How to diagnose AWS Elemental MediaTailor ad insertion flows using debugging

Logging, metrics, and tracing

CONTENTS

Introduction

AWS Elemental MediaTailor allows you to serve targeted ads to viewers while maintaining broadcast quality in over-the-top (OTT) video applications. This document explains how to use the debugging feature of MediaTailor to look into an ad insertion session lifecycle to troubleshoot or visualize what is happening.

MediaTailor has always been deeply integrated into CloudWatch, publishing to the following Log Groups.

Note that permissions must be configured prior to receiving logs as documented here:
https://docs.aws.amazon.com/mediatailor/latest/ug/monitoring-permissions.html

Log group	Retention
MediaTailor/AdDecisionServerInteractions	Never expire
MediaTailor/ManifestService	Never expire
MediaTailor/TranscodeService	Never expire

Before we dive into debugging, let's go over the basics of MediaTailor logging.

Log group deep-dive

AdDecisionServerInteractions stores all logs related to the request and response from the upstream ad server including request, response, avail information, and beacon information. An example of an issue with an ad server response is:

```
Ad Decision Server request/response encountered problem while
creating time-based avail plan for VOD template; no ads will be
inserted
```

TranscodeService stores all logs related to the transcode jobs associated with ad creatives returned from the ad server. This particular log is useful during initial setup, as it exposes why an ad is not yet available for display. One example is:

```
Transcoded video not yet ready - transcoding is in progress.
Using empty set of ads.
```

ManifestService returns any errors that occurred during processing or retrieval of the manifest. MediaTailor pulls the template manifest from the origin server prior to manipulating it, so there are typically two sets of messages. Note that this log only reports errors found in the process, but more on that later as this can be overridden. In the case that MediaTailor cannot retrieve the manifest from the origin server, the following log would be written:

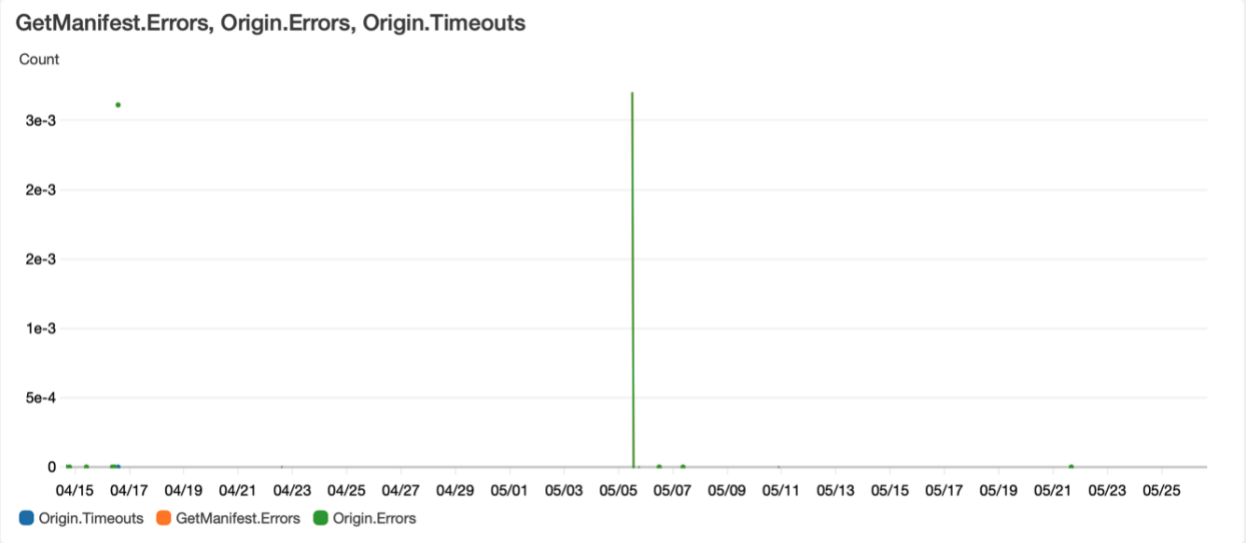
```
Manifest request timed out
Error obtaining template manifest from HTTP origin
```

More about MediaTailor metrics

Along with logs, the following metrics are supported within MediaTailor. Metrics are counters for important information that is useful to display on a dashboard or alert on if their values fall above or below a certain threshold.

Metric	Description
AdDecisionServer.Ads	The count of ads included in ad decision server (ADS) responses for the time period that you specified.
AdDecisionServer.Duration	The total duration, in milliseconds, of all ads that MediaTailor received from the ADS for the time period that you specified.
AdDecisionServer.Errors	The number of non-HTTP 200 status code responses, empty responses, and timed-out responses that MediaTailor received from the ADS in the time period that you specified.
AdDecisionServer.FillRate	The simple average of the rates at which the responses from the ADS filled the corresponding individual ad avails for the time period that you specified. To get the weighted average, calculate the AdDecisionServer.Duration as a percentage of the Avail.Duration. For more information about simple and weighted averages, see Simple and Weighted Averages.
AdDecisionServer.Timeouts	The number of timed-out requests to the ADS in the time period that you specified.
AdNotReady	The number of times that the ADS pointed at an ad that wasn't yet transcoded by the internal transcoder service in the time period that you specified. A high value for this metric might contribute to a low overall Avail.FillRate.
Avail.Duration	The total duration, in milliseconds, of all ad avails that MediaTailor encountered in the time period that you specified.
Avail.FilledDuration	The total duration, in milliseconds, of ad avail time that MediaTailor filled with ads in the time period that you specified.
Avail.FillRate	The simple average of the rates at which MediaTailor filled the individual ad avails for the time period that you specified. To get the weighted average, calculate the Avail.FilledDuration as a percentage of the Avail.Duration. For more information about simple and weighted averages, see Simple and Weighted Averages. The maximum Avail.FillRate that MediaTailor can attain is bounded by the AdDecisionServer.FillRate. If the Avail.FillRate is low, compare it to the AdDecisionServer.FillRate. If the AdDecisionServer.FillRate is low, your ADS might not be returning enough ads for the avail durations.
Origin.Errors	The number of non-HTTP 200 status code responses and timed-out responses that MediaTailor received from the origin server in the time period that you specified.
GetManifest.Errors	The number of errors received while MediaTailor was generating manifests in the time period that you specified.
Origin.Timeouts	The number of timed-out requests to the origin server in the time period that you specified.

You can use a variety of the metrics above to create dashboards to monitor MediaTailor for any abnormal events, such as Origin Errors below:



