                "delivery",
                "type",
                "apiFramework",
                "width",
                "height",
                "bitrate"
            ],
            "additionalProperties": false,
            "properties": {
                "uri": { "$ref": "#/definitions/adMezzanineUri" },
                "id": {
                    "$id": "#/properties/vastAd/vastMediaFiles/items/properties/id",
                    "type": "string",
                    "description": "The value of the id attribute of the <code>MediaFile</
code> tag.",
                    "examples": [
                        "GDFP"
                    ]
                },
                "delivery": {
                    "$id": "#/properties/vastAd/vastMediaFiles/items/properties/delivery",
                    "type": "string",
                    "description": "The protocol used for the media file, set to either
progressive or streaming.",
                    "examples": [
                        "progressive"
                    ]
                }
            }
        }
    }
}

```

```

    ],
    },
    "type": {
      "$id": "#/properties/vastAd/vastMediaFiles/items/properties/type",
      "type": "string",
      "description": "The MIME type of the media file, taken from the type
attribute of the <code>MediaFile</code> tag.",
      "examples": [
        "video/mp4"
      ]
    },
    },
    "apiFramework": {
      "$id": "#/properties/vastAd/vastMediaFiles/items/properties/
apiFramework",
      "type": "string",
      "description": "The API framework needed to manage the media file.
Example: <code>VPAID</code>."
    },
    },
    "width": {
      "$ref": "#/definitions/width"
    },
    },
    "height": {
      "$ref": "#/definitions/height"
    },
    },
    "bitrate": {
      "$ref": "#/definitions/bitrate"
    }
  }
}
},
"trackingEvents": { "$ref": "#/definitions/trackingEvents" },
"vastAdTagUri": {
  "$id": "#/properties/vastAd/vastAdTagUri",
  "type": "string",
  "description": "The VMAP-specific redirect URI for an ad.",
  "examples": [
    "https://ads.redirect.com/redirect1"
  ]
}
},
},
},
},
"eventPreAdsRequestHookSummary": {
  "$id": "#/definitions/eventPreAdsRequestHookSummary",

```

```
    "required": [
      "eventType",
      "eventId",
      "functionId",
      "executionTimeMs",
      "status"
    ],
    "properties": {
      "eventType": {
        "type": "string",
        "const": "PRE_ADS_REQUEST_HOOK_SUMMARY"
      }
    }
  },

  "eventPreAdsRequestHookError": {
    "$id": "#/definitions/eventPreAdsRequestHookError",
    "required": [
      "eventType",
      "eventId",
      "functionId",
      "executionTimeMs",
      "errorType",
      "cause"
    ],
    "properties": {
      "eventType": {
        "type": "string",
        "const": "PRE_ADS_REQUEST_HOOK_ERROR"
      }
    }
  },

  "eventPreAdsRequestFunctionCompleted": {
    "$id": "#/definitions/eventPreAdsRequestFunctionCompleted",
    "required": [
      "eventType",
      "eventId",
      "functionId",
      "functionType",
      "executionTimeMs"
    ],
    "properties": {
      "eventType": {
```

```

        "type": "string",
        "const": "PRE_ADS_REQUEST_FUNCTION_COMPLETED"
    }
}
},

"eventPreAdsRequestFunctionError": {
    "$id": "#/definitions/eventPreAdsRequestFunctionError",
    "required": [
        "eventType",
        "eventId",
        "functionId",
        "functionType",
        "executionTimeMs",
        "errorType",
        "cause"
    ],
    "properties": {
        "eventType": {
            "type": "string",
            "const": "PRE_ADS_REQUEST_FUNCTION_ERROR"
        }
    }
}
}
}
}
}

```

AWS Elemental MediaTailor 매니페스트 로그 설명 및 이벤트 유형

다음 섹션에서는 매니페스트를 요청하고 수신할 때 MediaTailor가 오리진 서버의 이벤트를 설명하기 위해 내보내는 로그를 설명합니다. 로그입니다ManifestService.

주제

- [ManifestService 이벤트](#)
- [매니페스트 로그 속성](#)

ManifestService 이벤트

MediaTailor와 오리진의 상호 작용 중에 다음 이벤트가 발생합니다.

Log	설명
CONFIG_SECURITY_ERROR	MediaTailor 구성에 보안 문제가 있습니다.
CONFIG_SYNTAX_ERROR	오리진 및 자산 경로로 인해 URL 형식이 잘못되었습니다.
CONNECTION_ERROR	오리진에 대한 MediaTailor 연결이 거부되거나 실패했습니다.
GENERATED_MANIFEST	MediaTailor에서 매니페스트를 생성했습니다. 이러한 로그를 수신하려면 디버그 모드가 활성화되어 있어야 합니다. 활성화 방법을 포함하여 디버그 로그 모드에 대한 자세한 내용은 섹션을 참조하세요 디버그 로그 생성 .
HOST_DISALLOWED	MediaTailor는 이 호스트에 대한 HTTP 요청을 허용하지 않습니다.
INCOMPATIBLE_HLS_VERSION	매니페스트는 호환되지 않는 HLS 버전을 사용합니다. MediaTailor에는 버전 3 이상이 필요합니다.
INVALID_SINGLE_PERIOD_DASH_MANIFEST	단일 기간 DASH 매니페스트가 잘못되었습니다. MediaTailor가 단일 기간 DASH 매니페스트를 통과하고 있습니다.
IO_ERROR	MediaTailor에서 오리진과 통신하는 동안 IO 오류가 발생했습니다.
LAST_PERIOD_MISSING_AUDIO	오리진 오디오 또는 비디오 정렬 오류로 AdaptationSets 인해 DASH 매니페스트의 마지막 기간에 모든 오디오가 누락되었습니다. 재생 문제를 방지하려면 최소한 다음 요청까지 마지막 기간 게시를 연기합니다.
LAST_PERIOD_MISSING_AUDIO_WARNING	오리진 오디오 또는 비디오 정렬 오류로 AdaptationSets 인해 DASH 매니페스트의

Log	설명
	마지막 기간에 모든 오디오가 누락되었습니다. 마지막 기간을 게시(지연하지 않음)하도록 선택합니다. 오디오가 누락되면 재생 문제가 발생할 수 있습니다.
MANIFEST_ERROR	MediaTailor 매니페스트 요청이 실패했습니다.
NO_MASTER_OR_MEDIA_PLAYLIST	오리진 응답에는 기본 재생 목록 또는 미디어 재생 목록이 포함되어 있지 않습니다.
NO_MASTER_PLAYLIST	오리진 응답에는 예상 기본 재생 목록이 포함되어 있지 않습니다.
NO_MEDIA_PLAYLIST	오리진 응답에 예상 미디어 재생 목록이 포함되어 있지 않습니다.
ORIGIN_MANIFEST	MediaTailor가 오리진 매니페스트를 가져왔습니다. 이러한 로그를 수신하려면 디버그 모드가 활성화되어 있어야 합니다. 활성화 방법을 포함하여 디버그 로그 모드에 대한 자세한 내용은 섹션을 참조하세요 디버그 로그 생성 .
PARSING_ERROR	오리진이 매니페스트 요청을 구문 분석할 수 없습니다.
PRE_SESSION_INIT_FUNCTION_COMPLETED	세션 전 초기화 후크의 개별 함수가 완료되었습니다. 이는 옵트인 이벤트 유형입니다.
PRE_SESSION_INIT_FUNCTION_ERROR	세션 전 초기화 후크의 개별 함수가 실패했습니다.
PRE_SESSION_INIT_HOOK_ERROR	세션 전 초기화 후크 실행이 실패했습니다.
PRE_SESSION_INIT_HOOK_SUMMARY	성공 또는 오류 상태를 포함한 세션 전 초기화 후크 실행 요약입니다. 이는 옵트인 이벤트 유형입니다.

Log	설명
SCTE35_PARSING_ERROR	MediaTailor가 매니페스트에서 Signal Binary 요소를 구문 분석할 수 없습니다.
SESSION_INITIALIZED	세션이 초기화되었습니다. 이러한 로그를 수신하려면 디버그 모드가 활성화되어 있어야 합니다. 활성화 방법을 포함하여 디버그 로그 모드에 대한 자세한 내용은 섹션을 참조하세요 디버그 로그 생성 .
TIMEOUT_ERROR	MediaTailor 매니페스트 요청 시간이 초과되었습니다.
TRACKING_RESPONSE	MediaTailor는 추적 응답을 제공했습니다. 이러한 로그를 수신하려면 디버그 모드가 활성화되어 있어야 합니다. 활성화 방법을 포함하여 디버그 로그 모드에 대한 자세한 내용은 섹션을 참조하세요 디버그 로그 생성 .
UNKNOWN_ERROR	MediaTailor에서 알 수 없는 오류가 발생했습니다.
UNKNOWN_HOST	호스트를 알 수 없습니다.
UNSUPPORTED_SINGLE_PERIOD_DASH_MANIFEST	단일 기간 DASH 매니페스트는 지원되지 않습니다. MediaTailor가 단일 기간 DASH 매니페스트를 통과하고 있습니다.

매니페스트 로그 속성

이 섹션에서는 매니페스트 로그의 속성을 설명합니다.

속성	유형	필수
awsAccountId	문자열	true

속성	유형	필수
eventTimestamp	문자열	true
originId	문자열	true
customerId	문자열	false
eventType	문자열	false
sessionId	문자열	false
originRequestUrl	문자열	false
mediaTailorPath	문자열	false
requestId	문자열	false
responseBody	문자열	false
sessionType	문자열(법정 값: [DASH, HLS])	false
requestNextToken	문자열	false
eventDescription	문자열	false
errorType	문자열	false
eventId	문자열	false
executionTimeMs	정수	false
functionId	문자열	false
functionType	문자열	false
http	객체	false
input	객체	false
output	객체	false

속성	유형	필수
status	문자열	false
assetPath	문자열	false
originFullUrl	문자열	false
originPrefixUrl	문자열	false
additionalInfo	문자열	false
cause	문자열	false
response	문자열	false
httpCode	문자열	false
errorMessage	문자열	false
adAdsResponse	문자열	false
adAdsRawResponse	문자열	false
adAdsRequest	문자열	false
adNumNewAvails	문자열	false
generatedMediaPlay list	문자열	false
requestStartTime	문자열	false
requestEndTime	문자열	false
requestStartTimeEp ochMillis	문자열	false
requestEndTimeEpoc hMillis	문자열	false

AWS Elemental MediaTailor 트랜스코딩 로그 설명 및 이벤트 유형

다음 섹션에서는 광고 스티칭을 위해 크리에이티브를 준비할 때 MediaTailor가 트랜스코딩 서비스의 이벤트를 설명하기 위해 보내는 로그를 설명합니다. 로그입니다TranscodeService.

주제

- [TranscodeService 이벤트](#)
- [트랜스코딩 로그 속성](#)

TranscodeService 이벤트

광고를 트랜스코딩하는 동안 MediaTailor 상호 작용 중에 다음 이벤트가 발생합니다.

Log	설명
IMPORT_ERROR	가져오기 작업 중에 MediaTailor에서 내부 오류가 발생했습니다(미리 조정된 광고의 경우). 빈 광고 세트 사용.
INITIALIZED	MediaTailor는 트랜스코딩 작업 또는 가져오기 작업(미리 조정된 광고의 경우)을 초기화했습니다.
INTERNAL_ERROR	MediaTailor에서 내부 오류가 발생했습니다. 빈 광고 세트 사용.
MISSING_VARIANTS	MediaTailor에서 변형이 누락되어 광고를 트랜스코딩할 수 없습니다. 빈 광고 세트 사용.
PROFILE_NOT_FOUND	트랜스코딩할 프로필이 누락되어 MediaTailor가 광고를 트랜스코딩할 수 없습니다. 빈 광고 세트 사용.
TRANSCODE_COMPLETED	비디오 트랜스코딩이 완료되었습니다. 광고는 광고 삽입에 사용할 수 있습니다.
TRANSCODE_ERROR	MediaTailor에서 트랜스코딩 작업 중에 내부 오류가 발생했습니다. 빈 광고 세트 사용.

Log	설명
TRANSCODE_IN_PROGRESS	비디오 트랜스코딩이 진행 중입니다. 트랜스코딩된 비디오가 준비되지 않았습니다. 빈 광고 세트 사용.

트랜스코딩 로그 속성

이 섹션에서는 트랜스코딩 로그의 속성을 설명합니다.

속성	유형	필수	설명
awsAccountId	문자열	true	세션에 사용된 MediaTailor 구성의 AWS 계정 ID입니다.
eventTimestamp	문자열	true	이벤트의 날짜 및 시간
originId	문자열	true	MediaTailor 구성의 구성 이름. 이것은 구성의 일부이기도 한 동영상 콘텐츠 원본과 다릅니다.
eventType	문자열	false	이 로그 메시지를 트리거한 이벤트의 코드. 예시: TRANSCODE_ERROR .
eventDescription	문자열	false	이 로그 메시지를 트리거한 이벤트에 대한 짧은 설명으로, MediaTailor 서비스에서 제공합니다. 기본적으로 이 문자열은 비어 있습니다.

속성	유형	필수	설명
sessionId	문자열	false	MediaTailor가 플레이어 세션에 지정한 고유한 숫자 식별자. 세션에 대해 플레이어가 하는 모든 요청은 세션 ID가 동일합니다. 예시: e039fd39-09f0-46b2-aca9-9871cc116cde .
creativeUniqueId	문자열	false	트랜스코딩되는 광고 크리에이티브의 고유 식별자입니다.
profileName	문자열	false	
adUri	문자열	false	광고 크리에이티브의 URI입니다.
transcodeRequestId	문자열	false	이 트랜스코딩 요청의 고유 식별자입니다.
cacheStatus	문자열	false	MediaTailor가 트랜스코딩된 광고를 캐싱했는지 여부를 나타냅니다.

벤딩된 로그를 사용하여 AWS Elemental MediaTailor 로그 전송

벤딩된 로그를 사용하면 유연성을 높이고 MediaTailor가 재생 구성에서 내보내는 로그를 전송할 위치를 제어할 수 있습니다.

MediaTailor는 벤딩된 로그를 사용하여 구성과 관련된 모든 로그 활동을 Amazon CloudWatch Logs로 전송합니다. 그런 다음 CloudWatch Logs는 지정한 로그의 백분율을 선택한 대상으로 전송합니다. 지

원되는 대상은 Amazon CloudWatch Logs 로그 그룹, Amazon S3 버킷 또는 Amazon Data Firehose 스트림입니다.

판매 로그는 볼륨 할인 요금으로 사용할 수 있으므로 로그를 CloudWatch Logs로 직접 전송하는 것과 비교하여 비용을 절감할 수 있습니다. 요금은 [Amazon CloudWatch 요금](#)의 로그 탭에서 판매 로그를 참조하세요.

벤딩 로그를 사용하려면 다음을 수행해야 합니다.

1. [권한 추가](#).
2. [로그 전송 대상 생성](#).
3. [CloudWatch Logs에서 로그 전송 구성](#).
4. [MediaTailor에서 판매 로그 활성화](#).

판매 로그에 대한 자세한 내용은 CloudWatch Logs 사용 설명서의 [AWS 서비스에서 로깅 활성화](#)를 참조하세요. MediaTailor는 벤딩된 로그의 V2를 지원합니다.

1단계: MediaTailor 로그 전송에 대한 권한 추가

판매 로그를 설정하는 사람에게는 전송 대상을 생성하고, 로그 전송을 구성하고, MediaTailor에서 판매 로그를 활성화할 수 있는 권한이 있어야 합니다. 다음 정책을 사용하여 판매 로그를 설정할 수 있는 적절한 권한이 있는지 확인합니다.

CloudWatch Logs 및 전송 대상에 대한 정책

Amazon CloudWatch Logs 사용 설명서의 다음 섹션에서는 CloudWatch Logs 및 전송 대상의 로그를 사용할 수 있는 정책을 제공합니다. 여러 위치로 로그를 전송하는 경우 여러 정책을 생성하는 대신 정책 설명을 하나의 정책으로 결합할 수 있습니다.

- [CloudWatch Logs로 전송된 로그](#)
- [Amazon S3로 보낸 로그](#)
- [Firehose로 전송된 로그](#)

콘솔에서 설정하기 위한 정책

API 또는 대신 콘솔을 통해 판매 로그 전송을 설정하는 경우 정책에 다음과 같은 추가 권한이 AWS CLI 있어야 합니다.

MediaTailor의 판매 로그에 대한 정책

MediaTailor에서 판매 로그 전송을 생성, 확인 또는 수정하려면 정책에 다음 권한이 있어야 합니다.

권한 추가 및 정책 작업에 대한 자세한 내용은 [섹션을 참조하세요](#) [에 대한 자격 증명 및 액세스 관리 AWS Elemental MediaTailor](#).

2단계: MediaTailor 로그의 전송 대상 생성

로그가 전송될 리소스를 생성합니다. 이후 단계에서 로그 전송을 구성하는 데 사용할 리소스의 ARN을 기록합니다.

CloudWatch Logs 로그 그룹 전송 대상

로그 그룹을 생성하는 데 도움이 되도록 다음 중 하나를 사용합니다.

- 콘솔의 경우 Amazon [CloudWatch Logs 사용 설명서의 CloudWatch Logs에서 로그 그룹 생성을 참조하세요](#). Amazon CloudWatch
- API의 경우 Amazon CloudWatch Logs API 참조의 [CreateLogGroup](#)을 참조하세요. Amazon CloudWatch
- SDKs 및 CLI의 경우 Amazon CloudWatch Logs 사용 설명서의 [AWS SDK와 CreateLogGroup 함께 사용을 AWS CLI](#) 참조하세요. Amazon CloudWatch

Amazon S3 버킷 전송 대상

버킷을 생성하는 데 도움이 되도록 다음 중 하나를 사용합니다.

- 콘솔, SDKs 및 CLI의 경우 Amazon Simple Storage Service 사용 설명서의 [버킷 생성을 참조하세요](#).
- API의 경우 Amazon Simple Storage Service API 참조의 [CreateBucket](#)을 참조하세요.

Firehose 스트림 전송 대상

스트림 생성에 대한 도움말은 Amazon Data [Firehose 개발자 안내서의 콘솔에서 Firehose 스트림 생성을 참조하세요](#).

3단계: MediaTailor 재생 구성에 대해 판매 로그 활성화

이전 단계에서 생성한 전송 대상으로 로그를 전송할 재생 구성을 생성하거나 업데이트합니다. 이후 단계에서 로그 전송을 구성하는 데 사용할 구성의 이름을 기록합니다.

- 콘솔을 통해 벤딩된 로그를 활성화하려면 [구성 생성](#) 또는 구성 [구성 설정 편집](#) 편집을 사용하여 로깅 설정에 액세스합니다. 로깅 전략에서 판매된 로그를 선택합니다.
- API를 통해 벤딩된 로그를 활성화하려면 기존 구성이 있어야 합니다. `ConfigureLogsForPlaybackConfiguration`를 사용하여 로깅 전략을 추가합니다 Vended logs.

CloudWatch Logs로 로그를 직접 전송하는 레거시 MediaTailor 로깅 전략을 사용 중이고 벤딩된 로그로 마이그레이션하려는 경우 섹션을 참조하세요 [로깅 전략 마이그레이션](#).

Important

로그 전략을 레거시 CloudWatch에서 판매 로그로 변경하면 업데이트를 저장하는 즉시 MediaTailor가 변경을 수행합니다. 판매 로깅을 완전히 구성할 때까지 로그 수신을 중지합니다.

4단계: CloudWatch Logs에서 로그 전송 구성

CloudWatch Logs에서는 로그 전송 부분을 나타내는 세 가지 요소를 생성해야 합니다. 이러한 요소는 Amazon CloudWatch Logs API 참조의 [CreateDelivery](#)에 자세히 설명되어 있습니다. CloudWatch Logs API를 사용하여 전송을 구성하는 상위 단계는 다음과 같습니다.

CloudWatch Logs(API)에서 로그 전송을 구성하려면

1. [PutDeliverySource](#)를 사용하여 로그 소스를 추가합니다.

는 로그를 생성하는 재생 구성을 DeliverySource 나타냅니다. 를 생성하려면 재생 구성의 이름이 필요합니다 DeliverySource.

2. [PutDeliveryDestination](#)를 사용하여 로그를 작성할 대상을 추가합니다.

는 전송 대상을 DeliveryDestination 나타냅니다. 를 생성하려면 로그 그룹, 버킷 또는 스트림의 ARN이 필요합니다 DeliveryDestination.

3. 계정 간에 로그를 전송하는 [PutDeliveryDestinationPolicy](#) 경우를 사용합니다.

전송 대상이 재생 구성과 다른 계정에 있는 경우이 필요합니다 DeliveryDestinationPolicy. 이 정책은 CloudWatch Logs가 로그를 전송하도록 허용합니다 DeliveryDestination.

4. [CreateDelivery](#)를 사용하여 DeliverySource에 연결합니다 DeliveryDestination.

는 DeliverySource와 간의 연결을 Delivery 나타냅니다 DeliveryDestination.

AWS Elemental MediaTailor 로깅 전략 마이그레이션

로그 전략을 레거시 CloudWatch에서 판매 로그로 변경하면 업데이트를 저장하는 즉시 MediaTailor가 변경을 수행합니다. 로깅 워크플로의 종단을 방지하려면 다음 단계를 사용하여 로깅 전략을 마이그레이션하세요.

1. 에 설명된 단계를 따릅니다 [벤딩된 로그 사용](#). 에서 두 로깅 전략(벤딩 로그 및 레거시 CloudWatch)을 모두 [MediaTailor에서 판매 로그 활성화](#) 선택합니다.

MediaTailor는 판매된 로그를 통해 CloudWatch Logs로 직접 로그를 전송합니다.

2. 로깅 전략 및 전송 대상에 따라 워크플로에서 필요한 변경을 수행합니다.
3. 다시 살펴보고 로깅 전략에서 레거시 CloudWatch를 [MediaTailor에서 판매 로그 활성화](#) 제거합니다.

Amazon CloudWatch Logs에 직접 AWS Elemental MediaTailor 로그 작성

MediaTailor는 세션 활동 및 광고 결정 서버 상호 작용에 대한 자세한 정보가 포함된 로그를 생성하여 Amazon CloudWatch에 기록합니다. 로그는 세션 중에 발생하는 활동에 대한 순차적 설명을 제공합니다.

또한 MediaTailor는 벤딩 로그를 사용하여 로그 전송 및 볼륨 할인 요금의 유연성을 높일 수 있습니다. 벤딩된 로그에 대한 자세한 내용은 섹션을 참조하세요 [벤딩된 로그 사용](#).

주제

- [Amazon CloudWatch Logs에 대한 권한](#)
- [AWS Elemental MediaTailor 채널 어셈블리에 대한 "실행 중" 로그](#)
- [AWS Elemental MediaTailor Amazon CloudWatch Logs Insights의 ADS 로그 분석](#)

Amazon CloudWatch Logs에 대한 권한

AWS Identity and Access Management (IAM)을 사용하여 Amazon CloudWatch에 대한 AWS Elemental MediaTailor 액세스 권한을 부여하는 역할을 생성합니다. 계정에 대해 CloudWatch Logs를 게시하려면 다음 단계를 수행해야 합니다. CloudWatch는 계정에 대한 지표를 자동으로 게시합니다.

CloudWatch에 대한 MediaTailor 액세스를 허용하려면

1. IAM 콘솔(<https://console.aws.amazon.com/iam/>)을 엽니다.
2. IAM 콘솔의 탐색 창에서 역할을 선택하고 역할 생성을 선택합니다.
3. 다른 AWS 계정 역할 유형을 선택합니다.
4. 계정 ID에 AWS 계정 ID를 입력합니다.
5. Require external ID(외부 ID 필요)를 선택하고 **Midas**를 입력합니다. 이 옵션은 요청에 올바른 sts:ExternalId가 포함된 경우에만 서비스가 역할을 맡을 수 있도록 허용하는 조건을 신뢰 정책에 자동으로 추가합니다.

6. 다음: 권한을 선택합니다.
7. 역할이 완료할 수 있는 작업을 지정하는 권한 정책을 추가합니다. 다음 옵션 중 하나를 선택한 다음 Next: Review(다음: 검토)를 선택합니다.
 - Amazon CloudWatch Logs에 대한 전체 액세스를 제공하는 CloudWatchLogsFullAccess Amazon CloudWatch
 - Amazon CloudWatch에 대한 전체 액세스를 제공하는 CloudWatchFullAccess Amazon CloudWatch
8. 역할 이름에 **MediaTailorLogger**를 입력한 다음 역할 생성을 선택합니다.
9. 역할 페이지에서 방금 생성한 역할을 선택합니다.
10. 보안 주체를 업데이트하도록 신뢰 관계를 편집합니다.
 1. 역할의 요약 페이지에서 신뢰 관계 탭을 선택합니다.
 2. 신뢰 관계 편집을 선택합니다.
 3. 정책 문서에서 보안 주체를 MediaTailor 서비스로 변경합니다. 형식은 다음과 같아야 합니다.

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

전체 결과는 다음과 같습니다.

4. 신뢰 정책 업데이트를 선택합니다.

AWS Elemental MediaTailor 채널 어셈블리에 대한 "실행 중" 로그

CloudWatch 로그 그룹의 실행 중 MediaTailor/Channel/AsRunLog 로그에는 재생되는 프로그램 및 광고 시간에 대한 정보가 표시됩니다.

채널을 생성하면 실행 중 로그가 기본적으로 비활성화됩니다. 콘솔 또는 AWS Command Line Interface (AWS CLI)를 사용하여 계정의 각 채널에 대해 실행 중 로그 상태를 활성화 및 비활성화할 수 있습니다.

실행 중 로그를 활성화하면 MediaTailor는 자동으로 서비스 연결 역할을 생성하여 MediaTailor가 CloudWatch Logs 계정에서 실행 중 로그를 작성하고 관리할 수 있도록 합니다. 서비스 연결 역할에 대한 자세한 내용은 [MediaTailor에 서비스 연결 역할 사용](#)를 참조하세요.

Note

실행 로그는 현재 기본 프로그램만 지원합니다. 지금은 프로그램 규칙에 의해 생성된 alternateMedia를 지원하지 않습니다. 즉, 현재 alternateMedia에 대한 As Run Log를 생성하지 않습니다.

주제

- [실행 중 로그 활성화](#)
- [실행 중 로그 비활성화](#)

실행 중 로그 활성화

실행 중 로그를 활성화하려면 채널 이름을 지정하고 해당 채널에 대해 실행 중 로그 유형을 활성화합니다.

Console

채널을 생성할 때 실행 중 로그를 활성화하려면

1. 에 로그인 AWS Management Console 하고 <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.
2. 탐색 창에서 채널 어셈블리 > 채널을 선택합니다.
3. 탐색 모음에서 채널 생성을 선택합니다.
4. 채널 세부 정보 설정, 출력 구성 및 액세스 제어 창에서 원하는 대로 채널을 구성합니다.
5. 액세스 제어 창에서 다음을 선택합니다.
6. 로깅 창의 로그 유형에서 실행 중 활성화를 선택하여 실행 중 로그를 활성화합니다.

채널을 업데이트할 때 실행 중 로그를 활성화하려면

Note

채널이 현재 실행 중인 경우 채널을 업데이트하려면 먼저 해당 채널을 중지해야 합니다. 채널을 중지한 후 작업 > 편집을 선택하여 채널 업데이트를 시작할 수 있습니다.

1. 에 로그인 AWS Management Console 하고 <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.
2. 탐색 창에서 채널 어셈블리 > 채널을 선택합니다.
3. 업데이트하려는 채널을 선택하여 실행 중 로그를 활성화합니다.
4. 작업(Actions) > 편집(Edit)을 선택합니다.
5. 채널 세부 정보 설정, 출력 구성 및 액세스 제어 창에서 채널 구성을 원하는 대로 업데이트합니다.
6. 액세스 제어 창에서 다음을 선택합니다.
7. 로깅 창의 로그 유형에서 실행 중 활성화를 선택하여 실행 중 로그를 활성화합니다.

로깅 탭에서 실행 중 로그를 활성화하려면

Note

채널이 현재 실행 중인 경우 작업 > 편집을 선택하여 실행 중 로그를 활성화하는 대신 로깅 탭을 사용해야 합니다.

1. 에 로그인 AWS Management Console 하고 <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.
2. 탐색 창에서 채널 어셈블리 > 채널을 선택합니다.
3. 실행 중 로그를 활성화할 채널을 선택합니다.
4. 탐색 모음의 채널 이름 아래에서 로깅을 선택합니다.
5. 로깅 > 로그 유형에서 실행 중을 선택하여 실행 중 로그를 활성화합니다.

AWS Command Line Interface (AWS CLI)

실행 중 로그를 활성화하려면

[configure-logs-for-channel](#) 명령을 실행하고 필요한 파라미터에 적절한 값을 지정합니다.

이 예제는 Linux, macOS 또는 Unix용으로 형식이 지정되며, 가독성을 높이기 위해 백슬래시(\) 줄 연속 문자를 사용합니다.

```
$ aws mediatailor configure-logs-for-channel \
  --channel-name MyChannel \
```

```
--log-types AS_RUN
```

이 예제는 Microsoft Windows용으로 포맷되었으며 가독성을 높이기 위해 캐럿(^) 줄 연속 문자를 사용합니다.

```
C:\> aws mediatailor configure-logs-for-channel ^  
--channel-name MyChannel ^  
--log-types AS_RUN
```

위치:

- *MyChannel*는 사용자가 소유하고 실행 중 로그를 활성화하려는 채널의 이름입니다.

이 명령이 성공적으로 실행되면 다음과 비슷한 출력이 표시됩니다.

```
{  
  "ChannelName": "MyChannel",  
  "LogTypes": [  
    "AS_RUN"  
  ]  
}
```

실행 중 로그 비활성화

채널이 활성화된 채널에 대해 실행 중 로그를 비활성화하려면 채널 이름을 지정하고 해당 채널에 대해 실행 중 로그 유형을 비활성화합니다.

Console

채널을 업데이트할 때 실행 중 로그를 비활성화하려면

Note

채널이 현재 실행 중인 경우 채널을 업데이트하려면 먼저 해당 채널을 중지해야 합니다. 채널을 중지한 후 작업 > 편집을 선택하여 채널 업데이트를 시작할 수 있습니다.

1. 에 로그인 AWS Management Console 하고 <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.

2. 탐색 창에서 채널 어셈블리 > 채널을 선택합니다.
3. 업데이트하려는 채널을 선택하여 실행 중 로그를 활성화합니다.
4. 작업(Actions) > 편집(Edit)을 선택합니다.
5. 채널 세부 정보 설정, 출력 구성 및 액세스 제어 창에서 채널 구성을 원하는 대로 업데이트합니다.
6. 액세스 제어 창에서 다음을 선택합니다.
7. 로깅 창의 로그 유형에서 실행 중 활성화를 선택 취소하여 실행 중 로그를 비활성화합니다.

로깅 탭에서 실행 중 로그를 비활성화하려면

Note

채널이 현재 실행 중인 경우 작업 > 편집을 선택하여 실행 중 로그를 비활성화하는 대신 로깅 탭을 사용해야 합니다.

1. 에 로그인 AWS Management Console 하고 <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.
2. 탐색 창에서 채널 어셈블리 > 채널을 선택합니다.
3. 실행 중 로그를 비활성화할 채널을 선택합니다.
4. 탐색 모음의 채널 이름 아래에서 로깅을 선택합니다.
5. 로깅 > 로그 유형에서 실행 중을 선택 취소하여 실행 중 로그를 비활성화합니다.

AWS Command Line Interface (AWS CLI)

실행 중 로그를 비활성화하려면

[configure-logs-for-channel](#) 명령을 실행하고 필요한 파라미터에 적절한 값을 지정합니다.

이 예제는 Linux, macOS 또는 Unix용으로 형식이 지정되며, 가독성을 높이기 위해 백슬래시(\) 줄 연속 문자를 사용합니다.

```
$ aws mediatailor configure-logs-for-channel \
  --channel-name MyChannel \
  --log-types
```

이 예제는 Microsoft Windows용으로 포맷되었으며 가독성을 높이기 위해 캐럿(^) 줄 연속 문자를 사용합니다.

```
C:\> aws mediatailor configure-logs-for-channel ^  
--channel-name MyChannel ^  
--log-types
```

위치:

- *MyChannel*는 사용자가 소유하고 실행 중 로그를 비활성화하려는 채널의 이름입니다.

이 명령이 성공적으로 실행되면 다음과 비슷한 출력이 표시됩니다.

```
{  
  "ChannelName": "MyChannel",  
  "LogTypes": []  
}
```

AWS Elemental MediaTailor Amazon CloudWatch Logs Insights의 ADS 로그 분석

Amazon CloudWatch Logs Insights를 사용하여 AWS Elemental MediaTailor 광고 결정 서버(ADS) 로그를 보고 쿼리할 수 있습니다. MediaTailor는 정상적인 처리 및 오류 조건을 위해 이벤트 로그를 CloudWatch로 전송합니다. 로그는 JSON 스키마를 준수합니다. CloudWatch Logs Insights를 통해 기간별로 로그를 선택한 다음 이에 대해 쿼리를 실행할 수 있습니다.

일반적인 내용은 [CloudWatch Logs 인사이트를 사용하여 로그 데이터 분석을](#) 참조하세요.

Note

로그에 액세스하려면 Amazon CloudWatch에 액세스할 수 있는 권한이 필요합니다. 지침은 [Amazon CloudWatch Logs에 대한 권한](#) 섹션을 참조하세요.

CloudWatch 콘솔을 사용하여 ADS 로그를 보고 쿼리하려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창의 로그에서 Insights를 선택합니다.

3. 검색 창에 입력한 **AdDec** 다음 드롭다운 목록에서 **MediaTailor/AdDecisionServerInteractions**.
4. (선택 사항) 보고 싶은 기간을 조정합니다.
5. (선택 사항) 대화 상자의 쿼리를 변경합니다. 일반 지침은 [CloudWatch Logs 인사이트 쿼리 구문](#)을 참조하세요. MediaTailor ADS에 대한 쿼리 예제는 섹션을 참조하세요 [ADS 로그 쿼리](#).
6. 쿼리 실행을 선택합니다. 쿼리는 몇 초 정도 걸릴 수 있으며, 이 시간 동안 쿼리 실행 대신 취소가 나타납니다.
7. (선택 사항) 결과를 CSV 파일 형식으로 내보내려면 작업을 선택한 후 **Download query results (CSV)**(쿼리 결과 다운로드(CSV))를 선택합니다.

Note

콘솔은 쿼리 결과에 반환되고 내보내는 레코드 수를 제한하므로 대량 데이터의 경우 API, AWS Command Line Interface (AWS CLI) 또는 SDK를 사용합니다.

주제

- [ADS 로그 쿼리](#)

ADS 로그 쿼리

CloudWatch Logs Insights는 로그 쿼리를 위한 다양한 옵션을 제공합니다. 쿼리 구문에 대한 자세한 내용은 [CloudWatch Logs 인사이트 쿼리 구문](#)을 참조하세요. 이 섹션에서는 ADS 로그 쿼리를 시작할 수 있도록 일반적인 쿼리의 예를 제공합니다. 모든 쿼리는 현재 시간 범위 설정의 로그에 대해 실행됩니다.

다음 쿼리에서는 ADS 로그의 모든 정보를 검색합니다.

```
fields @timestamp, eventType, sessionId, requestId, @message
| sort sessionId, @timestamp asc
```

다음 쿼리에서는 ADS에 대한 모든 요청을 검색합니다. 이 쿼리는 MediaTailor 로그에 대한 요청 헤더 콘텐츠를 검색하는 방법을 보여줍니다.

```
fields @timestamp, adsRequestUrl, requestHeaders.0.value as @userAgent,
requestHeaders.1.value as @xForwardedFor, sessionId, requestId
| filter eventType = "MAKING_ADS_REQUEST"
```

```
| sort @timestamp asc
```

다음 쿼리는 지정된 세션에 대해 삽입된 MediaTailor 광고를 검색합니다.

```
fields @timestamp, sessionId, requestId, @message
| filter eventType = "FILLED_AVAIL"
| sort @timestamp asc
```

다음 쿼리는 MediaTailor가 플레이어 대신하여 호출한 추적 URLs을 검색합니다.

```
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
```

다음 쿼리에서는 sessionId를 기준으로 결과를 필터링하여 특정 재생 세션에 대한 정보를 검색합니다.

```
fields @timestamp, eventType, sessionId, requestId, @message
| filter sessionId = "0aaf6507-c6f9-4884-bfe7-f2f841cb8195"
| sort @timestamp asc
```

다음 쿼리에서는 requestId를 기준으로 결과를 필터링하여 단일 요청에 대한 정보를 검색합니다.

```
fields @timestamp, eventType, sessionId, requestId, @message
| filter requestId = "f5d3cf39-6258-4cf1-b3f6-a34ff8bf641d"
| sort @timestamp asc
```

다음 쿼리에서는 기록된 각 이벤트 유형에 대해 로그 항목의 개수를 검색합니다.

```
fields eventType
| stats count() as @eventCount by eventType
```

다음 쿼리에서는 건너뛴 광고가 있었던 모든 가능 구간에 대해 가능 구간 ID와 건너뛴 광고의 목록을 검색합니다.

```
fields avail.availId
| parse @message '"skippedAds":[*]' as @skippedAdsList
| filter ispresent(@skippedAdsList)
```


AWS Elemental MediaTailor 로그 볼륨 제어

MediaTailor 광고 삽입 세션 로그는 경우에 따라 상세하게 표시됩니다. 로그 비용을 줄이기 위해 MediaTailor가 Amazon CloudWatch Logs로 보내는 세션 로그의 비율을 정의할 수 있습니다. 예를 들어 재생 구성에 1,000개의 광고 삽입 세션이 있고 백분율 활성화 값을 로 설정하면 60 MediaTailor는 600개의 세션에 대한 로그를 CloudWatch Logs로 전송합니다. MediaTailor는 로그를 전송할 세션을 무작위로 결정합니다. 특정 세션에 대한 로그를 보려면 [디버그 로그 모드](#)를 사용할 수 있습니다.

로깅 백분율을 설정하면 MediaTailor는 계정에 CloudWatch Logs를 작성하는 데 필요한 권한을 MediaTailor에 부여하는 서비스 연결 역할을 자동으로 생성합니다. MediaTailor가 서비스 연결 역할을 사용하는 방법에 대한 자세한 내용은 섹션을 참조하세요 [MediaTailor에 서비스 연결 역할 사용](#).

로그 구성 생성

MediaTailor가 CloudWatch Logs에 쓰는 세션 로그의 비율을 제어하려면 재생 구성에 대한 로그 구성을 생성합니다. 로그 구성을 생성할 때 재생 구성 이름과 활성화된 백분율 값을 지정합니다.

Console

기존 재생 구성에 대한 로그 구성을 생성하려면

1. 에 로그인 AWS Management Console 하고 <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.
2. 재생 구성 창에서 로그 구성을 설정할 재생 구성을 선택합니다.
3. 편집을 선택합니다.
4. 로그 구성에서 활성화된 백분율 값을 지정합니다.

새 재생 구성에 대한 로그 구성을 생성하려면

- [로그 구성](#)의 프로시저를 따르세요.

AWS Command Line Interface (AWS CLI)

기존 재생 구성에 대한 로그 구성을 생성하려면

를 사용하여 로그 구성을 생성하려면 [configure-logs-for-playback-configuration](#) 명령을 AWS CLI 실행하고 필요한 파라미터에 적절한 값을 지정합니다.

이 예제는 Linux, macOS 또는 Unix용으로 형식이 지정되며, 가독성을 높이기 위해 백슬래시(\) 줄 연속 문자를 사용합니다.

```
$ aws mediatailor configure-logs-for-playback-configuration \
--percent-enabled 10 \
--playback-configuration-name MyPlaybackConfiguration
```

이 예제는 Microsoft Windows용으로 포맷되었으며 가독성을 높이기 위해 캐럿(^) 줄 연속 문자를 사용합니다.

```
C:\> aws mediatailor configure-logs-for-playback-configuration ^
--percent-enabled 10 ^
--playback-configuration-name MyPlaybackConfiguration
```

위치:

- *percent-enabled*는 MediaTailor가 CloudWatch Logs로 보내는 재생 구성 세션 로그의 백분율입니다.
- *playback-configuration-name*는 로그 구성 설정을 위한 재생 구성의 이름입니다.

이 명령이 성공적으로 실행되면 다음과 비슷한 출력이 표시됩니다.

```
{
  "PercentEnabled": 10,
  "PlaybackConfigurationName": "MyPlaybackConfiguration"
}
```

새 재생 구성에 대한 로그 구성을 생성하려면

- [put-playback-configuration](#) 명령에 `configure-logs-for-playback-configuration` 옵션을 사용합니다.

로그 구성 비활성화

로그 구성을 생성한 후에는 삭제할 수 없으며 비활성화만 할 수 있습니다. 로그 구성을 비활성화하려면 MediaTailor 콘솔 또는 API를 사용하여 활성화된 백분율 값을 0으로 설정합니다. 그러면 해당 재생 구성에 대한 모든 세션 로깅이 꺼집니다.

MediaTailor가 계정의 로그 구성에 사용하는 서비스 연결 역할을 삭제하려면 먼저 모든 로그 구성을 비활성화해야 합니다. 서비스 연결 역할을 삭제하는 방법에 대한 자세한 내용은 섹션을 참조하세요 [MediaTailor에 서비스 연결 역할 사용](#).

Console

재생 구성에서 로그 구성을 비활성화하려면

1. 에 로그인 AWS Management Console 하고 <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.
2. 재생 구성 창에서 로그 구성을 비활성화할 재생 구성을 선택합니다.
3. 편집을 선택합니다.
4. 로그 구성에서 활성화된 백분율 값을 로 설정합니다. 이렇게 하면이 재생 구성에 대한 모든 세션 로깅이 꺼집니다.
5. 저장을 선택합니다.

AWS Command Line Interface (AWS CLI)

로그 구성을 비활성화하려면

- [configure-logs-for-playback-configuration](#) 명령을 0 사용하여 percent-enabled 값을 로 설정합니다.

AWS Elemental MediaTailor 로그 및 이벤트 필터링

MediaTailor의 재생 구성에서 내보낸 로그에는 재생 세션 중에 발생하는 다양한 활동에 대한 정보가 포함됩니다. 이러한 활동은 로그의 이벤트 유형에서 식별됩니다. 기본적으로 많은 이벤트가 기록됩니다. Amazon CloudWatch에서 로그 비용을 제어하는 데 도움이 되도록 MediaTailor가 내보내는 로그를 지정할 수 있습니다.

MediaTailor는 로그 필터링을 제어하므로 다음을 수행할 수 있습니다.

- 로그에서 제외할 로그 이벤트 지정
- 광고 결정 서버(ADS)에서 원시 응답 로깅 활성화

이러한 로그 필터링 기본 설정을 각 재생 세션에 대해 독립적으로 설정하거나 재생 구성에 대한 모든 재생 세션의 기본값으로 설정할 수 있습니다.

- 세션별로 로그를 필터링하려면 재생 세션 초기화 요청에 쿼리 파라미터를 포함합니다.
- 재생 구성별로 로그를 필터링하려면 MediaTailor 콘솔 또는 API를 사용하여 재생 구성 설정에서 기본 설정을 지정합니다.

다음 섹션에서는 세션 및 재생 구성에서 로그 필터링을 활성화하기 위한 지침을 제공합니다.

세션별 로그 필터

각 세션에 대한 사용자 지정 로그 세부 정보 수준을 정의하려면 초기 서버 측 또는 클라이언트 측 재생 세션 요청에 다음 파라미터를 추가합니다. 파라미터에 값을 추가하여 포함하거나 제외하려는 이벤트를 심볼로 구분된 형식으로 나타냅니다.

- `aws.adsInteractionLogPublishOptInEventTypes` 특정 광고 결정 서버(ADS) 상호 작용에 대한 로그를 수신합니다.
- `aws.adsInteractionLogExcludeEventTypes` 특정 ADS 상호 작용에 대한 로그 수신을 중지합니다.
- `aws.manifestServiceLogExcludeEventTypes` 특정 매니페스트 서비스 상호 작용에 대한 로그 수신을 중지합니다.

MediaTailor에서 내보내는 로그 및 이벤트 유형 목록은 [매니페스트 로그](#), [ADS 로그](#) 및 섹션을 참조하세요 [로그 트랜스코딩](#).

로그 필터링을 위한 쿼리 파라미터를 전달하지 않으면 MediaTailor는 모든 로그를 전송 대상에 기록합니다.

Example로그 필터를 사용한 서버 측 세션 초기화

매니페스트 로그 및 ADS 로그MAKING_ADS_REQUEST에서 GENERATED_MANIFEST 및 PARSING_ERROR 이벤트를 제외하려면 세션 초기화 요청은 다음과 같습니다.

```
GET <mediatailorURL>/v1/master/<hashed-account-id>/<origin-id>/index.m3u8?
aws.logMode=DEBUG&aws.manifestServiceLogExcludeEventTypes=GENERATED_MANIFEST,PARSING_ERROR&aws.
```

ADS에서 원시 로그를 활성화하려면 `AdsInteractionPublishOptInEventType` 파라미터 `RAW_ADS_RESPONSE` 값을 포함합니다.

```
GET <mediatailorURL>/v1/master/<hashed-account-id>/<origin-id>/index.m3u8?
aws.adsInteractionPublishOptInEventType=RAW_ADS_RESPONSE
```

Example로그 필터를 사용한 클라이언트 측 세션 초기화

클라이언트 측 세션 초기화 중에 로그 이벤트를 제외하려면 MediaTailor에 대한 클라이언트의 POST 요청에 `availSuppression` 및 로그 유형 파라미터를 포함합니다. 클라이언트 측 재생 세션 요청을 구성하는 방법에 대한 자세한 내용은 [클라이언트 측 광고 추적](#) 단원을 참조하십시오. 다음 예제에서는 매니페스트 로그 및 ADS 로그MAKING_ADS_REQUEST에서 CONFIG_SECURITY_ERROR 및 PARSING_ERROR 이벤트를 제외합니다.

```
POST parent.m3u8
{
  "adsInteractionLog": {
    ...
    "excludeEventTypes": [
      "MAKING_ADS_REQUEST"
    ]
  },
  "manifestServiceLog": {
    ...
    "excludeEventTypes": [
      "GENERATED_MANIFEST",
      "PARSING_ERROR"
    ]
  },
  "logMode": "DEBUG"
}
```

ADS에서 원시 로그를 활성화하려면 `publishOptInEventTypes` 파라미터 `RAW_ADS_RESPONSE` 값을 포함합니다.

```
POST parent.m3u8
{
  "adsInteractionLog": {
    "publishOptInEventTypes": ["RAW_ADS_RESPONSE"],
    "excludeEventTypes": [
      "MAKING_ADS_REQUEST"
    ]
  },
  "manifestServiceLog": {
    ...
    "excludeEventTypes": [
      "GENERATED_MANIFEST",
      "PARSING_ERROR"
    ]
  }
}
```

```

    ]
  },
  "logMode": "DEBUG"
}

```

재생별 구성 로그 필터

재생 구성의 설정을 사용하여 MediaTailor가 재생 구성에서 기본값으로 내보내는 로그 이벤트 유형을 정의합니다. MediaTailor는 세션 초기화 요청에 쿼리 파라미터 필터링을 포함하지 않는 모든 세션에 대해 이러한 기본 로그 필터링 설정을 사용합니다.

다음을 수행하도록 선택할 수 있습니다.

- 특정 광고 결정 서버(ADS) 상호 작용에 대한 로그를 수신합니다.
- 특정 ADS 상호 작용에 대한 로그를 제외합니다.
- 특정 매니페스트 서비스 상호 작용에 대한 로그를 제외합니다.

MediaTailor 콘솔에서 이러한 설정을 지정하려면 섹션을 참조하세요 [구성 생성](#). MediaTailor API의 경우 API 참조 [PutPlaybackConfiguration](#)의 AWS Elemental MediaTailor 섹션을 참조하세요.

MediaTailor에서 내보내는 로그 및 이벤트 유형 목록은 [매니페스트 로그](#), [ADS 로그](#) 및 섹션을 참조하세요 [로그 트랜스코딩](#).

AWS Elemental MediaTailor 디버그 로그 생성

디버그 로그를 사용하여 MediaTailor 광고 삽입 재생 세션 문제를 해결합니다. 디버그 로그를 생성하려면 플레이어의 MediaTailor 요청에서 디버그하도록 로그 모드를 설정합니다. 서버 측 보고의 경우 재생 요청에서 로그 모드를 설정합니다. 클라이언트 측 보고의 경우 세션 초기화 요청에서 로그 모드를 설정합니다.

로그 모드가 디버그로 설정된 경우 MediaTailor는 모든 로그 이벤트 유형을 CloudWatch Logs에 기록합니다. 로그는 다음 이벤트에 대한 정보를 제공합니다. 디버그 로그에서 생성된 데이터의 전체 목록은 [디버그 로그 필드](#)를 참조하세요.

- 오리진 상호 작용 - 오리진 서버와의 MediaTailor 상호 작용에 대한 세부 정보입니다. 오리진 매니페스트 응답, 매니페스트 유형 및 오리진 URL을 예로 들 수 있습니다.
- 생성된 매니페스트 - MediaTailor의 재생 세션 응답에 대한 세부 정보입니다. 예를 들어 MediaTailor가 생성하는 매니페스트입니다.

- 세션 초기화됨 - 세션 ID와 같은 세션 초기화 세부 정보입니다.

세션별로 수신하는 로그 이벤트 유형을 사용자 지정하려면 섹션을 참조하세요 [로그 및 이벤트 필터링](#).

사전 조건

로그 모드를 디버깅으로 설정하려면 먼저 아직 로그를 CloudWatch로 전송할 수 있는 권한을 MediaTailor에 부여해야 합니다. MediaTailor가 CloudWatch에 액세스할 수 있는 권한을 부여하면 디버그 로그 모드를 활성화할 준비가 된 것입니다. CloudWatch에 액세스할 수 있는 권한을 MediaTailor에 부여하는 방법에 대한 자세한 내용은 [Amazon CloudWatch에 대한 권한 설정을 참조하세요](#).

디버깅하도록 로그 모드를 설정하는 방법

이 섹션에서는 서버 측 보고 및 클라이언트 측 보고를 위해 디버깅하도록 로그 모드를 설정하는 방법을 설명합니다.

서버 측 보고

서버 측 보고의 경우 HLS 또는 DASH MediaTailor 엔드포인트에 대한 플레이어의 GET HTTP 재생 요청에 `?aws.logMode=DEBUG` 쿼리 파라미터와 값을 포함합니다. 서버 측 보고에 대한 일반적인 내용은 [서버 측 보고를 참조하세요](#).

Important

DEBUG 값은 대소문자를 구분합니다.

가 포함된 재생 요청은 다음과 `?aws.logMode=DEBUG` 같습니다.

Example HLS 엔드포인트에 대한 재생 요청

```
GET <mediatailorURL>/v1/master/<hashed-account-id>/<origin-id>/<asset-id>?
aws.logMode=DEBUG
```

로그 모드를 디버깅으로 설정한 후에는 디버깅 로깅 세션이 활성화 상태인지 확인하는 것이 좋습니다. 디버깅 세션이 활성화 상태인지 확인하려면 세션 ID에 대한 CloudWatch 로그가 있는지 확인합니다. 세션 ID는 MediaTailor가 제공하는 재생 엔드포인트에 포함됩니다. 자세한 내용은 [Verify that the debug log mode is active for your playback session](#) 단원을 참조하십시오.

클라이언트 측 보고

클라이언트 측 보고의 경우 MediaTailor /v1/session 엔드포인트에 대한 클라이언트의 POST HTTP 세션 초기화 요청 본문에 logMode 키와 DEBUG 값을 포함합니다. 클라이언트 측 보고에 대한 일반적인 내용은 [클라이언트 측 보고를 참조하세요](#).

Important

DEBUG 값은 대소문자를 구분합니다.

로그 모드를 디버그로 설정한 후에는 디버그 세션이 활성 상태인지 확인하는 것이 좋습니다. 디버그 세션이 활성 상태인지 확인하려면 CloudWatch 로그에 세션 ID와 연결된 SESSION_INITIALIZED 이벤트가 있는지 확인합니다. 세션 ID는 MediaTailor가 제공하는 재생 엔드포인트에 포함됩니다. 자세한 내용은 [Verify that the debug log mode is active for your playback session](#) 단원을 참조하십시오.

최대 활성 디버그 세션

최대 10개의 활성 디버그 로그 세션을 가질 수 있습니다. 플레이어는 세션 초기화 또는 재생 요청을 MediaTailor로 보내면 MediaTailor는 한도에 도달했는지 확인합니다. 있는 경우 MediaTailor는 오래된 세션이 있는지 확인합니다. 일정 기간 내에 액세스하지 않은 세션은 기한이 경과합니다. 라이브 스트림의 경우 이 기간은 10분이고 VOD 스트림의 경우 30분입니다.

최대 활성 디버그 로그 세션 한도에 도달하면 세션의 CloudWatch Logs에 디버그 로그가 기록되지 않습니다. 세션에 대한 CloudWatch Logs에 디버그 로그가 표시되지 않으면이 제한에 도달했을 수 있습니다. 한도에 도달했는지 확인하려면 섹션을 참조하세요 [Verify that the debug log mode is active for your playback session](#).

로그 필드 디버그

다음 표에는 MediaTailor가 CloudWatch에 쓰는 디버그 로그 필드가 나열되어 있습니다.

필드	설명
awsAccountId	AWS 계정 ID.
customerId	MediaTailor 고객 ID입니다.
eventTimestamp	디버그 로그 이벤트와 연결된 ISO 8601 타임스탬프입니다.

필드	설명
eventType	디버그 로그 이벤트의 유형입니다. 값: • ORIGIN_INTERACTION - 오리진 서버와의 MediaTailor 상호 작용에 대한 세부 정보입니다. 오리진 매니페스트 응답, 매니페스트 유형 및 오리진 URL을 예로 들 수 있습니다. • GENERATED_MANIFEST - MediaTailor의 재생 세션 응답에 대한 세부 정보입니다. 예를 들어 MediaTailor가 생성하는 매니페스트입니다. • SESSION_INITIALIZED - 세션 ID와 같은 세션 초기화 세부 정보입니다.
originRequestUrl	이 요청에 대해 검색되는 오리진 서버의 URL입니다.
mediaTailorPath	초기 매니페스트 요청에서 MediaTailor에 전달된 파라미터를 포함하여 호출된 MediaTailor 엔드포인트입니다.
requestId	MediaTailor에 대한 특정 HTTP 요청의 ID입니다.
responseBody	MediaTailor의 응답 본문에 있는 매니페스트입니다. 이는 원시 오리진 매니페스트 또는 MediaTailor에서 생성한 매니페스트입니다.
sessionId	재생 세션 ID입니다.
sessionType	재생 세션의 유형입니다. 값: HLS, DASH

디버그 로그 읽기

MediaTailor는 디버그 로그를 Amazon CloudWatch Logs에 기록합니다. 일반적인 CloudWatch Logs 요금이 적용됩니다. CloudWatch Insights를 사용하여 디버그 로그를 읽습니다. CloudWatch Logs Insights를 사용하는 방법에 대한 자세한 내용은 AWS [CloudWatch Logs 사용 설명서의 CloudWatch Logs Insights를 사용하여 로그 데이터 분석을](#) 참조하세요. CloudWatch

Note

디버그 로그가 CloudWatch에 표시되는 데 몇 분 정도 걸릴 수 있습니다. 로그가 나타나지 않으면 몇 분 정도 기다렸다가 다시 시도하십시오. 그래도 로그가 표시되지 않으면 최대 활성 디버그 로그 세션 수에 도달한 것일 수 있습니다. 이 경우 CloudWatch 쿼리를 실행하여 재생 세션에 대해 초기화된 디버그 세션이 있는지 확인합니다. 자세한 내용은 [Verify that the debug log mode is active for your playback session](#) 단원을 참조하십시오.

예제

이 섹션에는 MediaTailor 디버그 로그 데이터를 읽는 데 사용할 수 있는 예제 쿼리가 포함되어 있습니다.

Example 1: 재생 세션에 대해 디버그 로그 모드가 활성 상태인지 확인

```
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: 오리진의 응답 보기

```
fields @timestamp, responseBody, @message, mediaTailorPath
| filter eventType = "ORIGIN_MANIFEST" and sessionId = "32002de2-837c-4e3e-9660-f3075e8dfd90"
```

Example 3: 지정된 세션에 대해 MediaTailor에서 생성된 매니페스트 보기

```
fields @timestamp, responseBody, @message
| filter mediaTailorPath like "/v1/master/" and eventType = "GENERATED_MANIFEST" and
sessionId = "32002de2-837c-4e3e-9660-f3075e8dfd90"
```

Example 4: 지정된에 대한 모든 이벤트 보기 `requestId`

이 쿼리를 사용하여 오리진 매니페스트와 MediaTailor에서 생성된 매니페스트를 봅니다.

```
fields @timestamp, responseBody, @message, mediaTailorPath
```

```
| filter requestId = "e5ba82a5-f8ac-4efb-88a0-55bed21c45b4"
```

Amazon CloudWatch 지표 AWS Elemental MediaTailor 를 사용한 모니터링

CloudWatch를 사용하여 AWS Elemental MediaTailor 지표를 모니터링할 수 있습니다. CloudWatch는 서비스 성능에 대한 원시 데이터를 수집하고 해당 데이터를 읽기 가능하며 실시간에 가까운 지표로 처리합니다. 이러한 통계는 15개월간 보관되므로 기록 정보에 액세스하고 웹 애플리케이션 또는 서비스가 어떻게 실행되고 있는지 전체적으로 더 잘 파악할 수 있습니다. 특정 임계값을 주시하다가 해당 임계값이 충족될 때 알림을 전송하거나 조치를 취하도록 경보를 설정할 수도 있습니다. 자세한 내용은 [Amazon CloudWatch 사용 설명서](#)를 참조하세요.

지표는 오래된 매니페스트를 조사할 때 유용할 수 있습니다. 자세한 내용은 [지표를 사용하여 오래된 매니페스트 진단](#) 단원을 참조하십시오.

지표는 먼저 서비스 네임스페이스별로 그룹화된 다음 각 네임스페이스 내에서 다양한 차원 조합별로 그룹화됩니다.

CloudWatch 콘솔을 사용하여 지표를 보려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 지표(Metrics)를 선택합니다.
3. 모든 지표에서 MediaTailor 네임스페이스를 선택합니다.
4. 지표 차원을 선택하여 지표를 봅니다(예: originID).
5. 확인할 기간을 지정합니다.

AWS Command Line Interface (AWS CLI)를 사용하여 지표를 보려면

- 명령 프롬프트에서 다음 명령을 사용합니다.

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

AWS Elemental MediaTailor CloudWatch 지표

AWS Elemental MediaTailor 네임스페이스에는 다음 지표가 포함됩니다. 이 지표는 기본적으로 계정으로 게시됩니다.

채널 어셈블리(CA) 지표

다음 표에서는 채널 또는 채널 출력별로 모든 지표를 사용할 수 있습니다.

지표	설명
4xxErrorCount	4xx 오류 수입니다.
5xxErrorCount	5xx 오류 수입니다.
RequestCount	요청 총 수 트랜잭션 수는 주로 플레이어가 업데이트된 매니페스트를 요청하는 빈도와 플레이어 수에 따라 달라집니다. 각 플레이어 요청은 트랜잭션으로 간주됩니다.
TotalTime	클라이언트 및 네트워크에서 바이트를 수신하고 쓰는 데 사용된 시간을 포함하여 애플리케이션 서버가 요청을 처리하는 데 걸린 시간입니다.

서버 측 광고 삽입(SSAI) 지표

다음 표에는 서버 측 광고 삽입 지표가 나열되어 있습니다.

지표	설명
AdDecisionServer.Ads	지정한 CloudWatch 기간 내에 광고 결정 서버 (ADS) 응답에 포함된 광고 수입니다. 차원: • ConfigurationName • 계정 수준(차원 없음)
AdDecisionServer.Duration	지정한 CloudWatch 기간 내에 MediaTailor 가 ADS에서 수신한 모든 광고의 총 지속 시간입니다. 이 기간은 지정한 기간보다 클 수 Avail.Duration 있습니다. 차원: ConfigurationName

지표	설명
AdDecisionServer.Errors	지정한 CloudWatch 기간 내에 MediaTailor가 ADS로부터 수신한 HTTP 200이 아닌 상태 코드 응답, 빈 응답 및 시간 초과 응답 수입니다. 차원: ConfigurationName
AdDecisionServer.FillRate	ADS에서 응답이 해당 개별 광고 시간에 대해 지정된 시간을 채우는 비율의 단순 평균. 가중 평균을 얻으려면 AdDecisionServer.Duration 을 Avail.Duration 의 백분율로 계산합니다. 단순 및 가중 평균에 대한 자세한 내용은 단순 및 가중 평균 단원을 참조하십시오. 차원: ConfigurationName
AdDecisionServer.Latency	MediaTailor가 ADS에 보낸 요청에 대한 밀리초 단위의 응답 시간입니다. 차원: ConfigurationName
AdDecisionServer.Timeouts	지정한 CloudWatch 기간의 ADS에 대한 시간 초과 요청 수입니다. 차원: ConfigurationName
AdNotReady	지정한 기간 동안 내부 트랜스코더 서비스에 의해 아직 트랜스코딩되지 않은 광고에 대해 ADS가 가리킨 횟수. 이 지표에 대한 높은 값은 전체적으로 낮은 Avail.FillRate 의 한 원인이 되었을지도 모릅니다. 차원: ConfigurationName

지표	설명
AdsBilled	MediaTailor가 삽입에 따라 고객에게 청구하는 광고 수입입니다. 차원: • ConfigurationName • 계정 수준(차원 없음)
Avail.Duration	CloudWatch 기간 내에 계획된 총 광고 시간 밀리초 수입입니다. 계획된 합계는 오리진 매니페스트의 광고 가능 기간을 기준으로 합니다. 차원: • ConfigurationName • 계정 수준(차원 없음)
Avail.FilledDuration	MediaTailor가 CloudWatch 기간 내에 광고로 채울 계획된 광고 가능 시간 밀리초입니다. 차원: • ConfigurationName • 계정 수준(차원 없음)

지표	설명
Avail.FillRate	CloudWatch 기간 내에 MediaTailor가 개별 광고 시간을 채울 비율의 계획된 단순 평균입니다. 가중 평균을 얻으려면 Avail.FillRate 을 Avail.Duration 의 백분율로 계산합니다. 단순 및 가중 평균에 대한 자세한 내용은 단순 및 가중 평균 단원을 참조하십시오. MediaTailorAvail.FillRate 가 달성할 수 있는 최대값에 의해 제한됩니다. AdDecisionServer.FillRate . Avail.FillRate 가 낮은 경우 AdDecisionServer.FillRate 와 비교합니다. AdDecisionServer.FillRate 가 낮은 경우 ADS가 광고 시간 지속시간에 비해 충분한 광고를 반환하지 않고 있을 수 있습니다. 차원: ConfigurationName
Avail.Impression	서버 측 보고 중에 MediaTailor가 노출 비컨을 실행한 광고 수입니다. MediaTailor는 비컨이 최종적으로 전송되었는지 여부에 관계없이 하나 이상의 노출 URI가 있는 각 광고에 대해 하나의 개수를 내보냅니다. 이 지표를 분모로 사용하여 비디오 완성률(VCR)을 계산합니다(참조 Avail.Complete). 차원: - ConfigurationName - 계정 수준(차원 없음)

지표	설명
Avail.Impression.Fired	MediaTailor가 서버 측 보고 중에 지정된 광고 추적 도메인에 실행한 노출 비컨의 수로, 해당 비컨이 최종적으로 전달되었는지 여부와 무관합니다. 노출 비컨 URIs가 여러 개인 광고는 각 비컨의 각 도메인에 URI당 1개씩 기여합니다. 이 지표는 Avail.Impression.Retried 및와 동일한 비컨당, 도메인당 단위를 사용Avail.Impression.Recovered 하므로 도메인당 3개를 결합할 수 있습니다 (예: Avail.Impression.Retried /는 도메인에 대한 재시도 속도를 Avail.Impression.Fired 제공함). 차원: - AdTrackingDomain - 계정 수준(차원 없음)
Avail.Impression.Retried	서버 측 보고 중에 MediaTailor가 지정된 광고 추적 도메인에 대해 한 번 이상 시도한 노출 비컨 수입니다. 각 비컨은 재시도 횟수에 관계없이 도메인에 한 번 계산됩니다. 이 지표Avail.Impression.Recovered 를와 비교하여 최종적으로 전송된 재시도된 노출 비컨 수를 확인합니다. 차원: - AdTrackingDomain - 계정 수준(차원 없음)

지표	설명
Avail.Impression.Recovered	서버 측 보고 중에 재시도된 후 성공적으로 전달된 노출 비컨의 수입니다. 각 비컨은 도메인에 한 번 계산됩니다. 이러한 비컨은 재시도 없이 광고 추적 도메인에 도달하지 않았을 것입니다. Avail.Impression.Recovered /는 노출 비컨에 대한 재시도 성공률을 Avail.Impression.Retried 제공합니다. 차원: - AdTrackingDomain - 계정 수준(차원 없음)
Avail.Complete	서버 측 보고 중에 MediaTailor가 전체 비컨을 실행한 광고 수입니다. MediaTailor는 비컨이 최종적으로 전송되었는지 여부에 관계없이 추적 URI가 하나 이상 있는 각 광고에 대해 하나의 개수를 내보냅니다. 비디오 완성률(VCR)을 계산하려면 합계를 합계Avail.Complete 로 나눕니다Avail.Impression . 차원: - ConfigurationName - 계정 수준(차원 없음)

지표	설명
Avail.Complete.Fired	MediaTailor가 서버 측 보고 중에 지정된 광고 추적 도메인에 실행한 전체 비컨의 수로, 해당 비컨이 최종적으로 전송되었는지 여부와 무관합니다. 여러 개의 완전한 비컨 URIs가 있는 광고는 각 비컨의 각 도메인에 URI당 하나의 개수를 제공합니다. 이 지표는 Avail.Complete.Retried 및와 동일한 비컨당, 도메인당 단위를 사용Avail.Complete.Recovered 하므로 도메인당 3개를 결합할 수 있습니다(예: Avail.Complete.Retried / 는 도메인에 대한 재시도 속도를 Avail.Complete.Fired 제공함). 차원: - AdTrackingDomain - 계정 수준(차원 없음)
Avail.Complete.Retried	서버 측 보고 중에 MediaTailor가 지정된 광고 추적 도메인에 대해 한 번 이상 시도한 전체 비컨 수입니다. 각 비컨은 재시도 횟수에 관계없이 도메인에 한 번 계산됩니다. 이 지표Avail.Complete.Recovered 를와 비교하여 최종적으로 전송된 재시도된 전체 비컨 수를 확인합니다. 차원: - AdTrackingDomain - 계정 수준(차원 없음)

지표	설명
Avail.Complete.Recovered	서버 측 보고 중에 재시도된 후 성공적으로 전송된 전체 비컨 수입니다. 각 비컨은 도메인에 한 번 계산됩니다. 이러한 비컨은 재시도 없이 광고 추적 도메인에 도달하지 않았을 것입니다. Avail.Complete.Recovered /는 전체 비컨에 대한 재시도 성공률을 Avail.Complete.Retried 제공합니다. 차원: - AdTrackingDomain - 계정 수준(차원 없음)
Avail.ObservedDuration	CloudWatch 기간 내에 발생한 광고 시간의 관측된 총 밀리초 수입니다. Avail.ObservedDuration 는 광고 시간 종료 시 방출되며 광고 시간 동안 매니페스트에 보고된 세그먼트의 기간을 기반으로 합니다. 차원: ConfigurationName
Avail.ObservedFilledDuration	MediaTailor가 CloudWatch 기간 내에 광고로 채운 관찰된 광고 가능 시간 밀리초입니다. 차원: ConfigurationName
Avail.ObservedFillRate	MediaTailor가 CloudWatch 기간 내에 개별 광고 시간을 채운 비율의 관찰된 단순 평균입니다. 첫 번째 CUE-IN 태그에서 HLS 매니페스트에 대해서만 내보내집니다. CUE-IN 태그가 없는 경우 MediaTailor는 이 지표를 내보내지 않습니다. 차원: ConfigurationName

지표	설명
Avail.ObservedSlateDuration	CloudWatch 기간 내에 삽입된 슬레이트의 관찰된 총 밀리초 수입입니다. 차원: ConfigurationName
GetManifest.Age	매니페스트의 총 수명을 밀리초 단위로 표시합니다. 오리진이 매니페스트를 생성하는 시점부터 MediaTailor가 개인화된 매니페스트를 보내는 시점까지 측정됩니다. 매니페스트 수명 측정을 위한 지표에 대한 자세한 내용은 섹션을 참조하세요 지표를 사용하여 오래된 매니페스트 진단. 차원: ConfigurationName
GetManifest.Errors	지정한 CloudWatch 기간에 MediaTailor가 매니페스트를 생성하는 동안 수신된 오류 수입입니다. 차원: ConfigurationName
GetManifest.Latency	매니페스트 생성 요청에 대한 MediaTailor 응답 시간입니다. 매니페스트 수명 측정을 위한 지표에 대한 자세한 내용은 섹션을 참조하세요 지표를 사용하여 오래된 매니페스트 진단. 차원: ConfigurationName

지표	설명
GetManifest.MediaTailorAge	매니페스트가 MediaTailor에 밀리초 단위로 저장된 시간입니다. MediaTailor가 오리진 응답을 수신하는 시점부터 MediaTailor가 개인화된 매니페스트를 보내는 시점까지 측정됩니다. 매니페스트 수명 측정을 위한 지표에 대한 자세한 내용은 섹션을 참조하세요 지표를 사용하여 오래된 매니페스트 진단. 차원: ConfigurationName
Origin.Age	오리진에 매니페스트가 있는 밀리초 단위의 시간입니다. 오리진이 매니페스트를 생성할 때부터 MediaTailor가 오리진 요청을 보낼 때까지 측정됩니다. 모든 origin.* 지표는 오리진에서 직접 이행되는 요청에 대해 내보내집니다. 캐시된 오리진 응답에는 생성되지 않습니다. 매니페스트 수명 측정을 위한 지표에 대한 자세한 내용은 섹션을 참조하세요 지표를 사용하여 오래된 매니페스트 진단. 차원: ConfigurationName
Origin.Errors	지정한 CloudWatch 기간 동안 MediaTailor가 오리진 서버에서 수신한 비 HTTP 200 상태 코드 응답 및 시간 초과 응답 수입니다. 모든 origin.* 지표는 오리진에서 직접 이행되는 요청에 대해 내보내집니다. 캐시된 오리진 응답에는 생성되지 않습니다. 차원: ConfigurationName

지표	설명
Origin.ManifestFileSizeBytes	HLS 및 DASH 모두에 대한 오리진 매니페스트의 파일 크기입니다. 일반적으로이 지표는와 함께 사용됩니다Origin.ManifestFileSizeTooLarge . 모든 origin.* 지표는 오리진에서 직접 이행되는 요청에 대해 내보내집니다. 캐시된 오리진 응답에는 생성되지 않습니다. 차원: ConfigurationName
Origin.ManifestFileSizeTooLarge	매니페스트 크기가 구성된 양보다 큰 오리진의 응답 수입니다. 일반적으로이 지표는와 함께 사용됩니다Origin.ManifestFileSizeBytes . 모든 origin.* 지표는 오리진에서 직접 이행되는 요청에 대해 내보내집니다. 캐시된 오리진 응답에는 생성되지 않습니다. 차원: ConfigurationName
Origin.Timeouts	지정한 CloudWatch 기간 동안 오리진 서버에 대한 시간 초과 요청 수입니다. 모든 origin.* 지표는 오리진에서 직접 이행되는 요청에 대해 내보내집니다. 캐시된 오리진 응답에는 생성되지 않습니다. 차원: ConfigurationName

지표	설명
Requests	모든 요청 유형에서 초당 동시 트랜잭션 수입니다. 트랜잭션 수는 주로 플레이어 수와 플레이어가 업데이트된 매니페스트를 요청하는 빈도에 따라 달라집니다. 각 플레이어 요청은 트랜잭션으로 간주됩니다. 차원: • ConfigurationName • 계정 수준(차원 없음)
SkippedReason.DurationExceeded	ADS가 지정된 가용 기간보다 긴 광고 기간을 반환했기 때문에 가용 시간에 삽입되지 않은 광고 수입니다. 이 지표의 값이 높으면 AdsBilled와 AdDecisionServer.Ads 지표 간의 불일치가 발생할 수 있습니다. 건너뛴 광고 이유에 대한 자세한 내용은 섹션을 참조하세요 광고 건너뛰기 문제 해결. 차원: • ConfigurationName • 계정 수준(차원 없음)
SkippedReason.EarlyCueIn	초기 로 인해 건너뛴 광고 수입니다CUE-IN. 차원: • ConfigurationName • 계정 수준(차원 없음)

지표	설명
<code>SkippedReason.ImportError</code>	가져오기 작업의 오류로 인해 건너뛴 광고 수입입니다. 차원: • <code>ConfigurationName</code> • 계정 수준(차원 없음)
<code>SkippedReason.ImportInProgress</code>	기존 활성 가져오기 작업으로 인해 건너뛴 광고 수입입니다. 차원: • <code>ConfigurationName</code> • 계정 수준(차원 없음)
<code>SkippedReason.InternalError</code>	MediaTailor 내부 오류로 인해 건너뛴 광고 수입입니다. 차원: • <code>ConfigurationName</code> • 계정 수준(차원 없음)
<code>SkippedReason.NewCreative</code>	클라이언트가 자산을 처음 요청했기 때문에 가용성에 삽입되지 않은 광고 수입입니다. 이 지표의 값이 높으면 자산이 성공적으로 트랜스코딩 <code>Avail.FillRate</code> 될 때까지 일시적으로 전체가 낮아질 수 있습니다. 차원: • <code>ConfigurationName</code> • 계정 수준(차원 없음)

지표	설명
SkippedReason.NoVariantMatch	광고와 콘텐츠 간에 변형이 일치하지 않아 건너뛴 광고 수입입니다. 차원: • ConfigurationName • 계정 수준(차원 없음)
SkippedReason.PersonalizationThresholdExceeded	이 구성에서 개인화 임계값 설정을 초과하는 광고 기간입니다. 차원: • ConfigurationName • 계정 수준(차원 없음)
SkippedReason.ProfileNotFound	트랜스코딩 프로파일을 찾을 수 없어 건너뛴 광고 수입입니다. 차원: • ConfigurationName • 계정 수준(차원 없음)
SkippedReason.TranscodeError	트랜스코딩 오류로 인해 건너뛴 광고 수입입니다. 차원: • ConfigurationName • 계정 수준(차원 없음)

지표	설명
SkippedReason.TranscodeInProgress	광고가 아직 트랜스코딩되지 않았기 때문에 가능 구간에 삽입되지 않은 광고 수입입니다. 이 지표의 값이 높으면 자산이 성공적으로 트랜스코딩 Avail.FillRate 될 때까지 일시적으로 전체가 낮아질 수 있습니다. 차원: • ConfigurationName • 계정 수준(차원 없음)

Note

HLS Interstitials 세션의 경우 광고 결정의 지연 바인딩 특성으로 인해 일부 지표가 다르게 작동합니다.

- Avail.ObservedFilledDuration MediaTailor는 실제 클라이언트 측 재생 동작을 관찰할 수 Avail.FilledDuration 없으므로 일치합니다.
- Avail.ObservedSlateDuration는 관찰된 재생이 아닌 자산 목록 응답에서 계획된 슬레이트 기간을 보고합니다.
- "Observed" 접두사가 붙은 지표는 HLS Interstitials 세션의 예상 값을 제공합니다.

합수 지표

합수를 재생 구성과 함께 사용하면 [다음 지표가](#) 게시됩니다. 이러한 지표는 합수가 수명 주기 후크에 구성된 경우 항상 내보내집니다. 옵트인 또는 로그 구성은 필요하지 않습니다.

후크 수준 지표 - 수명 주기 후크 실행당 하나의 데이터 포인트:

지표	설명
PreSessionInitHook.Invocations	세션 전 초기화 후크가 호출된 횟수입니다. 차원: ConfigurationName

지표	설명
PreSessionInitHook.Errors	오류가 발생한 세션 전 초기화 후크 실행 수입니다. 차원: ConfigurationName
PreSessionInitHook.Latency	세션 전 초기화 후크의 밀리초 단위 실행 시간입니다. 차원: ConfigurationName
PreAdsRequestHook.Invocations	사전 광고 요청 후크가 호출된 횟수입니다. 차원: ConfigurationName
PreAdsRequestHook.Errors	오류가 발생한 사전 광고 요청 후크 실행 수입니다. 차원: ConfigurationName
PreAdsRequestHook.Latency	사전 광고 요청 후크의 밀리초 단위 실행 시간입니다. 차원: ConfigurationName
PostAdsResponseHook.Invocations	광고 후 응답 후크가 호출된 횟수입니다. 차원: ConfigurationName
PostAdsResponseHook.Errors	오류가 발생한 사후 광고 응답 후크 실행 수입니다. 차원: ConfigurationName
PostAdsResponseHook.Latency	광고 후 응답 후크의 밀리초 단위 실행 시간입니다. 차원: ConfigurationName

지표	설명
PreManifestInsertionHook.Invocations	사전 매니페스트 삽입 후크가 호출된 횟수입니다. 차원: ConfigurationName
PreManifestInsertionHook.Errors	오류가 발생한 사전 매니페스트 삽입 후크 실행 수입니다. 차원: ConfigurationName
PreManifestInsertionHook.Latency	매니페스트 전 삽입 후크의 밀리초 단위 실행 시간입니다. 차원: ConfigurationName

함수 수준 지표 - 개별 함수 실행당 하나의 데이터 포인트입니다. 이러한 지표에는 추가 차원 (FunctionId, FunctionType, HookType)이 포함되어 있으므로 어떤 특정 함수가 느리거나 실패하는지 식별할 수 있습니다.

지표	설명
Function.Invocations	개별 함수가 실행된 횟수입니다. 차원: - ConfigurationName - ConfigurationName , FunctionId , FunctionType , HookType
Function.Errors	오류가 발생한 개별 함수 실행 수입니다. 차원: - ConfigurationName - ConfigurationName , FunctionId , FunctionType , HookType

지표	설명
Function.Latency	개별 함수의 밀리초 단위 실행 시간입니다. 차원: - ConfigurationName - ConfigurationName , FunctionId , FunctionType , HookType

단순 및 가중 평균

ADS에서 MediaTailor의 광고 요청에 대한 응답과 MediaTailor가 광고 시간을 채우는 방법에 대한 단순 평균 및 가중 평균을 검색할 수 있습니다.

- 단순 평균이 AdDecisionServer.FillRate와 Avail.FillRate에 제공됩니다. 해당 기간 동안의 개별 광고 시간의 채움 비율(백분율)의 평균입니다. 단순 평균은 개별 광고 시간의 지속시간 간 차이를 감안하지 않습니다.
- 가중 평균은 모든 광고 시간 지속시간의 합계에 대한 채움 비율(백분율)입니다. 단순 평균과 가중 평균은 $(\text{AdDecisionServer.Duration} \times 100) / \text{Avail.Duration}$ 과 $(\text{Avail.FilledDuration} \times 100) / \text{Avail.Duration}$ 으로 계산됩니다. 이러한 평균은 각 광고 시간의 지속 시간의 차이를 감안하여 지속시간이 길수록 더 많은 가중치를 부여합니다.

단 하나의 광고 시간이 포함된 기간의 경우 AdDecisionServer.FillRate로 제공되는 단순 평균은 $(\text{AdDecisionServer.Duration} \times 100 / \text{Avail.Duration})$ 으로 계산되는 가중 평균과 동일한 값입니다. Avail.FillRate로 제공되는 단순 평균은 $(\text{Avail.FilledDuration} \times 100) / \text{Avail.Duration}$ 으로 계산되는 가중 평균과 동일한 값입니다.

예제

지정한 기간에 다음과 같이 2가지 광고 시간이 있다고 가정합니다.

- 첫 번째 광고 시간의 지속시간은 90초입니다.
 - 광고 시간에 대한 ADS 응답은 45초의 광고를 제공합니다(50% 채움).
 - MediaTailor는 45초 분량의 광고 시간을 채웁니다(50% 채워짐).
- 두 번째 광고 시간의 지속시간은 120초입니다.
 - 광고 시간에 대한 ADS 응답은 120초의 광고를 제공합니다(100% 채움).

- MediaTailor는 90초 분량의 광고 시간을 채웁니다(75% 채워짐).

지표는 다음과 같습니다.

- Avail.Duration은 두 광고 시간 지속시간의 합인 201입니다($90 + 120$).
- AdDecisionServer.Duration은 두 응답 지속시간의 합인 165입니다($45 + 120$).
- Avail.FilledDuration은 두 채워진 지속시간의 합인 135입니다($45 + 90$).
- AdDecisionServer.FillRate는 각 광고 시간에 대한 백분율의 평균인 75%입니다($((50\% + 100\%) / 2)$). 이것은 단순 평균입니다.
- ADS 채움 비율의 가중 평균은 AdDecisionServer.Duration으로서 Avail.Duration의 평균인 78.57%입니다($((165 * 100) / 210)$). 이 계산을 통해 지속시간 간의 차이를 설명할 수 있습니다.
- Avail.FillRate는 각 광고 시간에 대한 채움 백분율의 평균인 62.5%입니다($((50\% + 75\%) / 2)$). 이것은 단순 평균입니다.
- MediaTailor 가용 채우기 속도의 가중 평균은 64.29%이며, 이는 Avail.Duration: $(135 * 100) / 210$ 의 백분율Avail.FilledDuration로 표시됩니다. 이 계산을 통해 지속시간 간의 차이를 설명할 수 있습니다.

MediaTailorAvail.FillRate가 모든 광고 가능 구간에 대해 달성할 수 있는 가장 높은 값은 100%입니다. ADS는 가용 영역에서 사용 가능한 시간보다 더 많은 광고 시간을 반환할 수 있지만 MediaTailor는 사용 가능한 시간만 채울 수 있습니다.

AWS Elemental MediaTailor CloudWatch 차원

다음 차원을 사용하여 AWS Elemental MediaTailor 데이터를 필터링할 수 있습니다.

차원	설명
ConfigurationName	지표가 속하는 구성을 나타냅니다. 도메인별 비컨 지표(Avail.Impression.Fired, , Avail.Impression.Retried, , Avail.Impression.Recovered, 및 Avail.Complete.Recovered)를 제외한 모든 지표에서 사용할 Avail.Complete.Fired, Avail.Complete.Retried 수 있습니다.

차원	설명
AdTrackingDomain	하위 도메인이 제거된 광고 추적 비컨 대상의 등록 가능한 도메인입니다. 등록 가능한 도메인으로 확인할 수 없는 호스트는 로 보고됩니다unknown. Avail.Impression.Fired , , Avail.Impression.Retried , Avail.Impression.Recovered Avail.Complete.Fired , 및 에서 사용할 Avail.Complete.Retried 수 있습니다Avail.Complete.Recovered .
FunctionId	함수의 식별자입니다. Function.* 지표에서만 사용할 수 있습니다.
FunctionType	함수 유형: CUSTOM_OUTPUT , HTTP_REQUEST , VAST_REQUEST SEQUENTIAL_EXECUTOR , 또는 CONCURRENT_EXECUTOR . Function.* 지표에서만 사용할 수 있습니다.
HookType	함수를 트리거한 수명 주기 후크: PRE_SESSION_INITIALIZATION , PRE_AD_REQUEST POST_AD_RESPONSE , 또는 PRE_MANIFEST_INSERTION . Function.* 지표에서만 사용할 수 있습니다.

지표를 사용하여에서 오래된 매니페스트 진단 AWS Elemental MediaTailor

오래된 매니페스트는 최근에 업데이트되지 않은 매니페스트입니다. 다양한 요인(예: 다운스트림 시스템 요구 사항)에 따라 광고 삽입 워크플로마다 매니페스트가 기한 경과로 간주되기 전에가 경과해야 하는 시간에 대한 허용 오차가 다를 수 있습니다. Amazon CloudWatch 지표를 사용하여 워크플로의 기한 경과 허용치를 초과하는 매니페스트를 식별하고 매니페스트 업데이트 지연의 원인을 식별할 수 있습니다.

다음 지표는 오래된 매니페스트와 그 원인을 식별하는 데 도움이 됩니다.

MediaTailor가 내보내는 모든 지표에 대한 자세한 내용은 섹션을 참조하세요 [AWS Elemental MediaTailor CloudWatch 지표](#).

지표	정의	사용
GetManifest.Age	이 구성에 Origin.Age 대해 GetManifest.MediaTailorAge 및를 모두 포함하여 매니페스트의 총 수명을 측정합니다.	이 지표를 사용하여 업데이트 임계값을 초과하고 오래된 매니페스트를 식별할 수 있습니다. 기한 경과 매니페스트가 제공될 때 알림을 받도록 이 지표에 대한 경보를 설정합니다. 경보에 대한 자세한 내용은 Amazon CloudWatch 사용 설명서의 지표에 대한 경보를 참조하세요. Amazon CloudWatch 경보가 수신되면 Origin.Age 및 GetManifest.MediaTailorAge 를 사용하여 MediaTailor 또는 오리진이 기한 경과를 유발하는지 확인합니다.
Origin.Age	이 구성을 위해 MediaTailor로 보내기 전에 오리진에 매니페스트가 있는 기간을 측정합니다. 응답이 콘텐츠 전송 네트워크(CDN)에서 오는 경우 이 지표는 내보내지지 않습니다. 를 내보내 Origin.Age려면 오리진에서 응답을 가져와야 합니다.	를 사용하여 오래된 매니페스트를 식별할 때 Origin.Age 지표와 지표를 분석하여 어떤 GetManifest.MediaTailorAge 가 매니페스트 기한 경과와 원인이 되는지 확인할 GetManifest.Age 수 있습니다. Origin.Age 가 오리진에서 일반적인 처리 시간보다 길면 업스트림 시스템에서 문제가 발생하고 진단에 집중해야 할 수 있습니다.
GetManifest.MediaTailorAge	MediaTailor가 이 구성을 위해 매니페스트를 저장한 기간을 측정합니다.	를 사용하여 오래된 매니페스트를 식별할 때 GetManifest.MediaTailorAge 지표와 지표를 분석하여 어떤 Origin.Age 가 매니페스트 기한 경과와 원인이 되는지 확인할 GetManifest.Age 수 있습니다.

지표	정의	사용
		<code>GetManifest.MediaTailorAge</code> 가 MediaTailor의 일반적인 매니페스트 개인화 시간보다 긴 경우 MediaTailor가 문제를 일으킬 가능성이 높으므로 여기에 진단에 집중해야 합니다. <code>GetManifest.Latency</code> 는 MediaTailor가 개인화된 매니페스트를 생성하는 데 걸리는 시간을 추가로 식별할 수 있습니다.
<code>GetManifest.Latency</code>	MediaTailor가 요청을 처리하고이 구성에 대한 개인화된 매니페스트를 생성하는 데 걸리는 시간을 측정합니다.	<code>Origin.Age</code> 및 <code>GetManifest.Latency</code> 를 비교 <code>GetManifest.MediaTailorAge</code> 하고 MediaTailor가 지연된 매니페스트 전송의 원인이라고 판단하면 지표를 분석하여 매니페스트 개인화 프로세스가 매니페스트 기한 초과에 기여하는지 확인할 수 있습니다. <code>GetManifest.MediaTailorAge</code> 는 매니페스트가 MediaTailor에 저장되는 총 시간을 측정합니다. 는 MediaTailor가 요청에 대한 응답으로 매니페스트를 개인화하는 스토리지 시간의 양을 <code>GetManifest.Latency</code> 측정합니다.

AWS Elemental MediaTailor API 호출 기록

AWS Elemental MediaTailor 는 MediaTailor에서 사용자 AWS CloudTrail, 역할 또는 서비스가 수행한 작업에 대한 레코드를 제공하는 AWS 서비스와 통합됩니다. CloudTrail은 MediaTailor에 대한 모든 API 호출을 이벤트로 캡처합니다. 캡처되는 호출에는 MediaTailor 콘솔의 호출과 MediaTailor API 작업에 대한 코드 호출이 포함됩니다. 추적을 생성하면 MediaTailor 이벤트를 포함한 CloudTrail 이벤트를 지

속적으로 Amazon S3 버킷에 배포할 수 있습니다. 추적을 구성하지 않은 경우에도 이벤트 기록에서 CloudTrail 콘솔의 최신 이벤트를 볼 수 있습니다. CloudTrail에서 수집한 정보를 사용하여 MediaTailor에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

CloudTrail에 대한 자세한 설명은 [AWS CloudTrail 사용자 가이드](#)를 참조하십시오.

AWS Elemental MediaTailor CloudTrail의 정보

AWS 계정을 생성할 때 계정에서 CloudTrail이 활성화됩니다. 활동이 발생하면 AWS Elemental MediaTailor 해당 활동이 이벤트 기록의 다른 AWS 서비스 이벤트와 함께 CloudTrail 이벤트에 기록됩니다. AWS 계정에서 최근 이벤트를 보고 검색하고 다운로드할 수 있습니다. 자세한 내용은 [CloudTrail 이벤트 기록을 사용하여 이벤트 보기](#)를 참조하세요.

에 대한 이벤트를 포함하여 AWS 계정에 이벤트를 지속적으로 기록하려면 추적을 AWS Elemental MediaTailor 생성합니다. CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 기본적으로 콘솔에서 추적을 생성하면 추적이 모든 AWS 리전에 적용됩니다. 추적은 AWS 파티션의 모든 리전에서 이벤트를 로깅하고 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 조치를 취하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 내용은 다음을 참조하세요.

- [AWS 계정에 대한 추적 생성](#)
- [AWS CloudTrail 로그와의 서비스 통합](#)
- [CloudTrail에 대한 Amazon SNS 알림 구성](#)
- [여러 리전에서 CloudTrail 로그 파일 받기](#) 및 [여러 계정에서 CloudTrail 로그 파일 받기](#)

모든 AWS Elemental MediaTailor 작업은 CloudTrail에서 로깅되며 [AWS Elemental MediaTailor API 참조](#)에 문서화됩니다. 예를 들어, PutPlaybackConfiguration 및 ListPlaybackConfigurations 작업에 대한 호출은 CloudTrail 로그 파일의 항목을 생성합니다.

모든 이벤트 및 로그 항목에는 요청을 생성한 사용자에 대한 정보가 들어 있습니다. ID 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 루트 사용자 또는 AWS Identity and Access Management (IAM) 자격 증명으로 요청이 이루어졌는지 여부
- 역할 또는 페더레이션 사용자에게 대한 임시 보안 자격 증명을 사용하여 요청이 생성되었는지 여부
- 요청이 다른 AWS 서비스에서 이루어졌는지 여부

자세한 내용은 [CloudTrail userIdentity 요소](#)를 참조하십시오.

AWS Elemental MediaTailor 로그 파일 항목 이해

트레이일은 지정된 S3 버킷에 이벤트를 로그 파일로 입력할 수 있게 하는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함될 수 있습니다. 이벤트는 모든 소스로부터의 단일 요청을 나타내며 요청 작업, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보가 들어 있습니다. CloudTrail 로그 파일은 퍼블릭 API 직접 호출의 주문 스택 트레이스가 아니므로 특정 순서로 표시되지 않습니다.

다음은 PutPlaybackConfiguration 태스크를 보여 주는 CloudTrail 로그 항목이 나타낸 예제입니다.

```
{
  "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://exampleleads.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://exampleleads.com",
    "CdnConfiguration": {},
  }
}
```

```

    "PlaybackEndpointPrefix": "https://777788889999.mediatailor.us-east-1.amazonaws.com",
    "HlsConfiguration": {
        "ManifestEndpointPrefix": "https://777788889999.mediatailor.us-east-1.amazonaws.com/v1/master/AKIAIOSFODNN7EXAMPLE/exemplename/"
    },
    "VideoContentSourceUrl": "http://examplevideo.com",
    "Name": "exemplename"
  },
  "requestID": "1a2b3c4d-1234-5678-1234-1a2b3c4d5e6f",
  "eventID": "987abc65-1a2b-3c4d-5d6e-987abc654def",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}

```

다음은 GetPlaybackConfiguration 태스크를 보여 주는 CloudTrail 로그 항목이 나타낸 예제입니다.

```

{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAEXAMPLE",
    "arn": "arn:aws:iam::111122223333:user/testuser",
    "accountId": "111122223333",
    "accessKeyId": "AIDAEXAMPLE",
    "userName": "testuser"
  },
  "eventTime": "2018-12-28T22:52:37Z",
  "eventSource": "mediatailor.amazonaws.com",
  "eventName": "GetPlaybackConfiguration",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "1.2.3.4",
  "userAgent": "PostmanRuntime/7.4.0",
  "requestParameters": {
    "Name": "exemplename"
  },
  "responseElements": null,
  "requestID": "0z1y2x3w-0123-4567-9876-6q7r8s9t0u1v",
  "eventID": "888ddd77-3322-eeew-uuii-abc123jkl343",
  "readOnly": true,
  "eventType": "AwsApiCall",

```

```
"recipientAccountId": "111122223333"
}
```

AWS Elemental MediaTailor 채널 어셈블리 알림 수신

MediaTailor는 채널 어셈블리 리소스에서 발생하는 문제 또는 잠재적 문제에 대한 알림을 생성합니다. 알림은 문제, 문제가 발생한 시간 및 영향을 받는 리소스를 설명합니다.

MediaTailor ListAlerts API를 사용하여, AWS Management Console AWS Command Line Interface (AWS CLI), AWS SDKs 또는 프로그래밍 방식으로 알림을 볼 수 있습니다. [ListAlerts](#)

Important

알림은 2021년 7월 14일 이후에 생성된 채널 어셈블리 리소스에만 사용할 수 있습니다.

채널 어셈블리 알림

알림 유형	알림 코드	알림 메시지	참고
VOD 소스	NOT_PROCESSED	MediaTailor가 패키지 구성 <i>configurationPath</i> 를 처리하지 않았습니다.	
	UNREACHABLE	URL <i>URL</i> 에 연결할 수 없습니다.	
	UNAUTHORIZED	URL이 요청을 승인하지 않았습니다.	
	TIMEOUT	<i>URL</i> 에 대한 연결이 시간 초과되었습니다.	
	UNPARSABLE_MANIFEST	MediaTailor가 <i>URL</i> 에서 매니페스트를 구문 분석하는 동안 문제가 발생했습니다.	
	VARIANT_DURATION_MISMATCH	MediaTailor는 <i>URL</i> 에서 매니페스트를 구문 분석하는 동안 총 기간이 일치하지 않는 변형을	매니페스트의 기간은 변형/표현에 따라 다릅니다. 이로 인해 캡션이 누락되거나 잘못되어

알림 유형	알림 코드	알림 메시지	참고
		발견했습니다. 이로 인해 재생 중에 정지가 발생할 수 있습니다.	MediaTailor가 광고를 삽입할 수 없게 될 수 있습니다.
	SEGMENT_DURATION_TOO_LONG	MediaTailor는 <i>url</i> 에서 매니페스트를 구문 분석하는 동안 지속 시간이 30초 이상인 세그먼트를 발견했습니다. 이로 인해 재생 중 정지, 누락되거나 잘못된 캡션, 광고를 삽입할 수 없게 될 수 있습니다.	매니페스트에는 30초보다 큰 세그먼트가 포함되어 있습니다.
	TARGET_DURATION_MISMATCH	MediaTailor에서 <i>url</i> 에서 매니페스트를 구문 분석하는 동안 HLS 매니페스트 간에 EXT-X-TARGETDURATION 값이 일치하지 않았습니다. 이로 인해 재생 중에 정지가 발생할 수 있습니다.	대상 기간은 소스의 모든 매니페스트에서 일치하지 않습니다.
소스 위치	NOT_PROCESSED	MediaTailor가 리소스 <i>resourceName</i> 을 처리하지 않았습니다.	
프로그램	VOD_SOURCE_ALERT	이 프로그램의 VOD 소스 <i>vodSourceName</i> 에는 <i>vodSourceAlertCode</i> : <i>vodSourceAlertMessage</i> 알림이 있습니다.	
	SOURCE_LOCATION_ALERT	이 프로그램에 포함된 소스 위치 <i>sourceLocationName</i> 에는 <i>sourceLocationAlertCode</i> : <i>sourceLocationAlertMessage</i> 알림이 있습니다.	

알림 유형	알림 코드	알림 메시지	참고
	CODEC_MISMATCH	MediaTailor가 <i>channelName</i> 일정에서 일치하지 않는 코덱을 발견했습니다. <i>programName1</i> 1의 매니페스트 manifestUrl과 <i>programName2</i> 의 매니페스트 manifestUrl 간에 <i>sourceGroup</i> <i>programName2</i> 이 일치하지 않습니다. <i>manifestUrl manifestUrl</i>	
	RESOLUTION_MISMATCH	MediaTailor가 <i>channelName</i> 일정에서 일치하지 않는 해상도를 발견했습니다. <i>programName1</i> 1의 매니페스트 manifestUrl과 <i>programName2</i> 의 매니페스트 manifestUrl 간에 <i>sourceGroup</i> <i>programName2</i> 이 일치하지 않습니다. <i>manifestUrl manifestUrl</i>	

알림 유형	알림 코드	알림 메시지	참고
	BANDWIDTH_MISMATCH	MediaTailor가 <i>channelName</i> 일정에서 일치하지 않는 대역폭을 발견했습니다. <i>programName1</i> 1의 매니페스트 manifestUrl과 <i>programName2</i> 의 매니페스트 manifestUrl 간에 <i>sourceGroup</i> <i>programName2</i> 이 일치하지 않습니다. <i>manifestUrl manifestUrl</i>	
	FRAMERATE_MISMATCH	<i>channelName</i> 일정에서 MediaTailor 프레임 속도가 일치하지 않습니다. <i>programName1</i> 1의 매니페스트 manifestUrl과 <i>programName2</i> 의 매니페스트 manifestUrl 간에 <i>sourceGroup</i> <i>programName2</i> 이 일치하지 않습니다. <i>manifestUrl manifestUrl</i>	

알림 유형	알림 코드	알림 메시지	참고
	TARGET_DURATION_MISMATCH	MediaTailor가 <i>channelName</i> 일정의 HLS 매니페스트에서 일치하지 않는 EXT-X-TARGETDURATION 값을 발견했습니다. <i>programName1</i> 1의 매니페스트 manifestUrl과 <i>programName2</i> 의 매니페스트 manifestUrl 간에 <i>sourceGroup</i> <i>programName2</i> 이 일치하지 않습니다. <i>manifestUrl manifestUrl</i>	
	SEGMENT_DURATION_MISMATCH	MediaTailor가 <i>channelName</i> 일정의 매니페스트에서 일치하지 않는 세그먼트 지속 시간 값을 발견했습니다. <i>programName1</i> 1의 매니페스트 manifestUrl과 <i>programName2</i> 의 매니페스트 manifestUrl 간에 <i>sourceGroup</i> <i>programName2</i> 이 일치하지 않습니다. <i>manifestUrl manifestUrl</i>	
	NO_COMMON_SEGMENT_BOUNDARY_FOR_AD_SLATE	MediaTailor가 <i>programName</i> 프로그램용 오프셋 <i>offsetMillis</i> 에 광고 슬레이트를 삽입할 수 없습니다. 광고 슬레이트의 시작 시간에는 공통 세그먼트 경계가 없습니다.	

알림 유형	알림 코드	알림 메시지	참고
	NOT_PROCESSED	MediaTailor가 리소스 <i>resourceName</i> 을 처리하지 않았습니다.	
	TOO_MANY_ALERTS	MediaTailor에서 알림이 너무 많아 <i>programName</i> 에 대한 알림을 더 이상 제공하지 않습니다. <i>programName</i> 에 대한 알림을 계속 받으려면 기존 알림을 지웁니다.	
채널	PROGRAM_ALERT	이 채널에 포함된 프로그램 <i>programName</i> 에는 <i>programAlertCode</i> : <i>programAlertMessage</i> <i>programAlertMessage</i> 알림이 있습니다.	

매니페스트 관련 알림 문제 해결

UNPARSABLE_MANIFEST, VARIANT_DURATION_MISMATCH 또는 같은 매니페스트 관련 알림이 발생하면 다음을 TARGET_DURATION_MISMATCH 확인합니다.

1. 소스 요구 사항 확인: 콘텐츠가에 설명된 요구 사항을 충족하는지 확인합니다 [the section called “콘텐츠 소스 통합”](#).
2. 매니페스트 로그 확인: 자세한 매니페스트 오류 분석 및 이벤트 유형은 섹션을 참조하세요 [the section called “매니페스트 로그”](#).
3. 오류 코드 확인: 자세한 오류 분석은 섹션을 참조하세요 [문제 해결](#).
4. 매니페스트 형식 검토: 매니페스트가 스트리밍 프로토콜(HLS 또는 DASH)의 올바른 형식 사양을 따르는지 확인합니다.

알림 보기

모든 MediaTailor 채널 어셈블리 리소스에 대한 알림을 볼 수 있습니다. 채널 및 프로그램에 대한 알림을 볼 때 MediaTailor에는 채널 또는 프로그램에 포함된 모든 관련 리소스가 포함됩니다. 예를 들어 특정 프로그램에 대한 알림을 볼 때 프로그램에 포함된 소스 위치 및 VOD 소스에 대한 알림도 표시됩니다.

알림을 보려면 다음 절차를 수행합니다.

Console

콘솔에서 알림을 보려면

1. <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.
2. 알림을 보려는 리소스를 선택합니다.
3. 알림을 보려면 알림 탭을 선택합니다.

AWS Command Line Interface (AWS CLI)

채널 어셈블리 리소스에 대한 알림을 나열하려면 리소스의 [Amazon 리소스 이름\(ARN\)](#)이 필요합니다. AWS Command Line Interface (AWS CLI)에서 `describe-resource_type` 명령을 사용하여 리소스의 ARN을 가져올 수 있습니다. 예를 들어 [describe-channel](#) 명령을 실행하여 특정 채널의 ARN을 가져옵니다.

```
aws mediatailor describe-channel --channel-name MyChannelName
```

그런 다음 [aws mediatailor list-alerts](#) 명령을 사용하여 리소스와 연결된 알림을 나열합니다.

```
aws mediatailor list-alerts --resource-arn arn:aws:mediatailor:region:aws-account-id:resource-type/resource-name
```

API

채널 어셈블리 리소스에 대한 알림을 나열하려면 리소스의 [Amazon 리소스 이름\(ARN\)](#)이 필요합니다. MediaTailor API에서 `DescribeResource` 작업을 사용하여 리소스의 ARN을 가져올 수 있습니다. 예를 들어 [DescribeChannel](#) 작업을 사용하여 특정 채널의 ARN을 가져옵니다.

그런 다음 [ListAlerts](#) API를 사용하여 리소스에 대한 알림을 나열합니다.

알림 처리

알림이 발생하면에서 알림을 보거나 AWS Command Line Interface (AWS CLI), AWS SDKs 또는 MediaTailor Alerts API를 AWS Management Console 사용하여 문제의 가능한 원인을 확인합니다.

문제를 해결하면 MediaTailor가 알림을 지웁니다.

AWS Elemental MediaTailor 리소스 태그 지정

태그는 사용자가 할당하거나가 AWS 리소스에 AWS 할당하는 메타데이터 레이블입니다. 각 태그는 키와 값으로 구성됩니다. 사용자가 할당하는 태그에 대해 키와 값을 정의합니다. 예를 들어 키를 stage로 정의하고 리소스 하나의 값을 test로 정의할 수 있습니다.

태그는 다음을 지원합니다.

- AWS 리소스를 식별하고 구성합니다. 많은 AWS 서비스가 태그 지정을 지원하므로 서로 다른 서비스의 리소스에 동일한 태그를 할당하여 리소스가 관련이 있음을 나타낼 수 있습니다. 예를 들어 AWS Elemental MediaTailor 구성에 할당한 AWS Elemental MediaPackage 채널 및 엔드포인트에 동일한 태그를 할당할 수 있습니다.
- AWS 비용을 추적합니다. AWS 결제 및 비용 관리 대시보드에서 이러한 태그를 활성화합니다.는 태그를 AWS 사용하여 비용을 분류하고 월별 비용 할당 보고서를 제공합니다. 자세한 내용은 [AWS Billing 사용자 가이드](#)의 [비용 할당 태그 사용](#)을 참조하세요.
- AWS 리소스에 대한 액세스를 제어합니다. 자세한 내용은 [IAM 사용 설명서](#)의 [태그를 사용한 액세스 제어](#)를 참조하십시오.

다음 섹션에서는 태그에 대한 자세한 정보를 제공합니다 AWS Elemental MediaTailor.

에서 지원되는 리소스 AWS Elemental MediaTailor

의 다음 리소스는 태그 지정을 AWS Elemental MediaTailor 지원합니다.

- 채널
- 구성
- SourceLocations
- VodSources

태그 제한 사항

AWS Elemental MediaTailor 리소스의 태그에는 다음과 같은 기본 제한이 적용됩니다.

- 리소스에 할당할 수 있는 최대 태그 수 - 50개
- 최대 키 길이 - 유니코드 128자
- 최대 값 길이 - 유니코드 256자
- 키 및 값에 사용할 수 있는 문자 - a-z, A-Z, 0-9, 공백 및 _ . : / = + - @ 문자
- 키와 값은 대/소문자를 구분합니다.
- 키 접두사로 aws:를 사용하지 마세요. AWS 전용입니다.

에서 태그 관리 AWS Elemental MediaTailor

태그를 리소스의 속성으로 설정합니다. AWS Elemental MediaTailor API 또는 AWS Command Line Interface ()를 통해 태그를 추가, 편집 및 삭제할 수 있습니다AWS CLI. 자세한 내용은 [AWS Elemental MediaTailor API 참조](#)를 참조하세요.

분석 대시보드를 사용하여 광고 삽입 성능 모니터링

MediaTailor 분석 대시보드는 MediaTailor가 서버 측 광고 삽입(SSAI)을 위해 게시하는 CloudWatch 지표를 단일 보기로 제공하는 콘솔 페이지입니다. 이를 사용하여 사용하는 모든 AWS 리전에서 수익 상태를 모니터링하고, 회귀를 조기에 찾아내고, 자체 CloudWatch 대시보드를 구축하지 않고도 AWS 리전 간의 성능을 비교할 수 있습니다. 대시보드에는 재생 구성 및 광고 추적 도메인에 대한 주요 수익화 지표, 추세 그래프 및 지표 테이블이 표시됩니다.

대시보드는 사용자를 대신하여 CloudWatch를 쿼리하므로 다음과 같은 질문에 답할 수 있습니다.

- 이번 주 전체 가중 채우기 비율은 얼마이며 지난 주와 어떻게 비교되나요?
- 노출 비컨 재시도를 가장 많이 유발하는 광고 추적 도메인은 무엇입니까?
- 한 AWS 리전의 광고 삽입 성능은 다른 AWS 리전과 어떻게 비교되나요?

이 페이지에 지정된 분석 대시보드에서 사용되는 기본 CloudWatch 지표에 대한 정의는 섹션을 참조하세요[Amazon CloudWatch 지표 AWS Elemental MediaTailor 를 사용한 모니터링](#). 이러한 지표에 사용자 지정 경고 또는 자체 CloudWatch 대시보드를 빌드하려면 CloudWatch 콘솔을 직접 사용합니다.

Note

분석 대시보드의 지표 테이블은 계정의 CloudWatch ListMetrics API 요청 사용량에 포함되는 CloudWatch API를 호출합니다. 주요 수익 지표 및 추세 그래프는 호출하지 않습니다. ListMetrics. 요금은 [Amazon CloudWatch 요금](#)을 참조하세요.

분석 대시보드 사용

분석 대시보드는 MediaTailor 콘솔의 최상위 페이지입니다. 열려면 로그인하고 탐색 창에서 선택합니다.

분석 대시보드를 열려면

1. <https://console.aws.amazon.com/mediatailor/> MediaTailor 콘솔을 엽니다.
2. 탐색 창의 광고 삽입에서 분석 대시보드를 선택합니다.

페이지가 열리면 대시보드는 현재 로그인한 AWS 리전에서 지난 1일 동안의 지표를 로드합니다. 언제든지 시간 범위 및 리전 선택을 변경할 수 있으며 페이지의 각 섹션은 새 선택에 대해 다시 로드됩니다.

분석 대시보드를 사용하려면 IAM 보안 주체가 MediaTailor 재생 구성을 나열하고 CloudWatch 지표를 읽을 수 있어야 합니다. 최소한 다음 작업이 필요합니다.

- mediatailor:ListPlaybackConfigurations
- cloudwatch:ListMetrics
- cloudwatch:GetMetricData

이러한 작업은 MediaTailor 및 CloudWatch에 대한 읽기 액세스 권한을 부여하는 AWS 관리형 정책에 포함됩니다. MediaTailor 권한에 대한 일반 정보는 섹션을 참조하세요 [에 대한 자격 증명 및 액세스 관리 AWS Elemental MediaTailor](#).

AWS 리전 선택

MediaTailor는 재생 구성을 실행하는 각 AWS 리전에 CloudWatch 지표를 게시합니다. 대시보드 상단의 리전 선택기는 페이지 쿼리를 실행하는 리전과 결과를 결합하는 방법을 제어합니다.

대시보드를 처음 열면 MediaTailor는 MediaTailor를 사용할 수 있는 각 AWS 리전을 확인하고 계정에 재생 구성이 하나 이상 있는 리전을 제공합니다. 검사를 완료할 수 없는 리전은 페이지에 기록되며 재

시도할 수 있습니다. 모든 AWS 리전에서 확인이 실패하면 대시보드는 현재 로그인한 AWS 리전으로 돌아갑니다.

대시보드는 리전 선택기 위의 리전 집계/리전 비교 선택기로 선택한 두 가지 보기 모드를 지원합니다.

리전 집계

선택한 모든 리전의 지표를 단일 값 집합으로 결합합니다. 전체 광고 삽입 공간의 합계 및 가중 평균을 보려면이 모드를 사용합니다.

리전 비교

선택한 리전을 그룹 A와 그룹 B라는 두 개의 명명된 그룹으로 분할하고 각 그룹의 값을 나란히 표시합니다. 각 키 수익화 지표에 대해 각 그룹의 타일에는 다른 그룹과의 백분율 차이도 표시됩니다. 예를 들어 기본 리전을 장애 조치 리전과 비교하려면이 모드를 사용합니다.

주요 수익화 지표 및 추세 그래프의 경우 대시보드가 선택한 리전의 값을 결합하는 방법은 지표 유형에 따라 달라집니다.

- 개수 지표의 경우 선택한 각 리전의 값이 합산됩니다.
- 속도 지표의 경우 선택한 각 리전의 분자와 분모가 합산되고 한 번 분할됩니다. 이렇게 하면 가중 평균이 생성되므로 트래픽이 많은 리전이 결과에 비례하여 더 많이 기여합니다.

지표 테이블은 재생 구성 또는 광고 추적 도메인별로 값을 표시하며, 각 테이블은 뷰에 따라 리전 값을 다르게 결합합니다. 자세한 내용은이 페이지의 뒷부분에 있는 테이블 섹션을 참조하세요.

주요 수익 지표

대시보드 상단의 주요 수익 지표 섹션에는 선택한 리전의 모든 재생 구성에서 선택한 시간 범위 동안의 광고 삽입 성능을 요약하는 요약 타일 세트가 표시됩니다. 리전 비교 보기에서 각 타일은 각 리전 그룹의 값을 표시하고 각 그룹의 타일은 다른 그룹과의 백분율 차이를 표시합니다.

대시보드에는 다음과 같은 주요 수익 지표가 표시됩니다. 기본 CloudWatch 지표에 대한 정의는 섹션을 참조하세요 [Amazon CloudWatch 지표 AWS Elemental MediaTailor 를 사용한 모니터링](#).

- 가중 채우기 속도 - $\text{SUM(Avail.FilledDuration)} / \text{SUM(Avail.Duration)} \times 100$
- 광고 삽입 속도 - $\text{SUM(AdsBilled)} / \text{SUM(AdDecisionServer.Ads)} \times 100$
- 광고 노출률 - $\text{SUM(Avail.Impression)} / \text{SUM(AdsBilled)} \times 100$

- 비디오 완성률 - $\text{SUM(Avail.Complete)} / \text{SUM(Avail.Impression)} \times 100$
- 광고 삽입 - SUM(AdsBilled)

추세 그래프

추세 섹션에는 선택한 시간 범위의 주요 수익화 지표가 시계열 선 차트로 표시됩니다. 추세 그래프를 사용하여 각 지표가 시간 경과에 따라 어떻게 움직이는지 확인하고 단일 요약 값이 숨길 수 있는 회귀를 파악합니다.

집계 리전 보기에서 각 차트는 선택한 모든 리전을 합친 한 줄을 표시합니다. 리전 비교 보기에서 각 차트는 리전 그룹당 한 줄을 표시하며, 범례는 그룹 A와 그룹 B를 식별합니다.

구성 테이블별 수익화 성능

구성별 수익화 성능 테이블에는 선택한 시간 범위 동안 선택한 리전의 각 재생 구성에 대한 주요 속도 지표가 표시됩니다. 이 표를 사용하여 다른 구성과 비교하여 성능이 저하된 구성을 찾습니다.

재생 구성은 특정 AWS 리전으로 범위가 지정됩니다. 선택한 두 개 이상의 리전에 동일한 구성 이름이 있는 경우 테이블에 각 리전에 대해 별도의 행이 표시됩니다. 구성 이름 옆의 해당 리전의 해당 구성에 대한 세부 정보 페이지로 연결됩니다.

이 표에는 다음과 같은 열이 있습니다.

- 구성 이름 - 재생 구성의 이름입니다.
- 리전 - 구성이 존재하는 AWS 리전입니다.
- 그룹 - 리전 비교 보기에서만 행의 리전이 속한 리전 그룹(그룹 A 또는 그룹 B)입니다.
- 가중 채우기 속도, 광고 삽입 속도, 광고 노출 속도 및 비디오 완료 속도 -에 설명된 속도 지표로 [주요 수익 지표](#), 선택한 시간 범위 동안 해당 리전의 해당 구성에 대해 계산됩니다.

테이블 위의 속성 필터를 사용하여 표시된 행의 범위를 좁힙니다. 구성 이름 또는 리전을 기준으로 필터링할 수 있습니다. 리전 비교 보기에서 그룹별로 필터링하여 한 번에 하나의 비교 그룹을 표시할 수도 있습니다. 각 속성은 연산자 = (같음), !=를 지원합니다. (같지 않음), : (포함) 및 !: (포함되지 않음).

CloudWatch API 사용을 제한하기 위해 테이블은 각 로드에서 선택한 리전당 최대 10개의 ListMetrics 호출을 수행합니다. 선택한 리전에 이러한 호출이 반환하는 것보다 더 많은 구성이 있는 경우 테이블에 영향을 받는 리전을 나열하는 정보 배너와 더 로드 버튼이 표시됩니다. 추가 로드를 선택하여 해당 리전에서 추가 구성을 검색합니다.

비컨 실행 성능 테이블

Beacon 실행 성능 테이블에는 선택한 시간 범위 동안 선택한 리전의 광고 추적 도메인당 노출 및 전체 비컨에 대한 CloudWatch 실행, 재시도 및 복구 지표가 표시됩니다. 이러한 지표에 대한 정의는 섹션을 참조하세요 [Amazon CloudWatch 지표 AWS Elemental MediaTailor 를 사용한 모니터링](#).

테이블 위의 노출 및 완료 선택기를 사용하여 테이블에 표시되는 비컨 유형을 선택합니다. 선택기는 찾아보기 보기와 순위 보기 모두에 적용됩니다.

테이블은 테이블 위의 찾아보기 및 순위 선택기에서 선택한 두 가지 보기 모드를 지원합니다.

보기 찾아보기

찾아보기 보기에는 선택한 리전에서 관찰된 각 광고 추적 도메인이 나열되며, 해당 리전에서 비컨 수가 합산됩니다. 리전 비교 보기에서 두 그룹에 나타나는 광고 추적 도메인은 그룹당 한 행으로 표시됩니다.

찾아보기 보기에는 다음 열이 표시됩니다.

- 광고 추적 도메인
- 그룹 - 리전 비교 보기에서만.
- 실행, 재시도, 복구 및 재시도 성공률

테이블 위의 속성 필터를 사용하여 표시된 행의 범위를 좁힙니다. 그룹별 리전 비교 보기에서 광고 추적 도메인을 기준으로 필터링할 수 있습니다. 속성 필터는 연산자 = (같음), != (다름)를 지원합니다. (같지 않음), : (포함) 및 !: (포함되지 않음).

CloudWatch API 사용을 제한하기 위해 찾아보기 보기는 각 로드에서 선택한 리전당 최대 10개의 ListMetrics 호출을 수행합니다. 선택한 리전에 이러한 호출이 반환하는 것보다 많은 광고 추적 도메인이 있는 경우 테이블에 영향을 받는 리전을 나열하는 정보 배너와 더 로드 버튼이 표시됩니다. 추가 로드를 선택하여 해당 리전에서 추가 도메인을 검색합니다.

순위 보기

순위 보기에는 선택한 비컨 수 지표의 상단 또는 하단 광고 추적 도메인이 나열되며, (도메인, 리전) 페어당 하나의 행이 있습니다. 리전 비교 보기에서 순위는 그룹별로 독립적으로 실행되므로 각 그룹이 자체 상위 또는 하위 도메인에 기여합니다.

테이블 위의 두 컨트롤을 사용하여 순위를 구성합니다.

- 지표 - 순위를 매길 비컨 수 지표: 실행됨, 재시도됨 또는 복구됨. 재시도 성공률은 계산된 비율이며 행의 순위를 매기는 데 사용할 수 없습니다.
- 순서 - 오름차순 또는 내림차순.

순위 보기에는 찾아보기 보기와 동일한 열과 시작 부분의 순위 열 및 각(도메인, 리전) 행이 속한 리전을 식별하는 리전 열이 표시됩니다. 광고 추적 도메인, 리전 및 (리전 비교 보기에서) 그룹을 기준으로 필터링할 수 있습니다.

순위 보기는 CloudWatch Metrics Insights를 사용하여 기본 비컨 지표를 정렬합니다. Metrics Insights는 지난 14일간의 데이터를 읽은 쿼리만 지원합니다. 선택한 시간 범위가 14일을 초과하면 테이블에 최근 14일의 데이터가 표시됩니다.

대시보드 데이터 내보내기

지표 테이블 및 추세 그래프의 데이터를 CSV 파일로 내보낼 수 있습니다.

지표 테이블에서 데이터를 내보내려면

1. 포함하려는 행 옆의 확인란을 선택합니다. 테이블의 페이지 사이를 이동하고 행을 계속 선택할 수 있습니다.
2. 테이블 위에서 CSV 내보내기를 선택합니다.

내보낸 파일에는 테이블에 표시된 열당 하나의 열이 포함됩니다. 내보낸 파일에 지표 값을 표시하려면 대시보드가 행의 지표 값을 로드하도록 선택한 각 행이 포함된 테이블의 페이지를 확인해야 합니다.

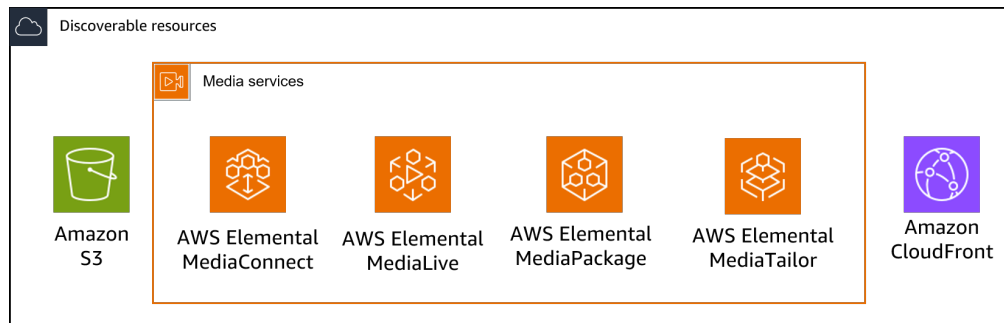
추세 그래프에서 데이터를 내보내려면 그래프에서 작업 메뉴를 선택한 다음 .csv로 다운로드를 선택합니다.

워크플로 모니터를 사용하여 AWS 미디어 서비스 모니터링

워크플로 모니터는 AWS 미디어 워크플로의 검색, 시각화 및 모니터링을 위한 도구입니다. 워크플로 모니터는 AWS 콘솔 및 API에서 사용할 수 있습니다. 워크플로 모니터를 사용하여 워크플로 리소스의 시각적 매핑인 시그널 맵을 검색하고 생성할 수 있습니다. 매핑된 리소스를 모니터링하기 위해 Amazon CloudWatch 경보와 Amazon EventBridge 규칙 템플릿을 생성하고 관리할 수 있습니다. 생성한 모니터링 템플릿은 반복성을 허용하기 위해 배포 가능한 AWS CloudFormation 템플릿으로 변환됩니다. AWS 권장 경보 템플릿은 사전 정의된 모범 사례 모니터링을 제공합니다.

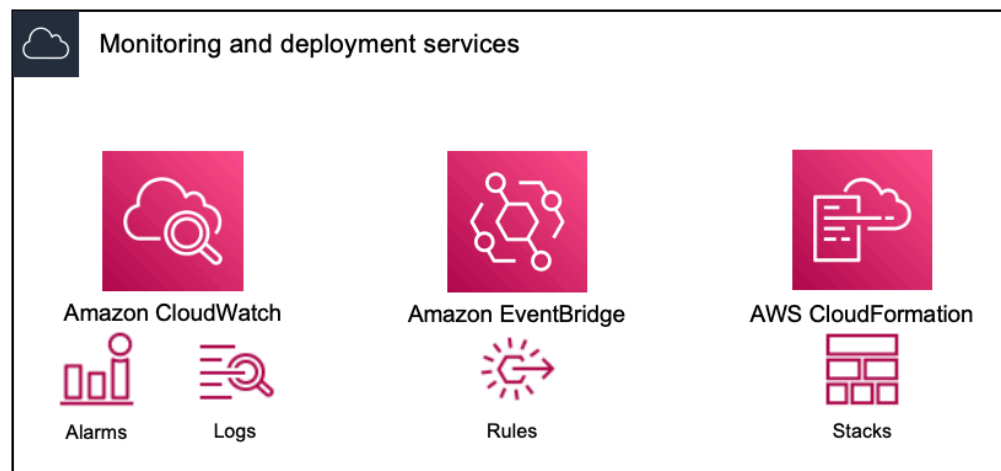
검색

시그널 맵을 활용하여 미디어 워크플로와 연결된 상호 연결된 AWS 리소스를 자동으로 검색합니다. 지원되는 모든 서비스 리소스에서 검색을 시작할 수 있으며 워크플로의 전체 매핑을 생성합니다. 시그널 맵은 독립 실행형 시각화 도구로 사용하거나 모니터링 템플릿과 결합하여 보강할 수 있습니다.



모니터링

사용자 지정 CloudWatch 경보 및 EventBridge 규칙 템플릿을 생성하여 미디어 워크플로의 상태를 모니터링할 수 있습니다. 모범 사례 경보 템플릿은 워크플로 모니터 환경으로 가져올 수 있습니다. 모범 사례 경보 템플릿을 그대로 사용하거나 워크플로에 맞게 편집할 수 있습니다. 생성한 템플릿은 반복적으로 배포할 수 있도록 CloudFormation 템플릿으로 변환됩니다.



Note

워크플로 모니터를 사용하는 데 드는 직접적인 비용은 없습니다. 그러나 워크플로를 모니터링하기 위해 생성하고 사용하는 리소스와 관련된 비용이 발생합니다.

모니터링이 배포되면 Amazon CloudWatch 및 Amazon EventBridge 리소스가 생성됩니다.

AWS Management Console을 사용할 때 모니터링을 시그널 맵에 배포하기 전에 생성되는 리소스 수에 대한 알림을 받게 됩니다. 요금에 대한 자세한 내용은 [CloudWatch 요금](#) 및 [EventBridge 요금](#) 단원을 참조하세요.

워크플로 모니터는 AWS CloudFormation 템플릿을 사용하여 CloudWatch 및 EventBridge 리소스를 배포합니다. 이러한 템플릿은 배포 과정 중 워크플로 모니터를 통해 생성되는 표준 클래스 Amazon Simple Storage Service 버킷에 저장되며 객체 스토리지 및 리콜 요금이 부과됩니다. 요금에 대한 자세한 내용은 [Amazon S3 요금](#)을 참조하세요.

AWS Elemental MediaPackage 채널에 대한 워크플로 모니터 시그널 맵에서 생성된 미리 보기는 MediaPackage 오리진 엔드포인트에서 전송되며 데이터 전송 요금이 발생합니다. 요금은 [MediaPackage 요금](#) 단원을 참조하세요.

워크플로 모니터 구성 요소

워크플로 모니터의 주요 구성 요소는 4가지입니다.

- CloudWatch 경보 템플릿 - CloudWatch를 사용하여 모니터링할 조건을 정의합니다. 자체 경보 템플릿을 생성하거나에서 생성한 사전 정의된 템플릿을 가져올 수 있습니다 AWS. 자세한 내용은 다음을 참조하세요. [AWS 미디어 워크플로를 모니터링하기 위한 CloudWatch 경보 그룹 및 템플릿](#)
- EventBridge 규칙 템플릿 - 경보가 트리거될 때 EventBridge에서 알림을 보내는 방법을 정의합니다. 자세한 내용은 다음을 참조하세요. [AWS 미디어 워크플로를 모니터링하기 위한 EventBridge 규칙 그룹 및 템플릿](#)
- 시그널 맵 - 자동화된 프로세스를 사용하여 기존 AWS 리소스를 사용하여 AWS Elemental 워크플로 맵을 생성합니다. 시그널 맵을 사용하면 워크플로의 리소스를 검색하고 해당 리소스에 모니터링을 배포할 수 있습니다. 자세한 내용은 [워크플로 모니터 시그널 맵](#) 단원을 참조하세요.
- 개요 - 개요 페이지에서는 여러 시그널 맵의 상태를 한 위치에서 직접 모니터링할 수 있습니다. 워크플로의 지표, 로그 및 경보를 검토할 수 있습니다. 자세한 내용은 [워크플로 모니터 개요](#) 단원을 참조하세요.

지원되는 서비스

워크플로 모니터는 다음 서비스와 연관된 리소스의 자동 검색과 시그널 매핑을 지원합니다.

- AWS Elemental MediaConnect
- AWS Elemental MediaLive
- AWS Elemental MediaPackage
- AWS Elemental MediaTailor
- Amazon S3
- Amazon CloudFront

주제

- [AWS 미디어 서비스를 모니터링하도록 워크플로 모니터 구성](#)
- [워크플로 모니터 사용](#)

AWS 미디어 서비스를 모니터링하도록 워크플로 모니터 구성

워크플로 모니터를 처음으로 설정하려면 경보 및 이벤트 템플릿을 생성하고 미디어 워크플로를 모니터링하는 데 사용되는 시그널 맵을 검색하세요. 다음 가이드에서는 관리자 및 운영자 수준의 IAM 역할을 모두 설정하고, 워크플로 모니터 리소스를 생성하고, 워크플로에 모니터링을 배포하는 데 필요한 단계를 다룹니다.

주제

- [워크플로 모니터 시작하기](#)
- [워크플로 모니터 그룹 및 템플릿](#)
- [워크플로 모니터 시그널 맵](#)
- [워크플로 모니터 할당량](#)

워크플로 모니터 시작하기

다음 단계에서는 워크플로 모니터를 처음 사용할 때 알아야 할 기본적인 개요를 제공합니다.

1. 관리자 및 운영자 수준 역할에 대한 워크플로 모니터 IAM 권한 설정: [워크플로 모니터링 IAM 정책](#)
2. 경보 템플릿을 빌드하거나 AWS다음에서 생성한 사전 정의된 템플릿을 가져옵니다. [CloudWatch 경보](#)
3. EventBridge에서 전달할 알림 이벤트 빌드: [EventBridge 규칙](#)
4. 기존 AWS Elemental 리소스를 사용하여 시그널 맵을 검색합니다. [시그널 맵](#)
5. 시그널 맵에 경보 템플릿 및 알림 규칙 연결: [템플릿 연결](#)
6. 템플릿을 배포하여 시그널 맵 모니터링 시작: [모니터링 템플릿 배포](#)
7. AWS 콘솔의 개요 섹션을 사용하여 워크플로 모니터 리소스 모니터링 및 검토: [개요](#)



워크플로 모니터링 IAM 정책

워크플로 모니터는 여러 AWS 서비스와 상호 작용하여 시그널 맵을 생성하고 CloudWatch 및 EventBridge 리소스와 CloudFormation 템플릿을 빌드합니다. 워크플로 모니터는 다양한 서비스와 상호 작용하므로 이러한 서비스에 대해 특정 AWS Identity and Access Management (IAM) 정책을 할당해야 합니다. 다음 예시는 관리자 및 운영자 IAM 역할 모두에 필요한 IAM 정책을 보여줍니다.

관리자 IAM 정책

다음은 관리자 수준의 워크플로 모니터 IAM 정책에 대한 예시 정책입니다. 이 역할을 통해 워크플로 모니터 리소스 및 워크플로 모니터와 상호 작용하는 지원 서비스 리소스를 생성하고 관리할 수 있습니다.

JSON

```
{

  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:List*",
        "cloudwatch:Describe*",
        "cloudwatch:Get*",
        "cloudwatch:PutAnomalyDetector",
        "cloudwatch:PutMetricData",
        "cloudwatch:PutMetricAlarm",
        "cloudwatch:PutCompositeAlarm",
        "cloudwatch:PutDashboard",
        "cloudwatch>DeleteAlarms",
        "cloudwatch>DeleteAnomalyDetector",
        "cloudwatch>DeleteDashboards",
        "cloudwatch:TagResource",
        "cloudwatch:UntagResource"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:List*",
        "cloudformation:Describe*",
```

```

        "cloudformation:CreateStack",
        "cloudformation:UpdateStack",
        "cloudformation>DeleteStack",
        "cloudformation:TagResource",
        "cloudformation:UntagResource"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "cloudfront:List*",
        "cloudfront:Get*"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:DescribeNetworkInterfaces"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "events:List*",
        "events:Describe*",
        "events:CreateEventBus",
        "events:PutRule",
        "events:PutTargets",
        "events:EnableRule",
        "events:DisableRule",
        "events>DeleteRule",
        "events:RemoveTargets",
        "events:TagResource",
        "events:UntagResource"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "logs:Describe*",

```

```
        "logs:Get*",
        "logs:TagLogGroup",
        "logs:TagResource",
        "logs:UntagLogGroup",
        "logs:UntagResource"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "mediaconnect:List*",
        "mediaconnect:Describe*"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "medialive:*"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "mediapackage:List*",
        "mediapackage:Describe*"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "mediapackagev2:List*",
        "mediapackagev2:Get*"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "mediapackage-vod:List*",
        "mediapackage-vod:Describe*"
    ]
}
```



```

    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "mediatailor:List*",
      "mediatailor:Describe*",
      "mediatailor:Get*"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "resource-groups:ListGroup",
      "resource-groups:GetGroup",
      "resource-groups:GetTags",
      "resource-groups:GetGroupQuery",
      "resource-groups:GetGroupConfiguration",
      "resource-groups:CreateGroup",
      "resource-groups:UngroupResources",
      "resource-groups:GroupResources",
      "resource-groups>DeleteGroup",
      "resource-groups:UpdateGroupQuery",
      "resource-groups:UpdateGroup",
      "resource-groups:Tag",
      "resource-groups:Untag"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:*"
    ],
    "Resource": "arn:aws:s3:::workflow-monitor-templates*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "sns:TagResource",
      "sns:UntagResource"
    ],
    "Resource": "*"
  }
]

```

```

    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "tag:Get*",
      "tag:Describe*",
      "tag:TagResources",
      "tag:UntagResources"
    ],
    "Resource": "*"
  }
]
}

```

운영자 IAM 정책

다음은 운영자 수준의 워크플로 모니터 IAM 정책에 대한 예시 정책입니다. 이 역할을 통해 워크플로 모니터 리소스 및 워크플로 모니터와 상호 작용하는 지원 서비스 리소스에 대해 제한된 액세스 권한과 읽기 전용 액세스 권한을 부여받을 수 있습니다.

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": "*"
  }
]
}

```

워크플로 모니터 그룹 및 템플릿

워크플로 모니터링을 시그널 맵에 배포하려면 먼저 CloudWatch 경보 및 EventBridge 알림을 위한 그룹과 템플릿을 생성해야 합니다. CloudWatch 템플릿은 경보를 트리거하는 데 사용할 시나리오와 임계값을 정의합니다. EventBridge 템플릿은 이러한 경보가 사용자에게 보고되는 방식을 결정합니다.

연결된 리소스의 매핑만 사용하고 워크플로 모니터의 모니터링 템플릿 기능은 사용하지 않으려는 경우 CloudWatch 및 EventBridge 템플릿 없이 시그널 맵을 사용할 수 있습니다. 시그널 맵 사용에 대한 자세한 내용은 [시그널 맵](#) 단원을 참조하세요.

주제

- [AWS 미디어 워크플로를 모니터링하기 위한 CloudWatch 경보 그룹 및 템플릿](#)
- [AWS 미디어 워크플로를 모니터링하기 위한 EventBridge 규칙 그룹 및 템플릿](#)

AWS 미디어 워크플로를 모니터링하기 위한 CloudWatch 경보 그룹 및 템플릿

워크플로 모니터 경보를 통해 기존의 CloudWatch 지표를 기반으로 시그널 맵에 대한 경보를 설정할 수 있습니다. 경보 템플릿 그룹을 생성하여 워크플로에 중요한 경보 유형을 정렬하고 분류할 수 있습니다. 각 경보 템플릿 그룹 내에 모니터링하려는 특정 CloudWatch 지표와 파라미터가 포함된 경보 템플릿을 생성합니다. 자체 경보 템플릿을 생성하거나에서 생성한 권장 경보 템플릿을 가져올 수 있습니다 AWS. 경보 템플릿 그룹을 생성하고 해당 그룹 내에 경보 템플릿을 생성한 후에는 이러한 경보 템플릿 그룹 중 하나 이상을 시그널 맵에 연결할 수 있습니다.

먼저 경보 템플릿 그룹을 생성해야 합니다. 경보 템플릿 그룹을 생성한 후 자체 템플릿을 생성하거나에서 생성한 권장 템플릿을 사용할 수 있습니다 AWS. 자체 경보 템플릿을 생성하려면 이 페이지를 계속 진행합니다. 권장 템플릿 가져오기에 대한 자세한 내용은 [권장 템플릿](#) 단원을 참조하세요.

이 단원에서는 워크플로 모니터를 사용하여 CloudWatch 경보를 생성하는 방법에 대해 다룹니다. CloudWatch 서비스에서 경보를 처리하는 방식 및 경보 구성 요소에 대한 자세한 내용은 Amazon CloudWatch 사용 설명서의 [CloudWatch 경보 사용](#)을 참조하세요.

경보 템플릿 그룹 생성

경보 템플릿 그룹을 사용하면 워크플로에 중요한 경보 유형을 정렬하고 분류할 수 있습니다.

경보 템플릿 그룹을 생성하려면

1. 워크플로 모니터 콘솔의 탐색 창에서 CloudWatch 경보 템플릿을 선택합니다.
2. 경보 템플릿 그룹 생성을 선택합니다.
3. 경보 템플릿 그룹에 고유한 그룹 이름과 설명(선택 사항)을 지정합니다.
4. 생성을 선택하면 새로 생성된 경보 템플릿 그룹의 세부 정보 페이지로 이동합니다.

경보 템플릿 생성

모니터링하려는 CloudWatch 지표와 파라미터가 포함된 경보 템플릿을 생성할 수 있습니다.

경보 템플릿을 생성하려면

1. 경보 템플릿 그룹의 세부 정보 페이지에서 경보 템플릿 생성을 선택합니다.
2. 경보 템플릿에 고유한 템플릿 이름과 설명(선택 사항)을 지정합니다.
3. 지표 선택 섹션에서 다음을 수행합니다.
 1. 대상 리소스 유형을 선택합니다. 대상 리소스 유형은 각 서비스에 대한 리소스(예: MediaLive 및 MediaPackage의 채널 또는 MediaConnect의 흐름)입니다.
 2. 지표 이름을 선택합니다. 이는 경보의 기반이 될 CloudWatch 지표입니다. 지표 목록은 선택한 대상 리소스 유형에 따라 달라집니다.
4. 경보 설정 섹션에서 다음을 수행합니다.

Note

CloudWatch 서비스에서 경보를 처리하는 방식 및 경보 구성 요소에 대한 자세한 내용은 Amazon CloudWatch 사용 설명서의 [CloudWatch 경보 사용](#)을 참조하세요.

1. 통계를 선택합니다. 이는 지표를 모니터링하는 데 사용할 합계 또는 평균 등의 값입니다.
 2. 비교 연산자를 선택합니다. 이 필드는 다음 단계에서 설정할 임계값을 참조합니다.
 3. 임계값을 설정합니다. 이는 비교 연산자가 보다 큼, 보다 작음, 같음 상태를 결정하는 데 사용하는 숫자 값입니다.
 4. 기간을 설정합니다. 이는 초 단위의 시간 값입니다. 기간은 통계, 비교 연산자, 임계값이 상호 작용하여 경보 트리거 여부를 결정하는 시간입니다.
 5. 데이터 포인트를 설정합니다. 이 값에 따라 경보 트리거에 필요한 데이터 포인트 수가 결정됩니다.
 6. 누락 데이터 처리 방법을 선택합니다. 이 선택 항목에 따라 해당 경보가 누락된 데이터에 반응하는 방식이 결정됩니다.
5. 생성을 선택하여 프로세스를 완료합니다.

완성된 경보 템플릿의 예시에는 다음과 같은 파라미터가 포함될 수 있습니다. MediaConnect 흐름의 대상 리소스 유형은 연결 끊김 지표 이름을 모니터링합니다. 통계 값은 합계로, 비교 연산자는 '크거나 같음'으로, 임계값은 10으로 설정되어 있습니다. 기간은 60초로 설정되어 있으며 데이터 포인트 1개 중 1개만 필요합니다. 누락 데이터 처리는 '무시'로 설정되어 있습니다.

이러한 설정의 결과로 워크플로 모니터는 흐름의 연결이 끊기는지 모니터링합니다. 60초 내에 연결이 10회 이상 끊기면 경보가 트리거됩니다. 60초 내에 10회 이상의 연결 끊김이 한 번이라도 발생하면 경보가 트리거됩니다.

AWS 미디어 워크플로를 모니터링하기 위한 권장 경보 템플릿

워크플로 모니터의 권장 템플릿은 지표에 적합한 사전 정의된 경보 설정이 있는 AWS 업선된 Elemental 서비스 지표입니다. 사용자 지정 경보 템플릿을 생성하지 않으려는 경우 권장 템플릿에서 생성한 모범 사례 모니터링 템플릿을 제공합니다 AWS.

워크플로 모니터에는 각 지원 서비스에 권장되는 템플릿 그룹이 포함되어 있습니다. 이러한 그룹은 특정 유형의 워크플로에 모범 사례 모니터링을 적용하도록 설계되었습니다. 각 템플릿 그룹에는 서비스 별 지표에서 구성된 업선된 경보가 포함되어 있습니다. 예를 들어 MediaLive 멀티플렉스 워크플로에

권장되는 템플릿 그룹에는 MediaConnect CDI 워크플로와 다른 사전 구성된 지표 세트가 포함되어 있습니다.

권장 경고 템플릿을 사용하려면

1. 단계에 따라 [경보 템플릿 그룹을 생성](#)하거나 기존 그룹을 선택합니다.
2. 경고 템플릿 섹션에서 가져오기를 선택합니다. AWS 권장 템플릿을 템플릿 그룹으로 가져와야 합니다.
3. CloudWatch 경고 템플릿 그룹 드롭다운을 사용하여 AWS 권장 그룹을 선택합니다. 이 그룹에는 특정 서비스에 대해 선별된 경보가 포함되어 있습니다.
4. 확인란을 사용하여 가져올 템플릿을 선택합니다. 각 템플릿에는 지표와 사전 구성된 모니터링 값이 나열되고 지표에 대한 설명이 제공됩니다. 템플릿 선택을 마쳤으면 추가 버튼을 선택합니다.
5. 템플릿을 선택하면 가져올 경고 템플릿 섹션으로 이동합니다. 선택 사항을 검토하고 가져오기를 선택합니다.
6. 가져오기가 완료되면 선택한 템플릿이 템플릿 그룹에 추가됩니다. 템플릿을 더 추가하려면 가져오기 프로세스를 반복합니다.
7. 가져오기 후 가져온 템플릿을 사용자 지정할 수 있습니다. 경고 설정은 경고 요구 사항에 맞게 수정할 수 있습니다.

AWS 미디어 워크플로를 모니터링하기 위한 EventBridge 규칙 그룹 및 템플릿

CloudWatch는 Amazon EventBridge 규칙을 사용하여 알림을 보냅니다. 먼저 이벤트 템플릿 그룹을 생성합니다. 이후 해당 이벤트 템플릿 그룹에 알림을 생성하는 조건과 알림을 받는 사람을 결정하는 이벤트 템플릿을 생성합니다.

이 단원에서는 워크플로 모니터를 사용하여 EventBridge 규칙을 생성하는 방법에 대해 다룹니다. EventBridge 서비스에서 규칙을 사용하는 방식에 대한 자세한 내용은 Amazon EventBridge 사용 설명서의 [EventBridge 규칙](#)을 참조하세요.

이벤트 템플릿 그룹 생성

이벤트 템플릿 그룹을 사용하면 사용 사례에 따라 이벤트를 정렬하고 분류할 수 있습니다.

이벤트 템플릿 그룹을 생성하려면

1. 워크플로 모니터 콘솔의 탐색 창에서 EventBridge 규칙 템플릿을 선택합니다.
2. 이벤트 템플릿 그룹 생성을 선택합니다.

3. 경보 템플릿 그룹에 고유한 그룹 이름과 설명(선택 사항)을 지정합니다.
4. 생성을 선택하면 새로 생성된 경보 템플릿 그룹의 세부 정보 페이지로 이동합니다.

이벤트 템플릿 생성

생성한 이벤트 템플릿에 따라 알림을 보낼 수 있습니다.

이벤트 템플릿을 생성하려면

1. 이벤트 템플릿 그룹의 세부 정보 페이지에서 이벤트 템플릿 생성을 선택합니다.
2. 이벤트 템플릿에 고유한 템플릿 이름과 설명(선택 사항)을 지정합니다.
3. 규칙 설정 섹션에서 다음을 수행합니다.
 1. 이벤트 유형을 선택합니다. 이벤트 유형을 선택할 때에서 생성한 여러 이벤트 중에서 선택하거나 신호 맵 활성 경보를 AWS 선택하여 경보 템플릿에서 생성한 경보를 사용할 수 있습니다.
 2. 대상 서비스를 선택합니다. 이 서비스에 따라 이 이벤트에 대한 알림 수신 방법이 결정됩니다. Amazon Simple Notification Service 또는 CloudWatch 로그를 선택할 수 있습니다.
 3. 대상 서비스를 선택한 후 대상을 선택합니다. 선택한 대상 서비스에 따라 Amazon SNS 주제 또는 CloudWatch 로그 그룹이 됩니다.
4. 생성을 선택하여 프로세스를 완료합니다.

워크플로 모니터 시그널 맵

시그널 맵은 미디어 워크플로에 있는 AWS 리소스의 시각적 매핑입니다. 워크플로 모니터를 사용하여 지원되는 리소스 유형에서 시그널 맵 검색을 시작할 수 있습니다. 검색 프로세스 중에 워크플로 모니터는 연결된 모든 AWS 리소스를 자동으로 재귀적으로 매핑합니다. 시그널 맵이 생성된 후에는 워크플로 모니터 콘솔을 사용하여 모니터링 템플릿 배포, 지표 보기, 매핑된 리소스의 세부 정보 보기 등의 작업을 수행할 수 있습니다.

주제

- [AWS 미디어 워크플로에 대한 시그널 맵 생성](#)
- [AWS 미디어 워크플로의 시그널 맵 보기](#)
- [AWS 미디어 워크플로의 시그널 맵에 경보 및 이벤트 템플릿 연결](#)
- [AWS 미디어 워크플로의 시그널 맵에 템플릿 배포](#)
- [AWS 미디어 워크플로의 시그널 맵 업데이트](#)

• [AWS 미디어 워크플로의 시그널 맵 삭제](#)

AWS 미디어 워크플로에 대한 시그널 맵 생성

워크플로 모니터 시그널 맵을 사용하여 미디어 워크플로에 연결된 모든 AWS 리소스의 시각적 매핑을 생성할 수 있습니다.

시그널 맵을 생성하려면

1. 워크플로 모니터 콘솔의 탐색 창에서 시그널 맵을 선택합니다.
2. 시그널 맵 생성을 선택합니다.
3. 시그널 맵에 이름과 설명을 지정합니다.
4. 새 시그널 맵 검색 섹션에 현재 계정과 선택한 리전의 리소스가 표시됩니다. 리소스를 선택하여 시그널 맵 검색을 시작합니다. 선택한 리소스가 검색의 시작점이 됩니다.
5. 생성을 선택합니다. 검색 프로세스가 완료될 때까지 잠시 기다리세요. 프로세스가 완료되면 새 시그널 맵이 표시됩니다.

Note

AWS Elemental MediaPackage 채널에 대한 워크플로 모니터 시그널 맵에서 생성된 미리 보기는 MediaPackage 오리진 엔드포인트에서 전송되며 데이터 전송 요금이 발생합니다. 요금은 [MediaPackage 요금](#) 단원을 참조하세요.

AWS 미디어 워크플로의 시그널 맵 보기

워크플로 모니터 시그널 맵을 사용하면 미디어 워크플로에서 연결된 모든 AWS 리소스의 시각적 매핑을 볼 수 있습니다.

시그널 맵 보기

시그널 맵을 선택하면 시그널 맵을 모니터링하거나 구성하는 데 사용할 수 있는 두 가지 보기가 표시됩니다. 시그널 맵 모니터링 및 시그널 맵 구성 버튼은 시그널 맵 콘솔 섹션의 오른쪽 상단에 있으며 상황에 따라 달라지는 버튼입니다.

탐색 창의 시그널 맵 섹션을 사용하여 시그널 맵을 선택하면 시그널 맵이 구성 보기에 표시됩니다. 구성 보기에서는 이 시그널 맵에 연결된 템플릿 그룹을 변경하고, 연결된 템플릿을 배포하고, 시그널 맵의 기본 세부 정보와 태그를 볼 수 있습니다.

탐색 창 의 개요 섹션을 사용하여 시그널 맵을 선택하면 시그널 맵이 모니터링 보기에 표시됩니다. 모니터링 보기에는 이 시그널 맵에 대한 CloudWatch 경보, EventBridge 규칙, 알림, 로그 및 지표가 표시됩니다.

오른쪽 상단의 시그널 맵 모니터링/구성 버튼을 선택하여 언제든지 보기를 변경할 수 있습니다. 구성 보기에는 관리자 수준의 IAM 권한이 필요합니다. 필요한 IAM 권한은 [워크플로 모니터링 IAM 정책](#)에서 확인할 수 있습니다.

시그널 맵 탐색

시그널 맵에는 워크플로 모니터에서 검색된 지원되는 모든 AWS 리소스에 대한 노드가 포함됩니다. 썸네일 미리 보기를 사용할 수 있는 경우 MediaLive 채널 및 MediaPackage 엔드포인트와 같은 특정 리소스에서 콘텐츠의 썸네일 미리 보기를 표시할 수 있습니다.

리소스 노드를 선택하고 작업 드롭다운 메뉴에서 선택한 리소스 세부 정보 보기를 선택하면 관련 서비스의 세부 정보 페이지로 이동합니다. 예를 들어 MediaLive 채널을 선택하고 선택한 리소스 세부 정보 보기를 선택하면 해당 채널에 대한 MediaLive 콘솔의 세부 정보 페이지가 열립니다.

리소스 노드를 선택하면 활성 경보 목록이 해당 노드로만 필터링됩니다. 활성 경보에서 리소스의 대상 ARN을 선택하면 선택한 리소스가 열린 상태로 관련 서비스의 세부 정보 페이지로 이동합니다.

AWS 미디어 워크플로의 시그널 맵에 경보 및 이벤트 템플릿 연결

경보 및 이벤트 템플릿을 생성한 후에는 이 템플릿을 시그널 맵에 연결해야 합니다. 생성한 모든 경보 및 이벤트 템플릿은 검색된 모든 시그널 맵에 연결할 수 있습니다.

시그널 맵에 경보 및 이벤트 템플릿을 연결하려면

1. 워크플로 모니터 콘솔의 탐색 창에서 시그널 맵을 선택하고 작업할 시그널 맵을 선택합니다.
2. 시그널 맵 페이지의 오른쪽 상단에 있는 CloudWatch 경보 템플릿 그룹 탭에서 CloudWatch 경보 템플릿 그룹 연결을 선택합니다.
 1. 새 섹션이 열리면 이 시그널 맵에 적용할 경보 템플릿 그룹을 모두 선택한 다음 추가를 선택합니다. 그러면 선택한 경보 템플릿 그룹이 연결된 CloudWatch 경보 템플릿 그룹 섹션으로 이동합니다.
 2. 저장을 선택하면 변경 사항이 저장되고 시그널 맵 페이지로 돌아갑니다.
3. 시그널 맵 페이지 오른쪽에서 EventBridge 규칙 템플릿 그룹 탭을 선택한 다음 EventBridge 규칙 템플릿 그룹 연결을 선택합니다.

1. 새 섹션이 열리면 이 시그널 맵에 적용할 이벤트 템플릿 그룹을 모두 선택한 다음 추가를 선택합니다. 그러면 선택한 규칙 템플릿 그룹이 연결된 EventBridge 규칙 템플릿 그룹 섹션으로 이동합니다.
2. 저장을 선택하면 변경 사항이 저장되고 시그널 맵 페이지로 돌아갑니다.
4. CloudWatch 경보 및 EventBridge 규칙 템플릿을 시그널 맵에 할당했지만 모니터링이 아직 배포되지 않았습니다. 다음 단원에서는 모니터링 리소스 배포에 대해 설명합니다.

AWS 미디어 워크플로의 시그널 맵에 템플릿 배포

경보 및 이벤트 템플릿을 시그널 맵에 연결한 후에는 모니터링을 배포해야 합니다. 배포가 완료될 때까지 시그널 맵 모니터링은 활성화되지 않습니다.

워크플로 모니터는 선택한 시그널 맵과 관련된 경보만 배포합니다. 예를 들어 연결된 경보 템플릿 그룹에는 MediaLive, MediaPackage, MediaConnect 등의 여러 서비스에 대한 경보가 포함될 수 있습니다. 선택한 시그널 맵에 MediaLive 리소스만 포함된 경우 MediaPackage 또는 MediaConnect 경보는 배포되지 않습니다.

모니터링 템플릿을 배포하려면

1. 경보 및 이벤트 템플릿 그룹을 시그널 맵에 연결하고 변경 사항을 저장한 후 작업 드롭다운 메뉴에서 모니터 배포를 선택합니다.
2. 배포를 확인하라는 메시지가 표시되고 생성될 CloudWatch 및 EventBridge 리소스 수가 표시됩니다. 계속하려면 배포를 선택합니다.

Note

워크플로 모니터를 사용하는 데 드는 직접적인 비용은 없습니다. 그러나 워크플로를 모니터링하기 위해 생성하고 사용하는 리소스와 관련된 비용이 발생합니다.

모니터링이 배포되면 Amazon CloudWatch 및 Amazon EventBridge 리소스가 생성됩니다. AWS Management Console을 사용할 때 모니터링을 시그널 맵에 배포하기 전에 생성되는 리소스 수에 대한 알림을 받게 됩니다. 요금에 대한 자세한 내용은 [CloudWatch 요금](#) 및 [EventBridge 요금](#) 단원을 참조하세요.

워크플로 모니터는 AWS CloudFormation 템플릿을 사용하여 CloudWatch 및 EventBridge 리소스를 배포합니다. 이러한 템플릿은 배포 과정 중 워크플로 모니터를 통해 생성되는 표준 클래스 Amazon Simple Storage Service 버킷에 저장되며 객체 스토리지 및 리콜 요금이 부과됩니다. 요금에 대한 자세한 내용은 [Amazon S3 요금](#)을 참조하세요.

3. 배포 상태가 시그널 맵의 이름 옆에 표시됩니다. 배포 상태는 CloudFormation 콘솔의 스택 섹션에도 표시됩니다. 잠시 후 리소스가 생성되고 배포되면 시그널 맵 모니터링이 시작됩니다.

AWS 미디어 워크플로의 시그널 맵 업데이트

워크플로가 변경되면 시그널 맵을 다시 검색하고 모니터링 리소스를 다시 배포해야 할 수 있습니다. 워크플로 모니터는 워크플로 변경 기능이 없는 시각화 및 모니터링 도구입니다. 시그널 맵은 워크플로의 특정 시점을 시각화한 것입니다. 미디어 워크플로의 일부를 추가, 제거하거나 크게 수정한 경우 시그널 맵을 다시 검색하는 것이 좋습니다. 모니터링 리소스가 시그널 맵에 연결된 경우 재검색 프로세스 후 모니터링을 재배포하는 것이 좋습니다.

시그널 맵을 다시 검색하려면

1. 워크플로 모니터 콘솔의 탐색 창에서 시그널 맵을 선택하고 작업할 시그널 맵을 선택합니다.
2. 시그널 맵 구성 보기에 있는지 확인합니다. 보기 변경에 대한 자세한 내용은 [시그널 맵 보기](#) 단원을 참조하세요.
3. 시그널 맵 페이지의 오른쪽 상단에서 작업 드롭다운 메뉴를 선택합니다. 재검색을 선택합니다.
4. 재검색 화면이 표시됩니다. 다시 검색하려는 워크플로에 속한 리소스를 선택합니다. 재검색 버튼을 선택합니다.
5. 현재 워크플로에 따라 시그널 맵이 다시 빌드됩니다. 모니터링 리소스를 재배포해야 하는 경우 이 시그널 맵 페이지에서 작업을 계속 진행합니다. 이전에 연결한 모니터링 템플릿은 계속 연결되어 있지만 재배포가 필요합니다.

시그널 맵 재검색 후 모니터링 템플릿을 재배포하려면

1. 재검색 후에는 업데이트된 시그널 맵으로 이동합니다. 모니터링 템플릿을 재배포하려면 작업 드롭다운 메뉴에서 모니터 배포를 선택합니다.
2. 배포를 확인하라는 메시지가 표시되고 생성될 CloudWatch 및 EventBridge 리소스 수가 표시됩니다. 계속하려면 배포를 선택합니다.
3. 배포 상태가 시그널 맵의 이름 옆에 표시됩니다. 잠시 후 리소스가 생성되고 배포되면 시그널 맵 모니터링이 시작됩니다.

AWS 미디어 워크플로의 시그널 맵 삭제

시그널 맵이 더 이상 필요하지 않은 경우 삭제할 수 있습니다. 시그널 맵에 배포된 모니터링 템플릿이 있는 경우 삭제 과정에서 이 시그널 맵에 배포된 CloudWatch 및 EventBridge 리소스를 삭제할 것인지

문는 메시지가 표시됩니다. 배포된 리소스를 삭제해도 해당 리소스를 생성한 템플릿에는 영향을 주지 않습니다. 이 리소스 삭제는 사용되지 않는 CloudWatch 및 EventBridge 리소스가 배포된 채로 남아 있지 않도록 하기 위한 것입니다.

시그널 맵을 삭제하려면

1. 워크플로 모니터 콘솔의 탐색 창에서 시그널 맵을 선택하고 삭제하려는 시그널 맵 옆의 라디오 버튼을 선택합니다.
2. 삭제 버튼을 선택합니다. 모니터링 리소스 삭제를 확인하라는 메시지가 표시됩니다. 삭제를 선택하여 모니터링 리소스 삭제 프로세스를 시작합니다.
3. 모니터 배포 옆에 현재 상태가 표시됩니다. 상태가 DELETE_COMPLETE로 변경되면 삭제 버튼을 다시 선택합니다.
4. 시그널 맵 삭제를 확인하라는 메시지가 표시됩니다. 삭제를 선택하여 진행하고 시그널 맵을 삭제합니다.

워크플로 모니터 할당량

다음 섹션에는 워크플로 모니터 리소스에 대한 할당량이 포함되어 있습니다. 각 할당량은 '계정별' 기준으로 적용됩니다. 계정의 할당량을 늘려야 하는 경우 다음 표에 달리 명시되지 않는 한 [AWS Service Quotas 콘솔](#)을 사용하여 증가를 요청할 수 있습니다.

할당량

리소스 유형	할당량
CloudWatch 경보 템플릿 그룹	20
CloudWatch 경보 템플릿	200
EventBridge 규칙 템플릿 그룹	20
EventBridge 규칙 템플릿	200
시그널 맵	30
시그널 맵: 단일 시그널 맵에 연결된 CloudWatch 경보 템플릿 그룹	5 이 할당량은 늘릴 수 없습니다.

리소스 유형	할당량
시그널 맵: 단일 시그널 맵에 연결된 EventBridge 규칙 템플릿 그룹	5
	이 할당량은 늘릴 수 없습니다.

워크플로 모니터 사용

워크플로 모니터 콘솔의 개요 및 시그널 맵 섹션에서는 워크플로의 현재 상태와 관련 경보, 지표 및 로그를 검토할 수 있습니다.

주제

- [워크플로 모니터 개요](#)
- [워크플로 모니터에 대한 개요 로그 및 지표](#)
- [워크플로 모니터 시그널 맵 사용](#)

워크플로 모니터 개요

워크플로 모니터 콘솔의 개요 섹션은 시그널 맵에 대한 정보를 한눈에 볼 수 있는 대시보드입니다. 개요 섹션에서는 각 시그널 맵 모니터링의 현재 상태와 CloudWatch 지표 및 관련 CloudWatch 로그를 확인할 수 있습니다. 시그널 맵을 선택하면 해당 시그널 맵의 콘솔 페이지로 이동할 수 있습니다.

개요 필터링

개요 섹션의 검색 창에서 상황에 맞는 제약 조건을 사용하여 시그널 맵 목록을 필터링할 수 있습니다. 검색 창을 선택하면 필터링할 수 있는 속성 목록이 표시됩니다. 속성을 선택하면 같음, 포함, 같지 않음, 포함하지 않음 등의 연산자가 표시됩니다. 연산자를 선택하면 선택한 속성 유형에 따른 리소스 목록이 생성됩니다. 이러한 리소스 중 하나를 선택하면 정의한 제약 조건에 맞는 시그널 맵만 시그널 맵 목록에 표시됩니다.

워크플로 모니터에 대한 개요 로그 및 지표

시그널 맵에 대한 CloudWatch 지표와 로그를 보려면 시그널 맵 이름 옆의 라디오 버튼을 선택합니다. 지표와 로그 모두에 대한 탭 인터페이스가 시그널 맵 목록 아래에 표시됩니다.

CloudWatch 지표

선택한 시그널 맵에 대한 CloudWatch 지표는 상황에 따라 달라지며 해당 시그널 맵 워크플로에 사용된 서비스와 관련된 지표만 표시합니다. 화면의 지표 도구를 사용하여 표시된 지표 기간과 시간 범위를 사용자 지정할 수 있습니다.

CloudWatch 로그

CloudWatch 로그 그룹을 시그널 맵에 연결한 경우 해당 그룹이 여기에 표시됩니다.

워크플로 모니터 시그널 맵 사용

콘솔의 개요 섹션에서 특정 시그널 맵을 선택하여 해당 시그널 맵 및 연결된 모니터링 리소스에 대한 자세한 정보를 볼 수 있습니다.

시그널 맵을 선택하면 시그널 맵과 추가 정보가 포함된 여러 개의 탭 섹션이 표시됩니다.

- CloudWatch 경보
- EventBridge 규칙
- AWS 요소 알림
- Metrics
- 로그
- 기본 세부 정보

시그널 맵 탐색

시그널 맵에는 워크플로 모니터에서 검색된 지원되는 모든 AWS 리소스에 대한 노드가 포함됩니다. 썸네일 미리 보기를 사용할 수 있는 경우 MediaLive 채널 및 MediaPackage 엔드포인트와 같은 특정 리소스에서 콘텐츠의 썸네일 미리 보기를 표시할 수 있습니다.

리소스 노드를 선택하고 작업 드롭다운 메뉴에서 선택한 리소스 세부 정보 보기를 선택하면 관련 서비스의 세부 정보 페이지로 이동합니다. 예를 들어 MediaLive 채널을 선택하고 선택한 리소스 세부 정보 보기를 선택하면 해당 채널에 대한 MediaLive 콘솔의 세부 정보 페이지가 열립니다.

리소스 노드를 선택하면 활성 경보 목록이 해당 노드로만 필터링됩니다. 활성 경보에서 리소스의 대상 ARN을 선택하면 선택한 리소스가 열린 상태로 관련 서비스의 세부 정보 페이지로 이동합니다.

용 듀얼 스택(IPv6) 엔드포인트 AWS Elemental MediaTailor

AWS Elemental MediaTailor 는 IPv4 또는 IPv6를 통해 연결하는 데 사용할 수 있는 듀얼 스택 엔드포인트를 지원합니다. 이러한 엔드포인트는 표준 IPv4-only 엔드포인트와 기능적으로 동일합니다. 서비스 동작은 동일하며 네트워크 전송 계층만 다릅니다.

듀얼 스택 엔드포인트란 무엇입니까?

듀얼 스택 엔드포인트는 IPv4 및 IPv6 주소를 모두 알립니다. 듀얼 스택 엔드포인트를 사용하는 경우 DNS는 네트워크 구성 및 클라이언트 기본 설정에 따라 IPv4 또는 IPv6 주소로 확인됩니다. MediaTailor 리소스, 재생 구성 또는 광고 삽입 동작을 변경할 필요가 없습니다.

지원되는 기능

다음 표에는 듀얼 스택 엔드포인트에 대한 MediaTailor 기능 지원이 나열되어 있습니다.

기능	듀얼 스택 지원
컨트롤 플레APIs(모든 리전)	지원됨
SSAI 데이터 영역(세션 초기화, 매니페스트, 세그먼트)	지원됨
채널 어셈블리 데이터 영역	지원됨
AWS Management Console	지원됨
AWS SDK	곧 출시 예정

듀얼 스택 엔드포인트 형식

MediaTailor 듀얼 스택 엔드포인트는 .api.aws 도메인을 사용합니다.

- 컨트롤 플레인: mediatailor.**region**.api.aws
- SSAI 데이터 영역:에서 재생 구성 세부 정보의 듀얼 스택 엔드포인트 섹션에 표시됩니다 AWS Management Console.

- 채널 어셈블리 데이터 영역:의 채널 출력 세부 정보에 표시됩니다 AWS Management Console.

기존 애플리케이션은 원래 IPv4-only 엔드포인트(`api.mediatailor.region.amazonaws.com`컨트를 플레인인 경우, 데이터 플레인의 `unique-id.mediatailor.region.amazonaws.com` 경우)에서 계속 작동합니다. 사용 가능한 엔드포인트의 전체 목록은 AWS 일반 참조의 [MediaTailor 엔드포인트 및 할당량을 참조하세요](#).

듀얼 스택 엔드포인트 사용

AWS Management Console

듀얼 스택 엔드포인트는 재생 구성 세부 정보 페이지의 듀얼 스택 엔드포인트 섹션에 표시됩니다. 플레이어 또는 CDN 구성에 사용하기 위해 이러한 URLs 직접 복사할 수 있습니다.

AWS Command Line Interface

API 호출 시 듀얼 스택 엔드포인트 URL을 지정합니다.

```
aws mediatailor get-playback-configuration \  
  --name my-config \  
  --endpoint-url https://mediatailor.us-east-1.api.aws
```

*my-config*를 재생 구성의 이름으로 바꾸고 *us-east-1*을 AWS 리전으로 바꿉니다.

AWS SDK

SDK 구성에서 듀얼 스택을 활성화합니다.

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
});
```

고려 사항

듀얼 스택 엔드포인트를 사용할 때는 다음 사항에 유의하세요.

- 듀얼 스택 엔드포인트는 MediaTailor를 사용할 수 AWS 리전 있는 모든에서 사용할 수 있습니다.
- 듀얼 스택 엔드포인트를 사용하기 위해 재생 구성, 광고 결정 서버 설정 또는 CDN 구성을 변경할 필요가 없습니다.
- CDN이 hostname-pattern-based 사용하는 경우 규칙이 `.api.aws` 도메인 형식을 수용하는지 확인합니다.
- 듀얼 스택 엔드포인트에 대한 표준 AWS 지침이 적용됩니다. 자세한 내용은 AWS 일반 참조의 [듀얼 스택 및 FIPS 엔드포인트](#)를 참조하세요.

의 할당량 AWS Elemental MediaTailor

MediaTailor 리소스 및 작업 요청에는 다음 할당량(이전에는 "제한"이라고 함)이 적용됩니다.

AWS Service Quotas 서비스를 사용하여 할당량을 보고 MediaTailor 및 기타 여러 AWS 서비스에 대한 할당량 증가를 요청할 수 있습니다. 자세한 내용은 [Service Quotas 사용 설명서](#)를 참조하세요.

광고 삽입에 대한 할당량

다음 표에서는 AWS Elemental MediaTailor 광고 삽입에 대한 할당량을 설명합니다. 달리 명시되지 않는 한 할당량은 조정할 수 없습니다.

이름	기본 할당량 값	설명
광고 의사결정 서버 (ADS) 길이	25,000	광고 의사결정 서버 (ADS) 사양에 사용할 수 있는 최대 문자 수입니다.
광고 의사결정 서버 (ADS) 리디렉션	7	MediaTailor가 VAST 래퍼 태그에서 따르는 리디렉션의 최대 깊이입니다. 추가 리디렉션이 있는 경우 MediaTailor가 포기합니다.
광고 의사결정 서버 (ADS) 제한 시간	3초(기본값)	ADS(광고 결정 서버)에 대한 연결이 열려 있을 때 MediaTailor가 시간 초과되기 전에 대기하는 최대 시간(초)입니다. ADS의 응답이 없어 연결 시간이 초과되면 MediaTailor 는 광고 게재 가능 영역을 광고로 채울 수

이름	기본 할당량 값	설명	
		없습니다. 이 제한 시간은 개별 ADS 요청 제한 시간 설정(범위: 250~6,000ms)을 통해 구성할 수 있습니다. 자세한 내용은 고급 설정 단원을 참조하십시오.	
광고 삽입 요청	10,000	서버 측 광고 삽입을 수행할 때 개인화된 매니페스트에 대해 수행할 초당 최대 요청 수입니다. 광고 삽입은 매니페스트, 세션 초기화, 추적 데이터 및 광고 세그먼트에 대한 수신 요청을 처리합니다. 이 할당량 은 조정할 수 있습니다.	
구성	1,000	MediaTailor가 허용하는 최대 구성 수입니다.	
content-length-range	512	콘텐츠 오리진 사양의 최대 문자 수입니다.	

이름	기본 할당량 값	설명	
콘텐츠 오리진 서버 제한 시간	2	템플릿 매니페스트를 요청할 때 MediaTailor가 콘텐츠 원본 서버에 대한 열린 연결에서 시간 초과되기 전에 대기하는 최대 시간(초)입니다. 제한 시간은 HTTP 504 (GatewayTimeoutException) 응답 오류를 발생시킵니다.	
매니페스트 크기	2	오리진 재생 매니페스트의 최대 크기입니다. 할당량을 유지하려면 gzip를 사용하여 입력 매니페스트를 MediaTailor로 압축합니다.	
패키지 구성	5	소스당 최대 패키지 구성 수(라이브 또는 온디맨드 비디오).	
미리 가져오기 일정	25	재생 구성당 최대 활성 미리 가져오기 일정 수입니다. 만료된 미리 가져오기 일정은이 제한에 포함되지 않습니다.	

이름	기본 할당량 값	설명	
서버 측 보고 비컨 요청 제한 시간	3초	서버 측 보고를 위해 비컨을 실행할 때 MediaTailor가 서버에 대한 열린 연결을 제한하기 전에 대기하는 최대 초 수입니다. 연결 시간이 초과되면 MediaTailor가 비컨을 실행할 수 없으며 서비스는 CloudWatch의 MediaTailor/AdDecisionServerInteractions 로그에 ERROR_FIRING_BEACON_FAILED 메시지를 기록합니다.	
PRE_SESSION_INITIALIZATION 후크 제한 시간	2초	모든 함수 단계 및 HTTP 호출을 포함하여 PRE_SESSION_INITIALIZATION 후크 실행이 수행할 수 있는 최대 초 수입니다. 후크가 제한 시간을 초과하면 MediaTailor는 모든 출력을 삭제하고 실행된 함수가 없는 것처럼 진행합니다.	

이름	기본 할당량 값	설명
PRE_ADS_REQUEST 후크 제한 시간	2초	모든 함수 단계 및 HTTP 호출을 포함하여 PRE_ADS_REQUEST 후크 실행이 수행할 수 있는 최대 초 수입니다. 후크가 제한 시간을 초과하면 MediaTailor는 모든 출력을 삭제하고 실행된 함수가 없는 것처럼 진행합니다.
POST_ADS_RESPONSE 후크 제한 시간	2초	모든 함수 단계 및 HTTP 호출을 포함하여 POST_ADS_RESPONSE 후크 실행이 수행할 수 있는 최대 초 수입니다. 후크가 제한 시간을 초과하면 MediaTailor는 모든 출력을 삭제하고 실행된 함수가 없는 것처럼 진행합니다.
PRE_MANIFEST_INSERTION 후크 제한 시간	2초	모든 함수 단계 및 HTTP 호출을 포함하여 PRE_MANIFEST_INSERTION 후크 실행이 수행할 수 있는 최대 초 수입니다. 후크가 제한 시간을 초과하면 MediaTailor는 모든 출력을 삭제하고 실행된 함수가 없는 것처럼 진행합니다.

이름	기본 할당량 값	설명	
요청당 총 후크 예산	2초	단일 요청 내에 서 PRE_ADS_REQUEST, POST_ADS_RESPONSE 및 PRE_MANIFEST_INSERTION 후크의 최대 통합 실행 시간입니다. 이전 후크에서 소요된 시간은 이후 후크에서 사용할 수 있는 예산을 줄입니다. 후크가 시작되기 전에 예산이 소진되면 MediaTailor는 해당 후크를 건너뛰고 요청을 계속 처리합니다.	
후크당 최대 함수 수	20	후크당 허용되는 총 함수 실행의 최대 수입니다.	
세션 만료	매니페스트 기간 10배	세션을 종료하기 전에 MediaTailor에서 세션이 비활성 상태로 유지되도록 허용하는 최대 시간입니다. 세션 활동은 플레이어 요청이거나 오리진 서버에 의한 어드밴스입니다. 세션이 만료되면 MediaTailor는 HTTP 400 (Bad Request) 응답 오류를 반환합니다.	

채널 어셈블리의 할당량

다음 표에서는 AWS Elemental MediaTailor 채널 어셈블리의 할당량을 설명합니다. 달리 명시되지 않는 한 할당량은 [조정할](#) 수 있습니다.

이름	기본 할당량 값	설명	
채널 매니페스트 요청, 계정당	400	계정의 모든 채널 어셈블리 채널에 대한 초당 최대 송신 매니페스트 요청 수입입니다.	
채널당 채널 매니페스트 요청	50	단일 채널 어셈블리 채널에 대한 초당 최대 송신 매니페스트 요청 수입입니다.	
채널 출력	5	채널당 최대 출력 수입입니다.	
계정당 채널 수	100	계정당 최대 채널 수입입니다.	
라이브 소스	50	소스 위치의 최대 라이브 소스 수입입니다.	
채널당 프로그램	400	채널당 최대 프로그램 수입입니다.	
세그먼트 전송 구성	5	소스 위치당 세그먼트 전송 구성의 최대 수입입니다.	
소스 위치	50	계정당 소스 위치의 최대 개수	
VOD 소스	1,000	소스 위치의 최대 온디맨드 비디오(VOD) 소스 수입입니다.	

다음 표에서는 AWS Elemental MediaTailor 채널 어셈블리의 제한 한도에 대해 설명합니다. 달리 명시되지 않는 한 할당량은 [조정할](#) 수 있습니다.

이름	transactions-per-second 최대 한도	설명	
ConfigureLogsForChannel	1	채널에 대한 로그를 구성합니다.	
CreateChannel	1	채널을 생성합니다.	
CreateLiveSource	1	라이브 소스를 생성합니다.	
CreateProgram	3	프로그램을 생성합니다.	
CreateSourceLocation	1	소스 위치를 생성합니다.	
CreateVodSource	1	VOD 소스를 생성합니다.	
DeleteChannel	1	채널을 삭제합니다.	
DeleteChannelPolicy	1	채널 정책을 삭제합니다.	
DeleteLiveSource	1	라이브 소스를 삭제합니다.	
DeleteProgram	3	프로그램을 삭제합니다.	
DeleteSourceLocation	1	소스 위치를 삭제합니다.	
DeleteVodSource	1	VOD 소스를 삭제합니다.	

이름	transactions-per-second 최대 한도	설명	
DescribeChannel	5	채널을 설명합니다.	
DescribeLiveSource	5	라이브 소스를 설명합니다.	
DescribeProgram	5	프로그램을 설명합니다.	
DescribeSourceLocation	5	소스 위치를 설명합니다.	
DescribeVodSource	5	VOD 소스를 설명합니다.	
GetChannelPolicy	5	채널 정책을 가져옵니다.	
GetChannelSchedule	5	채널 일정을 가져옵니다.	
ListAlerts	5	알림을 나열합니다.	
ListChannels	5	채널을 나열합니다.	
ListLiveSources	5	라이브 소스를 나열합니다.	
ListPrograms	5	프로그램을 나열합니다.	
ListSourceLocations	5	소스 위치를 나열합니다.	
ListTagsForResource	5	리소스에 대한 태그를 나열합니다.	

이름	transactions-per-second 최대 한도	설명	
ListVodSources	5	VOD 소스를 나열합니다.	
PutChannelPolicy	3	채널 정책을 적용합니다.	
StartChannel	1	채널을 시작합니다.	
StopChannel	1	채널을 중지합니다.	
TagResource	1	리소스에 태그를 지정합니다.	
UntagResource	1	리소스에서 태그를 제거합니다.	
UpdateChannel	1	채널을 업데이트합니다.	
UpdateLiveSource	1	라이브 소스를 업데이트합니다.	
UpdateProgram	1	프로그램을 업데이트합니다.	
UpdateSourceLocation	1	소스 위치를 업데이트합니다.	
UpdateVodSource	1	VOD 소스를 업데이트합니다.	

MediaTailor 문제 해결

로 작업할 때 재생 오류 또는 유사한 문제가 발생하는 경우 이 섹션의 주제를 AWS Elemental MediaTailor 참조하세요.

주제

- [MediaTailor 이벤트 흐름 문제 해결](#)
- [MediaTailor에서 재생 문제 해결](#)
- [비콘 중복 제거](#)
- [파라미터 문제 해결](#)
- [MediaTailor 광고 건너뛰기 문제 해결 가이드](#)

MediaTailor 이벤트 흐름 문제 해결

AWS Elemental MediaTailor 이벤트 흐름을 이해하면 광고 삽입 문제를 해결할 수 있는 강력한 기반을 제공합니다. 이벤트의 시퀀스, 타이밍 및 패턴을 분석하여 문제가 발생하는 위치를 빠르게 식별하고 대안 솔루션을 구현할 수 있습니다.

이 섹션에서는 이벤트 흐름 분석을 사용하여 문제를 진단하기 위한 실제 지침을 제공합니다. 기본 이벤트 흐름 개념을 이해하려면 섹션을 참조하세요 [광고 삽입 이벤트 흐름](#).

불완전한 이벤트 흐름 식별

불완전한 이벤트 흐름은 성공적인 매니페스트 개인화(매니페스트에 개인화된 광고 정보를 삽입하는 MediaTailor 프로세스)에 도달하기 전에 예상 이벤트 시퀀스가 중지될 때 발생합니다. 흐름이 중단되는 위치를 식별하면 광고 삽입 실패의 근본 원인을 정확히 파악하는 데 도움이 됩니다.

일반적인 불완전한 흐름 패턴

이벤트 흐름의 여러 장애 지점은 다음과 같은 특정 유형의 문제를 나타냅니다.

- 광고 기회 감지 후 흐름 중지: MediaTailor가 ADS 요청을 하지 못하게 하는 광고 마커 또는 매니페스트 자체의 문제를 나타냅니다. ADS 연결, 구성 또는 제한 시간 문제는 ADS 요청이 이루어진 후에 발생합니다.
- ADS 요청 후 흐름 중지: ADS 응답 문제, VAST 구문 분석 문제, 크리에이티브 처리 실패, ADS 제한 시간, 연결 오류 또는 요청 시에만 검색되는 잘못된 ADS URLs과 같은 구성 문제를 제안합니다.

- 추적 비컨 누락: 구성 문제 추적, 서버 측 보고 문제 또는 클라이언트 측 구현 격차를 나타낼 수 있습니다.

불완전한 흐름 분석을 위한 CloudWatch 쿼리

이러한 Amazon CloudWatch Logs Insights 쿼리를 사용하여 불완전한 이벤트 흐름을 식별합니다. 필요한 분석 유형에 따라 적절한 로그 그룹에 대해 이러한 쿼리를 실행합니다.

로그 그룹 선택:

- MediaTailor/AdDecisionServerInteractions - 광고 결정 서버 상호 작용, 광고 기회 및 ADS 관련 실패를 분석하는 쿼리에 사용합니다.
- MediaTailor/TranscodeService - 트랜스코딩 문제, 크리에이티브 처리 실패 또는 기타 비 ADS 관련 문제로 인해 광고가 삽입되지 않은 문제를 분석하는 데 사용합니다.

Example 성공적인 매니페스트 개인화 없이 광고 기회 식별

로그 그룹: MediaTailor/AdDecisionServerInteractions

다음 쿼리는 성공적인 매니페스트 개인화로 이어지지 않은 광고 기회를 식별합니다.

```
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 이벤트 흐름 완료율 분석

로그 그룹: MediaTailor/AdDecisionServerInteractions

다음 쿼리는 다양한 이벤트 유형에서 완료율을 분석합니다.

```
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
```

Example비컨 이벤트가 누락된 세션 찾기

로그 그룹: MediaTailor/AdDecisionServerInteractions

다음 쿼리는 가용 시간을 채웠지만 해당 비컨 이벤트가 없는 세션을 식별합니다.

```
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
```

Example트랜스코딩 관련 광고 삽입 실패 식별

로그 그룹: MediaTailor/TranscodeService

다음 쿼리는 성공적인 광고 삽입을 방해하는 트랜스코딩 문제를 식별합니다.

```
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
```

이벤트 타이밍 문제 분석

이벤트 타이밍 분석은 성능 병목 현상을 식별하고 광고 삽입 워크플로를 최적화하는 데 도움이 됩니다. 비정상적인 타이밍 패턴은 최종 사용자 경험에 영향을 미치는 근본적인 문제를 나타내는 경우가 많습니다.

성능 타이밍 임계값

이러한 타이밍 임계값을 사용하여 잠재적 성능 문제를 식별합니다.

- 5초를 초과하는 총 흐름 기간: 최종 사용자 경험에 영향을 미칠 수 있으며 ADS 성능 문제, 오리진 서버 문제(예: 매니페스트 검색 제한 시간) 또는 NAT 게이트웨이, DynamoDB, EC2 또는 기타 시스템 구성 요소의 인프라 문제를 포함한 내부 MediaTailor 문제를 나타낼 수 있습니다.
- 2초를 초과하는 ADS 응답 시간: ADS 성능 문제 또는 네트워크 지연 시간 문제를 제안합니다.
- 1초를 초과하는 매니페스트 개인화: NAT Gateway, DynamoDB, EC2 또는 기타 구성 요소의 인프라 제약을 포함한 크리에이티브 처리 지연, 오리진 서버 문제(예: 매니페스트 검색 제한 시간) 또는 내부 MediaTailor 시스템 문제를 나타낼 수 있습니다.

타이밍 분석 쿼리

이러한 쿼리를 사용하여 이벤트 타이밍 패턴을 분석합니다.

Example 총 이벤트 흐름 기간 측정

다음 쿼리는 이벤트 흐름의 총 지속 시간을 측정하고 5초를 초과하는 시간을 식별합니다.

```
fields @timestamp, eventType, avail.availId
| filter avail.availId = "your-avail-id"
| filter eventType in ["AD_MARKER_FOUND", "FILLED_AVAIL"]
| sort @timestamp asc
| stats min(@timestamp) as start_time, max(@timestamp) as end_time by avail.availId
| eval duration_seconds = (end_time - start_time) / 1000
| where duration_seconds > 5
```

Example ADS 응답 타이밍 분석

다음 쿼리는 ADS 응답 시간을 분석하고 2초를 초과하는 시간을 식별합니다.

```
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 느린 매니페스트 개인화 식별

다음 쿼리는 1초 이상 걸리는 매니페스트 개인화 프로세스를 식별합니다.

```
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
```

일반적인 이벤트 흐름 문제 및 해결 방법

이 섹션에서는 자주 발생하는 이벤트 흐름 문제에 대한 솔루션을 문제 유형 및 증상별로 정리하여 제공합니다.

광고 결정 서버 요청 실패

증상: 광고 기회 감지 후 이벤트 흐름이 중지됩니다. ADS 요청 이벤트가 로깅되지 않았습니다.

일반적인 원인 및 해결 방법

- ADS URL 구성 오류: 재생 구성의 ADS URL이 올바르고 액세스 가능한지 확인합니다. 광고 상호 작용 로그에는 ADS 요청 이벤트(MAKING_ADS_REQUEST)가 표시되지만 해당하는 VAST 응답은 표시되지 않으며, 종종 ERROR_UNKNOWN 또는 유사한 오류 이벤트가 수반됩니다.
- 네트워크 연결 문제: 방화벽 규칙 및 DNS 해결을 포함하여 MediaTailor와 ADS 간의 네트워크 연결을 확인합니다.
- SSL/TLS 인증서 문제: ADS가 신뢰할 수 있는 인증 기관의 유효한 SSL 인증서를 사용하는지 확인합니다. 특히 Google Ad Manager의 경우 [AWS Support](#)에 문의하여 Google의 SSL 인증서를 수락하는 구성 플래그를 활성화해야 할 수 있습니다.

진단 쿼리

다음 쿼리는 이벤트 시퀀스를 추적하여 ADS 요청 실패를 진단하는 데 도움이 됩니다.

```
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
```

광고 결정 서버 응답 실패

증상: ADS 요청이 성공했지만 MediaTailor가 응답을 받지 못하거나 구문 분석 오류가 발생합니다.

일반적인 원인 및 해결 방법

- 잘못된 VAST 형식: VAST 사양 표준과 비교하여 ADS VAST 응답을 검증합니다.
- ADS 제한 시간 문제: ADS 제한 시간 설정을 늘리거나 ADS 응답 시간을 최적화합니다.
- 빈 광고 인벤토리: ADS 구성에서 광고 인벤토리 가용성 및 대상 지정 기준을 확인합니다.

진단 쿼리

다음 쿼리는 요청 및 응답 이벤트를 검사하여 ADS 응답 실패를 진단하는 데 도움이 됩니다.

```
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
```

매니페스트 개인화 실패

증상: VAST 응답을 수신했지만 매니페스트 개인 맞춤이 실패하거나 광고를 건너뛵니다.

일반적인 원인 및 해결 방법:

- 크리에이티브 트랜스코딩 문제: 광고가 삽입 전에 NEW_CREATIVE 트랜스코딩이 필요한 인지 확인합니다. MediaTailor/TranscodeService 로그에서 INTERNAL_ERROR MISSING_VARIANTS, 또는와 같은 오류 이벤트를 검사하여 트랜스코딩 오류를 확인할 수도 있습니다 PROFILE_NOT_FOUND.
- 기간 불일치 문제: 광고 기간이 사용 가능한 광고 중단 기간 내에 있는지 확인합니다.
- 개인화 임계값 문제: 재생 구성에서 개인화 임계값 설정을 검토합니다.

진단 쿼리

다음 쿼리는 VAST 응답과 채워진 공간을 검사하여 매니페스트 개인화 실패를 진단하는 데 도움이 됩니다.

```
fields @timestamp, eventType, sessionId, skippedAds
```

```
| filter sessionId = "your-session-id"
| filter eventType in ["VAST_RESPONSE", "FILLED_AVAIL", "WARNING_NO_ADVERTISEMENTS"]
| sort @timestamp asc
```

건너뛴 광고 이유에 대한 쿼리

다음 쿼리는 광고를 건너뛰는 이유에 대한 자세한 정보를 제공합니다.

```
fields @timestamp, eventType, sessionId, skippedAds.reason, skippedAds.creativeUniqueId
| filter sessionId = "your-session-id"
| filter eventType = "WARNING_NO_ADVERTISEMENTS" or ispresent(skippedAds)
| sort @timestamp asc
```

건너뛴 광고 이유 및 창의적인 고유 IDs 대한 쿼리

다음 쿼리는 각 가용 구간의 처음 두 광고에 대한 이유와 창의적인 고유 IDs를 포함하여 건너뛴 자세한 광고 정보를 제공합니다.

```
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
```

비컨 실패 추적

증상: 매니페스트 개인화는 성공했지만 비컨 추적은 누락되거나 실패했습니다.

일반적인 원인 및 해결 방법

- 클라이언트 측 구현 문제: 대부분의 추적 비컨 문제는 클라이언트 측 추적에 충분한 빈도로 추적 URLs 폴링하지 않는 것과 같은 클라이언트 측 구현 문제 또는 플레이어별 비컨 실행 로직 문제에서 비롯됩니다.
- URL 접근성 문제 추적: VAST 응답의 추적 URLs에 액세스할 수 있는지 확인하고 적절한 응답을 반환합니다. URLs 연결할 수 없거나 MediaTailor에서 성공적인 추적 응답 전송을 방해하는 내부 문제가 발생할 때 문제가 발생할 수 있습니다.

- 플레이어 세그먼트 요청 문제: 클라이언트 플레이어가 실제로 세그먼트를 요청하지 않으면 명백한 추적 비컨 오류가 발생할 수 있습니다. 이로 인해 비컨이 전송되지 않고 추적 실패로 표시되지만 실제로는 비컨 문제가 아닌 플레이어 구현 문제입니다.

진단 쿼리

다음 쿼리는 채워진 가용 시간 및 비컨 이벤트를 검사하여 비컨 실패 추적을 진단하는 데 도움이 됩니다.

```
fields @timestamp, eventType, sessionId
| filter sessionId = "your-session-id"
| filter eventType in ["FILLED_AVAIL", "BEACON_FIRED", "ERROR_FIRING_BEACON_FAILED"]
| sort @timestamp asc
```

이벤트 흐름 모니터링 모범 사례

이러한 모니터링 사례를 구현하여 이벤트 흐름 문제를 사전에 식별하고 해결합니다.

CloudWatch 경보 설정

Amazon CloudWatch 경보를 생성하여 주요 이벤트 흐름 지표를 모니터링합니다.

- 흐름 완료율 경보: 광고 기회 대비 성공적인 매니페스트 개인화 비율이 허용 가능한 임계값 아래로 떨어지면 경고합니다.
- ADS 응답 시간 경보: 평균 ADS 응답 시간을 모니터링하고 성능 임계값을 초과할 때 경고합니다.
- 오류율 경보: 오류 이벤트 빈도를 추적하고 특정 오류 유형의 비정상적인 스파이크에 대해 경고합니다.

정기 모니터링 쿼리

이러한 쿼리를 정기적으로 실행하여 이벤트 흐름 상태에 대한 가시성을 유지합니다.

Example일일 이벤트 흐름 성공률

다음 쿼리는 이벤트 유형별 이벤트 흐름 성공률에 대한 일일 개요를 제공합니다.

```
fields @timestamp, eventType
| filter @timestamp > datefloor(@timestamp, 1d)
```

```
| stats count() as total_events by eventType
| sort total_events desc
```

Example시간당 오류율 추세

다음 쿼리는 시간별 오류율을 추적하여 유행하는 문제를 식별합니다.

```
fields @timestamp, eventType
| filter eventType like /ERROR_/
| stats count() as error_count by datefloor(@timestamp, 1h) as hour
| sort hour desc
```

성능 최적화 지침

이벤트 흐름 분석을 사용하여 광고 삽입 성능을 최적화합니다.

- ADS 최적화: ADS 공급자와 협력하여 응답 시간을 최적화하고 지연 시간을 줄입니다.
- 크리에이티브 준비: 콘텐츠 프로필과 일치하고 처리 지연을 줄이기 위해 광고 크리에이티브를 사전 트랜스코딩합니다.
- 구성 튜닝: 이벤트 흐름 분석을 기반으로 제한 시간 설정, 개인화 임계값 및 기타 구성 파라미터를 조정합니다.

추가 문제 해결 리소스

이벤트 흐름 분석 이외의 추가 문제 해결 지침:

- 자세한 로그 형식 정보 및 기술 사양은 섹션을 참조하세요 [로그 보기](#).
- 일반적인 광고 삽입 문제에 대한 포괄적인 문제 해결은 섹션을 참조하세요 [일반적인 문제 해결](#).
- 설정 지침 모니터링 및 알림은 섹션을 참조하세요 [Amazon CloudWatch 지표 AWS Elemental MediaTailor 를 사용한 모니터링](#).
- 디버그 로깅 절차는 섹션을 참조하세요 [디버그 로그 생성](#).

MediaTailor에서 재생 문제 해결

이 단원에서는 플레이어 소프트웨어를 테스트하거나 플레이어 요청을 정상적으로 처리하는 동안 수신할 수 있는 HTTP 오류 코드에 대한 정보를 제공합니다.

Note

PutPlaybackConfiguration 및와 같은 구성 작업 중에 AWS Elemental MediaTailor API에서 오류가 발생할 수도 있습니다GetPlaybackConfiguration. 이러한 유형의 오류에 대한 자세한 내용은 [AWS Elemental MediaTailor API 참조](#)를 참조하세요.

플레이어가 AWS Elemental MediaTailor직접 또는 CDN을 통해에 요청을 보내면 MediaTailor는 상태 코드로 응답합니다. MediaTailor가 요청을 성공적으로 처리하면 HTTP 상태 코드를 반환하여 채워진 매니페스트와 함께 성공을 200 OK나타냅니다. 요청이 실패하면 MediaTailor는 HTTP 상태 코드, 예외 이름 및 오류 메시지를 반환합니다.

AWS Elemental MediaTailor 는 두 가지 오류 클래스를 반환합니다.

- 클라이언트 오류 - 일반적으로 잘못된 형식의 요청, 잘못된 파라미터 또는 잘못된 URL과 같은 요청 자체의 문제로 인해 발생하는 오류입니다. 이러한 오류는 HTTP 4xx 응답 코드를 지닙니다.
- 서버 오류 - 일반적으로 MediaTailor 또는 광고 결정 서버(ADS) 또는 오리진 서버와 같은 종속성 중 하나에 문제가 발생하여 발생하는 오류입니다. 이러한 오류는 HTTP 5xx 응답 코드를 지닙니다.

주제

- [에서 반환한 클라이언트 재생 오류 AWS Elemental MediaTailor](#)
- [에서 반환한 서버 재생 오류 AWS Elemental MediaTailor](#)
- [재생 오류 예제](#)

에서 반환한 클라이언트 재생 오류 AWS Elemental MediaTailor

일반 지침:

- 응답의 머리말과 본문에서 대부분의 오류에 대한 자세한 정보를 확인할 수 있습니다.
- 일부 오류의 경우, 구성 설정을 확인해야 합니다. 재생 구성에 대한 설정을 검색할 수 있습니다 AWS Elemental MediaTailor. API의 경우, 리소스는 GetPlaybackConfiguration/**Name**입니다. 자세한 내용은 [AWS Elemental MediaTailor API 참조](#)를 참조하세요.

다음 표에는의 매니페스트 조작 활동에서 반환되는 클라이언트 오류 코드 AWS Elemental MediaTailor, 가능한 원인 및 이를 해결하기 위해 취할 수 있는 작업이 나열되어 있습니다.

코드	예외 이름	의미	수행할 작업
40	BadRequestException	형식 또는 콘텐츠에 하나 이상의 오류가 발생하여 MediaTailor가 요청을 처리할 수 없습니다. 파라미터의 형식이 부적절하게 지정되었거나 요청에 잘못된 재생 구성 또는 세션 ID가 포함되어 있을 수 있습니다.	요청의 형식이 적절히 지정되고 정확한 정보를 포함하는지 확인합니다. 플레이어의 재생 엔드포인트 설정이 GetPlaybackConfiguration 이 반환하는 ManifestEndpointPrefix 설정과 일치하는지 확인합니다. 요청을 재시도합니다.
40	AccessDeniedException	요청에 제공된 호스트 헤더가 MediaTailor 재생 URL에 구성된 매니페스트 엔드포인트 접두사와 일치하지 않습니다. CDN이 잘못 구성되었을 수 있습니다.	CDN 설정을 확인하고 MediaTailor에 올바른 매니페스트 엔드포인트 접두사를 사용하고 있는지 확인합니다. 요청을 재시도합니다.
40	NotFoundException	MediaTailor가 지정된 정보를 찾을 수 없습니다. 서비스 내의 어떤 항목에도 매핑되지 않은 URL, 정의되지 않은 구성 또는 사용할 수 없는 세션이 원인일 수 있습니다.	구성 및 요청의 타당성을 확인한 다음 세션을 재초기화합니다.
40	ConflictException	플레이어에서 한 세션에 대하여 여러 개의 재생 목록을 동시에 로드하려고 시도했습니다. 따라서 MediaTailor가 세션 일관성 충돌을 감지했습니다. 이 문제는 HLS 플레이어의 경우 발생합니다.	플레이어에서 재생 목록을 한 번에 하나씩 요청하는지 확인하십시오. 이는 HLS 사양에 따른 것입니다.
41	Gone	AWS 지원 운영자가 플레이어 세션 또는 고객 구성을 차단했습니다. 드물지만 단일 세션 또는 구성에 대한 잘못된 트래픽에서 발생하는 매우 많은 양의 4xx 요청이 감지되는 경	요청이 차단되지 않아야 한다고 생각되면 AWS Support 에 문의하십시오. 지원팀에서 상황을 확인하고 해당하는 경우 차단 필터를 제거할 수 있습니다.

코드	예외 이름	의미	수행할 작업
		우 AWS Support가 이를 수행합니다.	

추가 지원이 필요한 경우 [AWS Support](#)에 문의하십시오.

에서 반환한 서버 재생 오류 AWS Elemental MediaTailor

일반 지침:

- 응답의 머리말과 본문에서 대부분의 오류에 대한 자세한 정보를 확인할 수 있습니다.
- 일부 오류의 경우, 구성 설정을 확인해야 합니다. 재생 구성에 대한 설정을 검색할 수 있습니다 AWS Elemental MediaTailor. API의 경우, 리소스는 `GetPlaybackConfiguration/Name`입니다. 자세한 내용은 [AWS Elemental MediaTailor API 참조](#)를 참조하세요.

다음 표에는 매니페스트 조작 활동에서 반환된 서버 오류 코드 AWS Elemental MediaTailor, 가능한 원인 및 이를 해결하기 위해 취할 수 있는 작업이 나열되어 있습니다.

코드	예외 이름	의미	수행할 작업
50	InternalServiceError	처리되지 않은 예외입니다.	요청을 다시 시도하세요. 문제가 지속되면에서 해당 AWS 리전에 대해 MediaTailor의 보고된 상태를 확인합니다 https://status.aws.amazon.com/ .
50	BadGatewayException	오리진 서버 주소 또는 광고 결정 서버(ADS) 주소가 잘못되었습니다. 잘못된 주소의 예로 프라이빗 IP 주소와 localhost 를 들 수 있습니다.	구성에 ADS 및 오리진 서버에 대한 올바른 설정이 있는지 확인한 다음 요청을 재시도합니다.

코드	예외 이름	의미	수행할 작업
50	UnsupportedManifestException	MediaTailor가 개인화할 수 없도록 오리진 매니페스트가 변경되었거나 MediaTailor가 오리진의 매니페스트 형식을 지원하지 않습니다.	이런 상황은 개별 세션에만 영향을 미칠 수 있습니다. 세션을 재초기화합니다. 일반적으로 뷰어에서 페이지를 새로 고쳐 그렇게 할 수 있습니다. 문제가 지속되면 MediaTailor가 오리진의 매니페스트 형식을 지원하는지 확인합니다. 자세한 내용은 콘텐츠 소스 통합 단원을 참조하세요.
50	LoadShed	MediaTailor에서 요청을 처리하는 동안 리소스 제약이 발생했습니다.	요청을 다시 시도하세요. 문제가 지속되면에서 해당 AWS 리전에 대해 MediaTailor의 보고된 상태를 확인합니다 https://status.aws.amazon.com/ .
50	ThrottlingException	초당 트랜잭션이 할당량에 도달했으며 MediaTailor에서 사용을 제한하고 있습니다.	요청을 다시 시도하세요. 에서 해당 AWS 리전에 대해 보고된 MediaTailor 상태를 확인할 수도 있습니다 https://status.aws.amazon.com/ . 초당 트랜잭션 할당량을 늘려야 할 수 있습니다. 자세한 내용은 the section called “광고 삽입에 대한 할당량” 단원을 참조하십시오.
50	GatewayTimeoutException	MediaTailor가 오리진 서버에 연결하는 동안 제한 시간이 발생했습니다.	요청을 다시 시도하세요. 문제가 지속되는 경우 오리진 서버의 상태를 확인하고 오리진 서버가 the section called “광고 삽입에 대한 할당량” 에 나열된 콘텐츠 오리진 서버 시간 제한 내에 응답하고 있는지 확인합니다.

추가 지원이 필요한 경우 [AWS Support](#)에 문의하십시오.

재생 오류 예제

이 섹션에서는 명령줄 상호 작용에서 볼 수 있는 재생 오류의 몇 가지 예를 나열합니다 AWS Elemental MediaTailor.

다음 예제에서는 AWS Elemental MediaTailor 와 광고 결정 서버(ADS) 또는 오리진 서버 간에 제한 시간이 발생할 때의 결과를 보여줍니다.

```
~[ ]> 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: mediataylor.us-west-2.amazonaws.com
* Server certificate: Amazon
* Server certificate: Amazon Root CA 1
* Server certificate: Starfield Services Root Certificate Authority - G2
> GET /v1/master/123456789012/Multiperiod_DASH_Demo/index.mpd HTTP/1.1
> Host: 111122223333444455556666123456789012.mediatailor.us-west-2.amazonaws.com
> User-Agent: curl/7.43.0
> Accept: */*
>
< HTTP/1.1 504 Gateway Timeout
< Date: Thu, 29 Nov 2018 18:43:14 GMT
< Content-Type: application/json
< Content-Length: 338
< Connection: keep-alive
< x-amzn-RequestId: 123456789012-123456789012
< x-amzn-ErrorType: GatewayTimeoutException:http://internal.amazon.com/coral/com.amazon.elemental.midas.mms.coral/
<
* Connection #0 to host 111122223333444455556666123456789012.mediatailor.us-west-2.amazonaws.com left intact
{"message":"failed to generate manifest: Unable to obtain template playlist. origin URL:[https://777788889999.mediapackage.us-west-2.amazonaws.com/out/v1/444455556666111122223333/index.mpd], asset path: [index.mpd], sessionId:[123456789012123456789012] customerId:[123456789012]"}%
```

비콘 중복 제거

이 섹션에서는 서버 측 추적에서 MediaTailor의 비콘 중복 제거 기능에 대한 문제 해결 지침을 제공합니다.

증상: 이전에 관찰된 것보다 낮은 비콘 수

원인: 비콘 중복 제거는 이전에 여러 번 계산되었던 중복 이벤트를 방지하고 있습니다.

해결 방법: 이는 예상되는 동작이며 더 정확한 측정을 나타냅니다. 새 수는 기술 세그먼트 요청이 아닌 실제 광고 시청 이벤트를 나타냅니다. 중복 제거된 수를 반영하도록 보고 기준을 업데이트합니다.

증상: 비콘 수는 플레이어 구현마다 다릅니다.

원인: 플레이어마다 세그먼트 요청 패턴이 다를 수 있지만 중복 제거는 이러한 차이에 관계없이 일관된 비콘 수를 보장합니다.

해결 방법: 예상되는 동작입니다. 중복 제거는 다양한 플레이어 유형 및 네트워크 조건에서 일관된 측정을 제공합니다.

증상: 예상 비콘 수와 실제 비콘 수 간의 불일치

원인: 이전 예상은 중복 비콘 실행으로 인해 부풀려진 수를 기반으로 했을 수 있습니다.

해결 방법: 기록 데이터를 검토하여 중복 비콘의 패턴을 식별합니다. 새로운 중복 제거 개수를 측정 및 보고의 정확한 기준으로 사용합니다.

파라미터 문제 해결

동적 광고 변수, 매니페스트 쿼리 파라미터, 문자 제한, 길이 제한 및 구성 별칭과 관련된 문제를 해결하려면 [MediaTailor 파라미터 문제 해결 가이드](#)의 섹션을 참조하세요 [ADS 요청에 대한 MediaTailor 동적 광고 변수](#).

MediaTailor 광고 건너뛰기 문제 해결 가이드

광고 건너뛰기는 MediaTailor 고객이 보고하는 가장 일반적인 문제 중 하나입니다. 이 가이드는 광고 건너뛰기 문제를 식별하고 step-by-step 문제 해결 절차를 제공하는 데 도움이 됩니다.

증상 및 영향

광고 건너뛰기가 발생하면 다음 증상이 관찰될 수 있습니다.

- 예상 광고 시간 동안 광고가 표시되지 않음
- 광고 시간에는 광고 대신 콘텐츠가 표시됩니다.
- 다양한 보기 세션에서 일관성 없는 광고 재생
- 다양한 건너뛰기 이유가 있는 AdSkipped 이벤트를 보여주는 CloudWatch 로그

광고 건너뛰기는 수익에 직접적인 영향을 미치며 시청자 경험을 저하시킬 수 있습니다. 이러한 문제를 즉시 해결하여 최적의 성능을 유지합니다.

건너뛰기 이유 식별

CloudWatch Logs Insights를 사용하여 건너뛴 광고에 대해 MediaTailor/AdDecisionServerInteractions 로그 그룹을 쿼리합니다.

```
fields @timestamp, avail.availId, skippedAds.0.skippedReason,
skippedAds.0.creativeUniqueId
| filter eventType = "FILLED_AVAIL" and ispresent(skippedAds.0.skippedReason)
| sort @timestamp desc
```

이 쿼리는 패턴을 식별하는 데 도움이 되는 특정 이유와 함께 최신 광고 건너뛰기 이벤트를 반환합니다.

주제

- [MediaTailor NEW_CREATIVE 광고 건너뛰기 문제 해결](#)
- [MediaTailor ADS 제한 시간 광고 문제 해결 건너뛰기](#)
- [MediaTailor VAST 구문 분석 광고 건너뛰기 문제 해결](#)
- [MediaTailor 기간 불일치 광고 건너뛰기 문제 해결](#)
- [MediaTailor 세션 변수 광고 건너뛰기 문제 해결](#)
- [MediaTailor 광고 모니터링 및 알림 건너뛰기](#)
- [MediaTailor 광고 건너뛰기 방지 모범 사례](#)
- [MediaTailor 광고 건너뛰기 참조 가이드](#)

MediaTailor NEW_CREATIVE 광고 건너뛰기 문제 해결

NEW_CREATIVE 이유와 함께 광고를 건너뛰면 삽입 전에 트랜스코딩이 필요한 광고가 AWS Elemental MediaTailor 발생합니다. 이 문제 해결 가이드는 원인을 설명하고 step-by-step 해결 절차를 제공합니다.

NEW_CREATIVE 건너뛰기의 원인

MediaTailor는 크리에이티브 ID, AWS 계정 ID 및 트랜스코딩 변형 세트(기본 콘텐츠 스트림의 재생 변환)의 세 가지 주요 요소를 기반으로 광고를 트랜스코딩합니다. 크리에이티브 ID 또는 트랜스코딩 변형 세트의 일부가 다르면 MediaTailor는 광고를 트랜스코딩이 필요한 새 변형으로 인식합니다.

비트레이트 변경

비트레이트 변경으로 인해 가장 가까운 8,000비트로 반올림한 후 비트레이트가 일치하지 않으면 NEW_CREATIVE 건너뛰기가 발생할 수 있습니다. 이는 일반적으로 다음과 같은 상황에서 발생합니다.

- 기본 콘텐츠 스트림은 기본 매니페스트를 변경합니다.
- 새 세션은 기존 세션과 다른 변형으로 생성됩니다.
- 콘텐츠 스트림 비트레이트가 재생 세션 간에 일관되지 않음

수집 작업 대역폭 분산

수집 작업의 경우 MediaTailor는 광고를 콘텐츠 스트림과 일치시킬 때 15% 대역폭 분산 임계값을 사용합니다. 라이브 스트림의 대역폭과 수집된 HLS 파일의 대역폭 간에 15% 이상의 차이가 있는 경우 광고가 다시 트랜스코딩됩니다. 이로 인해 트랜스코딩 프로세스 중에 NEW_CREATIVE 건너뛰기가 발생할 수 있습니다.

이 동작은 일반적으로 다음과 같은 경우에 발생합니다.

- 동일한 사전 트랜스코딩된 광고가 라이브 스트리밍과 VOD 콘텐츠 모두에 사용됩니다.
- 수집된 콘텐츠의 인코딩 파라미터가 원래 라이브 스트림과 다릅니다.
- 대역폭 차이가 라이브 콘텐츠와 수집된 콘텐츠 간의 15% 임계값을 초과함

크리에이티브 ID 충돌

MediaTailor가 이미 트랜스코딩된 미디어 파일에 대해 다른 크리에이티브 ID를 발견하면 다음 시퀀스가 발생합니다.

1. NEW_CREATIVE 이유와 함께 광고를 건너뛵니다.
2. 이로 인해 불필요한 트랜스코딩 시도가 발생합니다.
3. 크리에이티브가 DUPLICATE_TRANSCODE 또는 COPY_DEDUP으로 표시됩니다.

Note

MediaTailor는 트랜스코딩된 광고를 만료하거나 삭제하지 않습니다. MediaTailor 소유 S3 버킷에 무기한 저장합니다.

해결 단계

NEW_CREATIVE 광고 건너뛰기 문제를 해결하려면:

1. 광고 결정 서버가 동일한 광고 콘텐츠에 대해 일관된 크리에이티브 IDs를 반환하는지 확인합니다.
2. 콘텐츠 스트림이 일관된 비트 전송률 및 변형 세트를 유지하는지 확인합니다.
3. 재생 전에 광고가 트랜스코딩되도록 광고 미리 가져오기를 구현하는 것이 좋습니다. 자세한 내용은 [the section called “광고 미리 가져오기”](#) 단원을 참조하십시오.
4. 지속적인 문제의 경우 [AWS Support](#)에 문의하여 추가 문제 해결 지원을 받으세요.

NEW_CREATIVE 패턴 모니터링

이 CloudWatch Logs Insights 쿼리를 사용하여 NEW_CREATIVE 광고 건너뛰기의 패턴을 분석할 수 있습니다.

```
fields @timestamp, sessionId, creativeId, skipReason, MediaFileSourceUrl
| filter skipReason = "NEW_CREATIVE"
| stats count() by creativeId, MediaFileSourceUrl
| sort count desc
| limit 50
```

MediaTailor ADS 제한 시간 광고 문제 해결 건너뛰기

ADS_TIMEOUT 또는 관련 이유로 광고를 건너뛰면 광고 결정 서버와의 연결 또는 성능 문제가 발생합니다.는 광고를 성공적으로 삽입하려면 ADS와의 안정적인 통신이 AWS Elemental MediaTailor 필요합니다. 이 문제 해결 가이드에서는 이러한 연결 문제를 식별하고 해결하는 방법을 설명합니다.

일반적인 ADS 연결 문제

일반적인 ADS 연결 문제는 다음과 같습니다.

- MediaTailor에서 광고 결정 서버에 액세스할 수 없음
- 구성된 제한 시간 내에 ADS가 응답하지 않음
- ADS가 피크 기간 동안 요청 볼륨을 처리할 수 없음
- MediaTailor와 ADS 간의 네트워크 연결 문제

해결 단계

ADS 제한 시간 문제를 해결하려면:

1. MediaTailor에서 광고 결정 서버에 액세스할 수 있는지 확인합니다.
2. ADS가 구성된 제한 시간 내에 응답하는지 확인합니다.
3. ADS가 피크 기간 동안 요청 볼륨을 처리할 수 있는지 확인합니다.
4. 기본 ADS를 사용할 수 없는 경우 폴백 광고 전략을 구현하는 것이 좋습니다.

ADS 성능 모니터링

AdDecisionServer.Timeouts 지표에 대한 CloudWatch 경보를 설정하여 ADS 연결 문제를 사전에 감지합니다.

MediaTailor VAST 구문 분석 광고 건너뛰기 문제 해결

VAST_PARSING_ERROR 또는를 사용하여 광고를 건너뛰면 VAST 응답 또는 광고 미디어 파일에 문제가 MEDIA_FILE_UNAVAILABLE 있습니다.는 성공적인 광고 삽입을 위해 적절한 형식의 VAST 응답과 액세스 가능한 미디어 파일이 AWS Elemental MediaTailor 필요합니다. 이 문제 해결 가이드에서는 이러한 문제를 식별하고 해결하는 방법을 설명합니다.

일반적인 VAST 문제

일반적인 VAST 문제는 다음과 같습니다.

- VAST 사양을 준수하지 않는 VAST 응답 형식
- 공개적으로 액세스할 수 없는 VAST 응답의 미디어 파일 URLs
- 특수 문자가 VAST XML로 제대로 인코딩되지 않음
- VAST 응답에 필수 미디어 파일 또는 형식이 누락됨

해결 단계

VAST 구문 분석 문제를 해결하려면:

1. VAST 사양과 비교하여 VAST 응답 형식을 검증합니다.
2. VAST 응답의 모든 미디어 파일 URLs에 공개적으로 액세스할 수 있는지 확인합니다.
3. VAST XML에서 특수 문자의 인코딩이 적절한지 확인합니다.
4. VAST 응답에 MediaTailor와 호환되는 형식의 미디어 파일이 포함되어 있는지 확인합니다.

VAST 래퍼 문제 해결

INVALID_VAST_WRAPPER_AD 또는 관련 문제의 경우REJECTED_REPLICA_VAST:

- VAST 사양과 비교하여 VAST 래퍼 응답 검증
- 모든 래퍼 요소의 형식이 올바르게 지정되고 유효한 VASTAdTagURI가 포함되어 있는지 확인합니다.
- 광고 서버 구성에서 중복 콘텐츠 감지 정책 확인
- VAST 응답에 동일한 세션 내의 고유한 크리에이티브 콘텐츠가 포함되어 있는지 확인합니다.

MediaTailor 기간 불일치 광고 건너뛰기 문제 해결

AVAIL_DURATION_EXCEEDED, LEFTOVER_AVAIL_EXCEEDED_THRESHOLD또는 형식 관련 이유로 광고를 건너뛰는 경우 기간 또는 형식 호환성 문제가 있습니다. 이는 광고 콘텐츠와 사용 가능한 광고 중단 시간 간의 적절한 기간 형식 지정 및 일치가 AWS Elemental MediaTailor 필요합니다. 이 문제 해결 가이드에서는 이러한 기간 관련 문제를 식별하고 해결하는 방법을 설명합니다.

일반적인 지속 시간 문제

일반적인 기간 문제는 다음과 같습니다.

- 사용 가능한 광고 중단 기간보다 긴 광고
- 잘못된 지속 시간을 알리는 콘텐츠의 잘못된 광고 중단 마커
- 광고 삽입에 대한 개인화 임계값이 충족되지 않음
- EXT-X-CUE-OUT 태그의 잘못된 기간 형식

기간 형식 요구 사항

EXT-X-CUE-OUT 태그 지속 시간 파라미터는 ISO 8601 지속 시간 형식이 아닌 정수 값으로 형식이 지정되어야 합니다.

기간 형식 요구 사항

형식	예제	Status
정수(정답)	32	지원됨 - 32초를 나타냅니다.
십진수(정답)	30.000	지원됨 - 30초를 나타냅니다.
ISO 8601(잘못됨)	PT32S	지원되지 않음 - 삽입 실패 발생

해결 단계

기간 불일치 문제를 해결하려면:

1. ADS가 사용 가능한 광고 중단 기간에 맞는 광고를 반환하는지 확인합니다.
2. 콘텐츠의 광고 중단 마커가 의도한 지속 시간을 올바르게 나타내는지 확인합니다.
3. EXT-X-CUE-OUT 기간 파라미터가 정수 형식을 사용하는지 확인합니다.
4. 사용 사례에 적합한 경우 개인화 임계값을 조정하는 것이 좋습니다.

MediaTailor 세션 변수 광고 건너뛰기 문제 해결

세션 변수는 광고 타겟팅 및 선택에 중요한 역할을 합니다 AWS Elemental MediaTailor. 잘못된 세션 변수 구성은 광고 건너뛰기 문제의 일반적인 원인입니다. 이 포괄적인 문제 해결 가이드에서는 성공적인 광고 삽입을 방해할 수 있는 세션 변수 문제를 식별하고 해결하는 방법을 설명합니다.

일반적인 세션 변수 문제

일반적인 세션 변수 문제는 다음과 같습니다.

- 필수 변수 누락: 광고 결정 서버에 제공되지 않는 특정 변수가 필요할 수 있습니다.
- 잘못된 변수 구문: 변수는 올바른 구문을 사용해야 합니다(예: [session.id] 대신 \${session.id}).
- URL 인코딩 문제: 변수 값의 특수 문자에 적절한 URL 인코딩이 필요할 수 있습니다.
- 일관되지 않은 플레이어 파라미터: 플레이어 파라미터는 세션 간에 일관되게 전달되어야 합니다.
- 동적 변수 확인 실패: 해결할 수 없는 변수는 빈 문자열로 대체됩니다.
- SCTE-35 UPID 구문 분석 문제: 분할 UPID 처리에 문제가 있으면 세션 변수 해결에 실패할 수 있습니다.

세션 변수 확인

세션 변수가 제대로 확인되고 있는지 확인하려면:

- MediaTailor 구성에 대한 디버그 로깅 활성화
- MediaTailor/AdDecisionServerInteractions 로그 그룹에서 실제 ADS 요청 URLs
- 템플릿 URL의 모든 변수가 적절한 값으로 대체되었는지 확인합니다.
- 해결 실패를 나타낼 수 있는 빈 문자열로 대체된 변수를 찾습니다.

SCTE-35 UPID 구문 분석 문제 해결

SCTE-35 분할 UPID 처리에 문제가 있으면 세션 변수 문제가 발생할 수 있습니다.

- 형식 요구 사항: UPIDsegmentation_upid_type은 12여야 하며 적절한 처리를 format_identifier 위해를 포함해야 합니다.
- 구문 분석 규칙: 디코딩된 UPID에는 여러 값에 대한 콜론 구분 기호가 포함될 수 있습니다. 템플릿 변수와 디코딩된 UPID 토큰의 수는 같아야 합니다.

- 잘못된 형식: 값(예: :: 또는 :46175218::4053)이 없는 이중 콜론은 구문 분석 실패를 일으키므로 사용하지 마세요.

MediaTailor 광고 모니터링 및 알림 건너뛰기

선제적 모니터링을 사용하면 수익에 상당한 영향을 미치기 전에 광고 건너뛰기 문제를 감지하고 해결할 수 있습니다.는 광고 삽입 성능을 효과적으로 모니터링할 수 있는 포괄적인 지표와 로깅 기능을 AWS Elemental MediaTailor 제공합니다. 이 가이드에서는 광고 건너뛰기 문제에 대한 효과적인 모니터링을 설정하는 방법을 설명합니다.

모니터링할 주요 CloudWatch 지표

이러한 주요 MediaTailor 지표에 대한 CloudWatch 경보를 설정합니다.

- AdDecisionServer.Ads.Skipped - 건너뛴 광고 수
- AdDecisionServer.Timeouts - ADS 제한 시간 수
- Avail.FilledDuration - 채워진 광고 중단 기간
- Avail.SlateOnly - 슬레이트로만 채워진 광고 시간 수

고급 CloudWatch Logs Insights 쿼리

자세한 문제 해결을 위해 다음과 같은 특수 쿼리를 사용합니다.

포괄적인 세션 분석

특정 세션의 광고 삽입 동작에 대한 자세한 분석:

```
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
```

크리에이티브 IDs 찾기

FILLED_AVAIL 이벤트에서 크리에이티브 IDs 식별하려면:

```
fields @timestamp, sessionId, eventType
| filter sessionId like /sessionId/ and eventType!='BEACON_FIRED'
| sort @timestamp desc
```

Note

를 조사 중인 실제 세션 IDsessionId로 바꿉니다.

MediaTailor 광고 건너뛰기 방지 모범 사례

이러한 모범 사례를 구현하면 광고 건너뛰기 문제가 발생하기 전에 방지하여 광고 삽입 성능과 수익 보호를 개선하는 데 도움이 됩니다 AWS Elemental MediaTailor. 이러한 사전 예방 조치는 광고 건너뛰기의 가장 일반적인 원인을 해결하고 일관된 광고 전송을 유지하는 데 도움이 됩니다.

사전 예방 조치

- 광고 미리 가져오기 구현: MediaTailor의 미리 가져오기 기능을 사용하여 재생 전에 광고가 트랜스코딩되도록 합니다. 구현 세부 정보는 섹션을 참조 [the section called “광고 미리 가져오기”](#) 하세요.
- 일관된 크리에이티브 IDs: 광고 결정 서버가 세션 간에 동일한 광고 콘텐츠에 대해 일관된 크리에이티브 IDs를 사용하는지 확인합니다.
- 적절한 기간 형식 확인: ISO 8601 형식 대신 EXT-X-CUE-OUT 기간 파라미터에 정수 값 사용
- VOD 최적화 구성: 서버 로드를 줄이기 위해 여러 광고 중단이 있는 VOD 스트림에 대한 최대 동시 ADS 요청 설정을 설정합니다. 자세한 내용은 [고급 설정](#) 섹션을 참조하세요.
- ADS 성능 최적화: 광고 결정 서버가 빠르게 응답하고 최대 트래픽 볼륨을 처리하도록 구성

구현 지침

- 적절한 오류 처리 구현: 광고를 삽입할 수 없을 때 광고 시간을 채우도록 슬레이트 콘텐츠를 구성합니다.
- 철저한 테스트: 다양한 디바이스 및 네트워크 조건에서 광고 삽입 워크플로 검증
- 폴백 전략 구현: 기본 광고 소스가 실패할 때 백업 광고 소스 또는 기본 광고 구성
- 트랜스코딩 패턴 모니터링: CloudWatch 로그에서 트랜스코딩 효율성을 모니터링하고 트랜스코딩 문제를 나타내는 패턴을 발견하면 [AWS Support](#)에 문의하세요.

MediaTailor 광고 건너뛰기 참조 가이드

이 섹션에서는 광고 건너뛰기 이유에 대한 포괄적인 참조 정보와 관련 설명서 링크를 제공합니다 AWS Elemental MediaTailor. 이 참조 가이드를 사용하여 다양한 건너뛰기 이유의 특정 의미를 이해하고 추가 문제 해결 리소스를 찾을 수 있습니다.

광고 건너뛰기 이유 참조 완료

MediaTailor는 로그 그룹의 FILLED_AVAIL 이벤트 로그 메시지에서 광고를 건너뛰는 특정 이유를 MediaTailor/AdDecisionServerInteractions 기록합니다.

광고 건너뛰기 이유 완료

사유 건너뛰기	설명
NEW_CREATIVE	광고가 아직 트랜스코딩되지 않았습니다. 이는 MediaTailor가 삽입 전에 트랜스코딩이 필요한 새 광고 크리에이티브를 발견할 때 발생합니다.
PROFILE_NOT_FOUND	세션의 구성과 연결된 MediaConvert 트랜스코딩 프로파일이 존재하지 않아 광고 준비를 방지합니다.
TRANSCODE_ERROR	광고 트랜스코딩 프로세스에서 오류가 발생하여 완료하지 못했습니다.
TRANSCODE_IN_PROGRESS	광고 트랜스코드가 아직 진행 중이며 아직 삽입할 준비가 되지 않았습니다.
INTERNAL_ERROR	광고를 처리하는 동안 내부 MediaTailor 오류가 발생하여 삽입이 방지되었습니다.
AVAIL_DURATION_EXCEEDED	광고가 광고 종단의 남은 기간 내에 맞지 않습니다.
LEFTOVER_AVAIL_EXCEEDED_THRESHOLD	삽입되었을 수 있는 모든 광고의 누적 기간이 세션에 대해 구성된 개인화 임계값을 충족하지 않습니다.
VAST_PARSING_ERROR	광고 결정 서버의 VAST 응답에 오류가 있거나 형식이 잘못되었습니다.

사유 건너뛰기	설명
ADS_TIMEOUT	광고 결정 서버가 구성된 제한 시간 내에 응답하지 않았습니다.
MEDIA_FILE_UNAVAILABLE	VAST 응답에 지정된 광고 미디어 파일에 액세스할 수 없습니다.
SESSION_INITIALIZATION_FAILED	종종 잘못된 세션 변수로 인해 MediaTailor 세션이 제대로 초기화되지 않았습니다.
EARLY_CUE_IN	광고 중단은 초기 신호로 인해 예상보다 일찍 종료되어 광고가 완전히 삽입되지 않습니다.
NO_VARIANT_MATCH	광고 크리에이티브에는 콘텐츠 스트림의 인코딩 파라미터(비트레이트, 해상도, 코덱)와 일치하는 변형이 없습니다.
NO_MODEL_CREATIVE_MATCH	광고 크리에이티브가 현재 재생 구성에 대한 예상 모델 또는 형식 요구 사항과 일치하지 않습니다.
REJECTED_REPLICA_VAST	복제본 또는 중복 콘텐츠 감지 정책으로 인해 VAST 응답이 거부되었습니다.
INVALID_VAST_WRAPPER_AD	VAST 래퍼 광고에는 광고 삽입 성공을 방해하는 유효하지 않거나 잘못된 래퍼 요소가 포함되어 있습니다.
IMPORT_ERROR	광고 가져오기 프로세스 중에 오류가 발생하여 광고가 삽입 처리되지 않습니다.
IMPORT_IN_PROGRESS	광고 가져오기 프로세스가 현재 진행 중이며 아직 완료되지 않았습니다.

관련 리소스

광고 건너뛰기 문제 해결에 대한 자세한 내용은 다음 관련 주제를 참조하세요.

- [the section called “동적 광고 변수”](#) - MediaTailor의 동적 광고 변수에 대한 포괄적인 가이드
- [the section called “광고 미리 가져오기”](#) - 트랜스코딩 관련 건너뛰기를 방지하기 위해 광고 미리 가져오기를 구현하는 방법

- [the section called “CloudWatch 지표를 사용한 모니터링”](#) - CloudWatch 지표를 사용하여 MediaTailor 모니터링
- [the section called “로그 보기”](#) - MediaTailor 로그를 보고 분석하는 방법
- [the section called “이벤트 흐름 문제 해결”](#) - 광고 삽입 이벤트 흐름 이해

AWS Elemental MediaTailor 리소스

다음 표에는 AWS Elemental MediaTailor을(를) 사용할 때 참조할 수 있는 관련 리소스가 나와 있습니다.

Resource	설명
SCTE 표준: SCTE 35	SCTE35에 대한 SCTE 표준 문서입니다.
클래스 및 워크숍	AWS 기술을 연마하고 실용적인 경험을 얻는 데 도움이 되는 역할 기반 및 특수 과정과 자기 주도형 실습에 대한 링크입니다.
AWS 개발자 도구	AWS 애플리케이션 개발 및 관리를 위한 개발자 도구, SDKs, IDE 도구 키트 및 명령줄 도구에 대한 링크입니다.
AWS 백서	AWS Solutions Architects 또는 기타 기술 전문가가 작성한 아키텍처, 보안 및 경제와 같은 주제를 다루는 포괄적인 기술 AWS 백서 목록 링크입니다.
AWS 지원 센터	AWS 지원 사례를 생성하고 관리하기 위한 허브입니다. 포럼, 기술 FAQs, 서비스 상태 및 AWS Trusted Advisor와 같은 기타 유용한 리소스에 대한 링크도 포함되어 있습니다.
AWS 지원	클라우드에서 애플리케이션을 구축하고 실행하는 데 도움이 되는 one-on-one 빠른 응답 지원 채널인 AWS Support에 대한 정보를 제공하는 기본 웹 페이지입니다.
문의처	AWS 결제, 계정, 이벤트, 남용 및 기타 문제에 대한 문의를 위한 중앙 연락 창구입니다.
AWS 사이트 용어	저작권 및 상표, 사용자 계정, 라이선스 및 사이트 액세스와 기타 주제에 대한 상세한 정보입니다.

에 대한 문서 기록 AWS Elemental MediaTailor

다음 표에서는 이 설명서에서 변경된 중요 사항에 대해 설명합니다.

변경 사항	설명	날짜
새 분석 대시보드 페이지	재생 구성을 실행하는 AWS 리전에서 수익화 성능 및 비컨 실행 성능을 시각화하는 MediaTailor 콘솔 페이지인 MediaTailor MediaTailor 분석 대시보드에 대한 설명서가 추가되었습니다.	2026년 8월 29일
새로운 도메인별 비컨 지표	Avail.Impression.Fired , Avail.Impression.Retried , Avail.Impression.Recovered , Avail.Complete.Fired , Avail.Complete.Retried 및 Avail.Complete.Recovered 지표와 AdTrackingDomain 차원이 추가되었습니다.	2026년 8월 11일
ADS 제한 시간, 개인화 시간 예산 및 동시성 설정 추가	새로운 미리 가져오기 제한 시간 설정을 포함하여 구성 가능한 광고 결정 서버(ADS) 요청 제한 시간, 개인화 시간 예산 및 재생 구성의 동시성에 대한 설명서가 추가되었습니다.	2026년 7월 23일
건너뛴 광고 이유에 대한 계정 수준 차원	SkippedReason 지표에 대한 계정 수준(차원 없음) 배출량이 추가되었습니다.	2026년 7월 21일

새로운 지표 및 계정 수준 차원

Avail.Complete 지표를 추가했습니다. , Requests, , , AdsBilled , Avail.Impression Avail.Complete Avail.FilledDuration Avail.Duration 및 AdDecisionServer.Ads 지표에 대한 계정 수준(차원 없음) 배출량이 추가되었습니다.

2026년 6월 30일

Google의 광고 플랫폼을 사용한 자동 서버 측 인증

MediaTailor가 이제 Google Ad Manager(GAM), Google Campaign Manager(GCM), Google Display & Video 360(DV360)을 사용하여 server-to-server 연결을 자동으로 인증함을 반영하도록 장을 업데이트했습니다. 고객 작업 또는 지원 티켓은 필요하지 않습니다. 서버 측 섹션을 업데이트하여 MediaTailor가 적용하는 URI 기반 규칙을 설명하고 수동 지원 티켓 절차를 제거했습니다.

2026년 4월 15일

TPS 트래픽 셰이핑 추가

TPS를 기반으로 ADS에 대한 요청을 제한하는 방법에 대한 정보를 포함하도록 섹션을 업데이트했습니다.

2025년 10월 2일

prefetchScheduleName 추가됨

ADS 로그prefetchScheduleName 에 대한 정보가 추가되었습니다.

2025년 9월 24일

[문제 해결 안내서를 건너뛰는 광고 재구성](#)

각 섹션을 수행할 특정 작업에 초점을 맞추도록 문제 해결 안내서를 재구성했습니다.

2025년 8월 26일

[쿼리 파라미터 처리 설명서 추가](#)

MediaTailor가 MediaPackage를 사용한 시간 이동 보기와 같은 오리진별 기능에 대한 쿼리 파라미터를 처리하는 방법을 설명하는 섹션이 추가되었습니다.

2025년 8월 26일

[SGAI VOD 구성 가이드](#)

video-on-demand 콘텐츠가 포함된 서버 가이드 광고 삽입에 대한 구성 지침이 추가되었습니다.

2025년 8월 7일

[통합 파라미터 문제 해결 지침](#)

문자 제한 오류, 길이 제한 문제, 구성 별칭 문제, 파라미터 처리 흐름 디버깅 및 보안 모범 사례를 다루는 포괄적인 문제 해결 섹션이 추가되었습니다.

2025년 8월 4일

[향상된 파라미터 형식 설명서](#)

매니페스트 쿼리 파라미터 및 ADS 파라미터에 대한 포괄적인 문자 제한, URL 인코딩 지원 및 길이 제한이 추가되었습니다.

2025년 8월 1일

[포괄적인 참조가 포함된 향상된 파라미터 설명서](#)

AWS 가이드 분석을 기반으로 포괄적인 파라미터 문자 제한, 길이 제한, URL 인코딩 지원, 구성 별칭 설명서 및 세션 초기화 방법이 추가되었습니다.

2025년 8월 1일

[향상된 매니페스트 쿼리 파라미터 설명서](#)

AWS 가이드 분석을 기반으로 포괄적인 파라미터 배치 동작, 문자 제한, URL 인코딩 지원 및 프로토콜별 차이점이 추가되었습니다.

2025년 8월 1일

[향상된 도메인 변수 설명서](#)

AWS 가이드 분석을 기반으로 포괄적인 구성 별칭 설명서, 지원되는 필드 및 도메인 수준 파라미터 규칙이 추가되었습니다.

2025년 8월 1일

[이벤트 흐름 문제 해결 지침 추가](#)

MediaTailor 이벤트 흐름 분석을 사용하여 광고 삽입 문제를 진단하고 해결하기 위한 실용적인 지침을 제공하는 새로운 섹션입니다.

2025년 7월 31일

[광고 삽입 이벤트 흐름 설명서 추가](#)

사용자가 워크플로를 이해하는데 도움이 되도록 매니페스트 개인화 중 MediaTailor 이벤트의 시간 흐름에 대해 설명하는 새 섹션입니다.

2025년 7월 23일

[향상된 클라이언트 측 추적 워크플로 설명서](#)

자세한 시퀀스 다이어그램 및 step-by-step 프로세스 설명과 함께 포괄적인 클라이언트 측 보고 워크플로 설명서가 추가되었습니다.

2025년 7월 16일

[CDN 통합 설명서](#)

MediaTailor를 사용하여 CDN을 설정하는 방법에 대한 보다 포괄적인 정보가 추가되었습니다.

2025년 6월 30일

채널 어셈블리에 CDN 통합 추가	CDN을 채널 어셈블리와 통합하여 콘텐츠 전송을 최적화하기 위한 설명서가 추가되었습니다.	2025년 5월 16일
HLS 재생 목록 유형	HLS 다변량 및 미디어 재생 목록에 대한 정의가 추가되었습니다.	2025년 5월 9일
DASH 매니페스트 유형	DASH 매니페스트 구성 요소에 대한 정의가 추가되었습니다.	2025년 5월 9일
HLS 재생 목록 유형 정보 추가	예제의 컨텍스트를 제공하기 위해 다변량 및 변형 재생 목록에 대한 설명을 추가했습니다.	2025년 5월 9일
DASH 매니페스트 유형 정보 추가	예제의 컨텍스트를 제공하기 위해 MPD 구조 및 구성 요소에 대한 설명이 추가되었습니다.	2025년 5월 9일
반복되는 미리 가져오기 추가	반복되는 미리 가져오기 일정에 대한 정보를 포함하도록 섹션을 업데이트했습니다.	2025년 4월 28일
구성별 로그 필터링 정보 추가	MediaTailor가 구성에 대해 내보내는 로그 필터링에 대한 정보가 추가되었습니다.	2025년 3월 31일
벤딩된 로그 정보 추가	MediaTailor에서 내보내는 로그를 게시하기 위해 벤딩된 로그를 사용하는 방법에 대한 새 단원이 추가되었습니다.	2025년 2월 4일
사전 조정된 광고 개요 추가	사전 조정된 광고를 사용할 때 MediaTailor가 광고 삽입을 관리하는 방법을 설명하는 새 섹션이 추가되었습니다.	2025년 1월 30일

광고 컨디셔닝 추가	스트리밍 미디어 파일 조정 설정이 추가되었습니다.	2025년 1월 30일
SGAI 라이브 구성 가이드	라이브 콘텐츠가 포함된 서버 가이드 광고 삽입에 대한 구성 지침이 추가되었습니다.	2025년 1월 15일
로그 유형 추가	MediaTailor에서 내보내는 로그 유형을 나열하는 새 섹션이 추가되었습니다.	2025년 1월 15일
로그 필터링 정보 추가	MediaTailor가 내보내는 로그 필터링에 대한 새 섹션이 추가되었습니다.	2025년 1월 15일
지원되는 쿼리 파라미터 형식 추가	매니페스트 쿼리 파라미터 및 ADS 쿼리 파라미터 형식 지정에 대한 섹션이 추가되었습니다.	2025년 1월 2일
Google Ad Manager에 대한 통합 정보	서버 측 및 클라이언트 측에서 MediaTailor를 Google Ad Manager와 통합하는 방법에 대한 섹션을 추가했습니다.	2024년 11월 25일
업데이트된 변수	가 breakabilityStartTime 로 변경되었습니다availabilityStartTime .	2024년 5월 6일
개인화 세부 정보 추가	Insertion Mode 개인화 세부 정보가 추가되었습니다.	2024년 5월 6일
프로그램 규칙	프로그램 규칙에 새 콘텐츠가 추가되었습니다.	2024년 4월 25일
채널 생성에 대한 업데이트된 콘텐츠	채널을 생성할 때 프로그램 규칙에 대한 정보가 추가되었습니다.	2024년 4월 20일

프로그램 추가에 대한 내용 업데이트	프로그램을 추가할 때 프로그램 규칙에 대한 정보가 추가되었습니다.	2024년 4월 20일
워크플로 모니터	AWS 미디어 서비스를 분석하고 해당 서비스 간에 미디어 워크플로의 시그널 맵, 시각화를 생성합니다. 시그널 맵을 활용하여 CloudWatch, EventBridge 및 CloudFormation을 통해 모니터링 경고 및 알림을 생성합니다.	2024년 4월 11일
AlternateMedia 및 As Run Log	AlternateMedia가 실행 로그에 미치는 영향에 대한 참고 사항이 추가되었습니다.	2024년 2월 28일
매니페스트 설정 업데이트	선택한 광고 마크업 유형에 따라 패스스루 태그에 대한 정보가 추가되었습니다.	2023년 11월 28일
광고 시간에 대한 SCTE-35 메시지	Daterange 대 Scte35 Enhanced 광고 마크업 유형에 삽입된 SCTE-35 태그에 대한 정보가 추가되었습니다.	2023년 11월 28일
Enhanced Scte35 광고 마크업 유형의 키값 페어	MediaTailor가 Enhanced Scte35 광고 마크업 유형에 대해 제출된 키값 페어를 처리하는 방법에 대한 정보를 추가했습니다.	2023년 11월 28일
VOD 소스 광고 기회	이제 MediaTailor가 VOD 소스에 대한 광고 기회를 자동으로 감지할 수 있습니다.	2023년 10월 6일

새로운 SigV4 인증 유형 자동 감지	MediaTailor는 이제 AUTODETECT_SIGV4 액세스 유형을 지원합니다.	2023년 8월 18일
클라이언트 측 추적 콘텐츠 업데이트	추가 정보를 포함하도록 클라이언트 측 추적 콘텐츠를 업데이트했습니다.	2023년 8월 12일
라이브 소스를 사용하도록 MediaTailor 및 MediaPackage 워크플로 설정	MediaTailor 및 AWS Elemental MediaPackage 워크플로가 라이브 소스를 사용할 때의 설정, 일반 요구 사항 및 동작에 대한 정보가 추가되었습니다.	2023년 5월 24일
매니페스트 쿼리 파라미터 설명서	매니페스트 쿼리 파라미터를 설명하는 섹션을 추가했습니다.	2023년 4월 26일
오버레이 광고 설명서	오버레이 광고를 설명하는 섹션을 추가했습니다.	2023년 4월 24일
매니페스트 설명서의 광고 ID 장식	매니페스트의 광고 ID 장식을 설명하는 섹션을 추가했습니다.	2023년 4월 24일
AFTER_LIVE_EDGE 억제 모드 추가	AFTER_LIVE_EDGE 억제 모드 외에도 광고 억제 BEFORE_LIVE_EDGE 모드를 사용할 수 있습니다.	2023년 2월 21일
New As Run 로그	실행 중 로그의 새 주제입니다.	2023년 1월 19일
IAM 모범 사례 업데이트	IAM 모범 사례에 따라 가이드가 업데이트되었습니다. 자세한 내용은 IAM의 보안 모범 사례 를 참조하세요.	2022년 12월 27일

IAM 모범 사례 업데이트	IAM 모범 사례에 따라 가이드가 업데이트되었습니다. 자세한 내용은 IAM의 보안 모범 사례 를 참조하세요.	2022년 12월 27일
새로운 세분화 UPID 프라이빗 데이터 변수	MediaTailor는 이제 콜론으로 구분된 프라이빗 데이터를 분할 UPID 유형 12(MPU)에서 분할 <code>scte.segmentation_upid.private_data.{index}</code> 하는 추가 세션 변수를 지원합니다.	2022년 12월 22일
업데이트된 할당량 콘텐츠	할당량 정보를 업데이트하고 재구성했습니다.	2022년 9월 13일
채널 어셈블리 할당량 추가 및 수정	MediaTailor 채널 어셈블리 서비스에 라이브 소스, 세그먼트 전송 구성, 매니페스트 요청 및 채널 트랜잭션에 대한 할당량이 추가되었습니다.	2022년 9월 13일
채널 어셈블리 할당량 추가 및 수정	MediaTailor 채널 어셈블리 서비스에 로깅, 채널, 라이브 소스, 프로그램, 소스 위치 및 채널 정책에 대한 할당량이 추가되었습니다.	2022년 9월 11일
새 채널 어셈블리 알림 테이블	이제 채널 어셈블리 알림을 설명하는 테이블을 볼 수 있습니다.	2022년 9월 1일
새로운 Amazon CloudWatch 지표	새로운 CloudWatch 지표가 추가되었습니다.	2022년 6월 26일
광고 호출 주제	VAST 응답에 대한 사전 가져오기 지원 링크가 추가되었습니다.	2022년 5월 25일

[새 ADS 요청 변수](#)

MediaTailor는 이제 ADS 요청에서 `scte.segmentation_type_id` , , `scte.avails_expected` , , `scte.delivery_not_restricted_flag` , `scte.segment_num` , `scte.sub_segment_num` `scte.segments_expected` , `scte.sub_segments_expected` , 및 등의 추가 SCTE-35 변수를 지원합니다. `scte.device_restrictions` `scte.no_regional_blackout_flag` `scte.archive_allowed_flag` `scte.segmentation_event_id` .

2022년 4월 27일

[새로운 IAM 관리형 정책 정책 주제](#)

MediaTailor에 대한 두 가지 새로운 관리형 정책이 추가되었습니다.

2021년 11월 24일

[새 AWSElementalMediaTailorReadOnly 관리형 정책](#)

MediaTailor 리소스에 대한 읽기 전용 액세스를 허용하는 권한을 부여하는 새로운 AWS 관리형 정책이 추가되었습니다.

2021년 11월 10일

[새 AWSElementalMediaTailorFullAccess 관리형 정책](#)

MediaTailor 리소스에 대한 전체 액세스를 허용하는 새로운 AWS 관리형 정책이 추가되었습니다.

2021년 11월 10일

새로운 혼동된 대리자 주제	혼동된 대리자 문제를 방지하는 방법을 설명하는 주제가 추가되었습니다.	2021년 11월 4일
광고 주제 미리 가져오기	이제 MediaTailor는 광고가 발생하기 전에 광고 시간에 대한 광고를 미리 가져올 수 있습니다.	2021년 10월 12일
재생 구성에 대한 로깅 구성 설정 추가	로깅 구성 설정을 사용하여 재생 구성 로그와 관련된 설정을 제어합니다.	2021년 9월 28일
광고 시간에 대한 SCTE-35 메시지	segmentation_descriptor 메시지와 함께 time_signal 메시지를 사용하는 방법에 대한 정보가 추가되었습니다.	2021년 9월 1일
새로운 선형 재생 모드	새로운 선형 재생 모드가 추가되었습니다.	2021년 9월 1일
새로운 절대 전환 유형	선형 채널에서 프로그램의 벽시계 시작 시간을 설정하는 데 사용할 수 있는 절대 전환 유형에 대한 지원이 추가되었습니다.	2021년 9월 1일
새 채널 어셈블리 알림 주제	이제 MediaTailor 알림을 사용하여 채널 어셈블리 리소스를 모니터링할 수 있습니다. 채널 어셈블리 리소스에 문제 또는 잠재적 문제가 발생하면 MediaTailor가 알림을 생성합니다.	2021년 7월 14일

채널 송신 요청에 대한 채널 어셈블리 할당량 수정	MediaTailor 채널 어셈블리 서비스의 채널 송신 요청에 대한 할당량을 수정했습니다.	2021년 6월 29일
새 소스 위치 인증 유형	MediaTailor는 이제 Secrets Manager 액세스 토큰 인증을 지원합니다.	2021년 6월 16일
새 티어 정보	각 티어가 지원하는 모드 및 소스 유형에 대한 정보가 추가되었습니다.	2021년 6월 13일
새 소스 유형 정보	표준 채널의 경우 프로그램이 재생하는 소스 유형에 대한 정보가 추가되었습니다.	2021년 6월 13일
새로운 MediaTailor 라이브 소스 설명서	라이브 소스는 소스 위치에 추가하는 단일 라이브 스트림을 나타냅니다. 채널을 생성한 후 소스 위치에 라이브 소스를 추가하고 각 라이브 소스를 프로그램에 연결할 수 있습니다.	2021년 6월 13일
추가 UPID 유형 지원	MediaTailor는 이제 ADS 정보(0xE) 및 사용자 정의(0x1) 분할 UPID 유형을 지원합니다.	2021년 4월 15일
새로운 분할 UPID 동적 변수	scte.segmentation_upid.assetId , scte.segmentation_upid.cueData.key 및 의 세 가지 새로운 동적 변수가 있습니다scte.segmentation_upid.cueData.value . 이러한 변수는 포드버스터 워크플로의 MPU 분할 UPID 유형(0xC)과 함께 사용됩니다.	2021년 4월 15일

새 채널 어셈블리 서비스 설명	새 채널 어셈블리 서비스에 대한 정보가 추가되었습니다.	2021년 3월 11일
새로운 MediaTailor 채널 어셈블리 서비스 설명서	채널 어셈블리는 기존 온디맨드 비디오(VOD) 콘텐츠를 사용하여 선형 스트리밍 채널을 생성할 수 있는 새로운 매니페스트 전용 서비스입니다.	2021년 3월 11일
채널 어셈블리 할당량 추가	새로운 MediaTailor 채널 어셈블리 서비스에 대한 할당량이 추가되었습니다.	2021년 3월 11일
새 채널 어셈블리 용어	새 채널 어셈블리 서비스에 해당하는 용어가 추가되었습니다.	2021년 3월 10일
채널 어셈블리 태그 지정 지원	에서 채널 어셈블리 리소스의 태그 지정에 대한 지원이 추가되었습니다 AWS Elemental MediaTailor. 채널, SourceLocations 및 VodSources는 태그 지정을 지원합니다.	2021년 3월 9일
새로운 동적 변수 주제	MediaTailor는 이제 동적 도메인 변수를 지원합니다.	2021년 2월 25일
선택적 구성 별칭 설정 추가	도메인 변수와 함께 구성 별칭을 사용하여 세션 초기화 중에 도메인을 동적으로 구성합니다.	2021년 2월 25일
새로운 scte.segmentation_upid 동적 광고 변수	scte.segmentation_upid 세션 데이터 동적 광고 변수에 대한 지원이 추가되었습니다.	2020년 12월 5일

새로운 광고 마커 패스스루 주제	이제 HLS 매니페스트에 광고 마커 패스스루를 사용할 수 있습니다.	2020년 10월 29일
구성 고급 설정 업데이트	광고 마커 패스스루는 새로운 재생 구성 고급 설정입니다.	2020년 10월 14일
새 디버그 로그 모드	DEBUG 로그 모드에 대한 새 주제입니다.	2020년 8월 14일
범퍼의 EXT-X-CUE-OUT 기간 속성에 대한 설명	HLS의 경우 각 EXT-X-CUE-OUT 태그에 duration 속성이 필요하도록 범퍼 요구 사항을 업데이트했습니다.	2020년 8월 5일
세션 초기화 보고 모드 파라미터	세션 초기화 요청에 reportingMode 파라미터에 대한 지원이 추가되었습니다. 이제 세션을 초기화할 때 MediaTailor가 클라이언트 측 또는 서버 측 광고 추적을 수행할지 여부를 지정할 수 있습니다.	2020년 7월 30일
새 범퍼 주제	새 범퍼 주제 추가	2020년 7월 27일
DASH에 광고 억제 사용 가능	이제 DASH에서 광고 억제를 사용할 수 있습니다. 광고 억제 주제에서 “HLS 전용” 제한을 제거했습니다.	2020년 6월 3일
콘솔별 이름 업데이트	콘솔 UI의 최신 버전을 반영하도록 콘솔별 이름이 업데이트되었습니다.	2020년 5월 1일
새로운 avail.index 동적 광고 변수	새 avail.index 세션 데이터 동적 광고 변수에 대한 지원이 추가되었습니다.	2020년 3월 13일

새 AdVerifications 및 Extensions 요소	클라이언트측 보고의 경우 AdVerifications 및 Extensions 요소가 지원됩니다.	2020년 3월 10일
개인화 임계값 구성	선택적인 개인화 임계값 구성에 대한 지원이 추가되었습니다.	2020년 2월 14일
DASH VOD 매니페스트	여러 기간의 매니페스트 출력을 사용하여 오리진 서버의 온디맨드 비디오(VOD)에 대한 지원이 추가되었습니다.	2019년 12월 23일
트랜스코딩 프로파일 이름에 대한 콘솔 지원	구성에 트랜스코드 프로파일 이름에 대한 설명이 추가되었습니다.	2019년 12월 23일
한도 테이블 업데이트	ADS 리디렉션 및 ADS 시간 초과에 대한 제한이 업데이트되었습니다.	2019년 12월 18일
라이브 프리롤 동작 문서화	라이브 프리롤 광고의 작동 방식을 설명하는 프리롤 광고 삽입 섹션이 추가되었습니다 AWS Elemental MediaTailor.	2019년 11월 26일
라이브 프리롤 광고 지원	라이브 스트림 시작 부분에 프리롤 광고 삽입 지원이 추가되었습니다.	2019년 9월 11일
Amazon CloudWatch Logs 인사이트에서 ADS 로그 분석	AWS Elemental MediaTailor ADS 로그 및 CloudWatch Logs Insights를 사용하여 MediaTailor 세션을 분석하는 방법에 대한 정보가 추가되었습니다.	2019년 8월 13일

<u>새로운 보안 장</u>	범위를 개선하고 표준화하기 위해 보안 장이 추가되었습니다.	2019년 5월 23일
<u>DASH 단일 기간 매니페스트</u>	오리진 서버의 단일 기간 DASH 매니페스트와 여러 기간 매니페스트 출력에 대한 지원이 추가되었습니다.	2019년 4월 4일
<u>ADS URL에서 SCTE-35 UPIDs 지원</u>	ADS(광고 의사결정 서버) URL의 고유 프로그램 ID(UPID)를 포함하기 위한 추가된 지원입니다. 이를 통해 ADS가 프로그램 수준의 광고 타겟팅을 라이브 선형 스트림 내에 제공할 수 있습니다.	2019년 3월 28일
<u>클라이언트 측 보고에서 컴패니언 광고 지원</u>	클라이언트 측 보고의 경우 이제 AWS Elemental MediaTailor 추적 URL 응답에 컴패니언 광고 메타데이터가 포함됩니다.	2019년 3월 28일
<u>HLS 광고 마커 설명서</u>	지원되는 HLS 광고 마커를 설명하는 단원이 추가되었습니다.	2019년 3월 1일
<u>태그 지정 지원</u>	AWS Elemental MediaTailor의 구성 리소스 태그 지정에 대한 지원이 추가되었습니다. 태그를 지정하면 AWS 리소스를 식별 및 구성하고 리소스에 대한 액세스를 제어하며 AWS 비용을 추적할 수 있습니다.	2019년 2월 14일

AWS CloudTrail 로깅 정보 추가	CloudTrail을 사용하여 API에 작업을 로깅하는 방법에 대한 주제가 AWS Elemental MediaTailor 추가되었습니다.	2019년 2월 11일
재생 오류에 대한 섹션 추가	플레이어 또는 콘텐츠 전송 네트워크(CDN)의 요청에 대한 응답으로 재생 중에 MediaTailor에서 반환할 수 있는 오류에 대한 정보가 추가되었습니다.	2019년 2월 4일
DASH base64 인코딩 바이너리	<scte35:Signal> <scte35:Binary> 마커 내 매니페스트의 스포라이싱 정보를 base64로 인코딩된 이진수로 제공하는 작업에 대한 지원이 추가되었습니다.	2019년 1월 4일
DASH 시간 신호	<scte35:TimeSignal> 마커 내 매니페스트의 스포라이싱 정보 제공에 대한 지원이 추가되었습니다.	2018년 12월 5일
DASH 위치 지원	MPEG-DASH <Location> 태그에 대한 지원이 추가되었습니다.	2018년 12월 4일
DASH 지원	MPEG-DASH 매니페스트에 대한 지원이 추가되었습니다.	2018년 11월 14일
한도 테이블 업데이트	업데이트된 구성 및 매니페스트 크기의 제한.	2018년 10월 13일

신규 및 업데이트된 지표	광고 의사결정 서버(ADS) 및 오리진 제한 시간에 대한 지표가 추가되었고 시간 초과된 응답을 포함하도록 ADS 및 오리진 오류 정의가 업데이트되었습니다.	2018년 10월 13일
서버 측 및 클라이언트 측 광고 삽입 사용 사례에 대한 설명서 적용 범위 개선	서버 측 광고 삽입과 클라이언트 측 광고 삽입에 대한 동적 광고 변수의 사용을 포함하여 설명 및 예제를 확장합니다.	2018년 10월 1일
새로운 리전	PDX 및 FRA 리전에 대한 지원이 추가되었습니다.	2018년 7월 18일
VAST/VPAID	VAST 및 VPAID에 대한 정보를 추가했습니다.	2018년 3월 16일
CloudWatch	사용 가능한 CloudWatch 지표, 네임스페이스 및 차원에 대한 정보가 추가되었습니다.	2018년 3월 16일
새로운 리전	아시아 태평양(싱가포르), 아시아 태평양(시드니) 및 아시아 태평양(도쿄) 리전에 대한 지원이 추가되었습니다.	2018년 2월 8일
IAM 정책 정보	관련 IAM 정책 정보가 추가되었습니다 AWS Elemental MediaTailor. 제한된 권한을 가진 비관리자 역할을 생성하기 위한 지침을 추가했습니다.	2018년 1월 3일
첫 번째 릴리스	이 설명서의 최초 릴리스입니다.	2017년 11월 27일

 Note

- AWS 미디어 서비스는 애플리케이션의 용도 또는 서비스 사용 불가, 중단 또는 장애로 인해 사망, 개인 부상, 재산 피해 또는 환경 피해가 발생할 수 있는 생명 안전 운영, 탐색 또는 통신 시스템, 항공 교통 관제 또는 생명 지원 시스템과 같이 안전하지 않은 성능이 필요한 상황에서 사용하도록 설계되거나 의도되지 않았습니다.

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.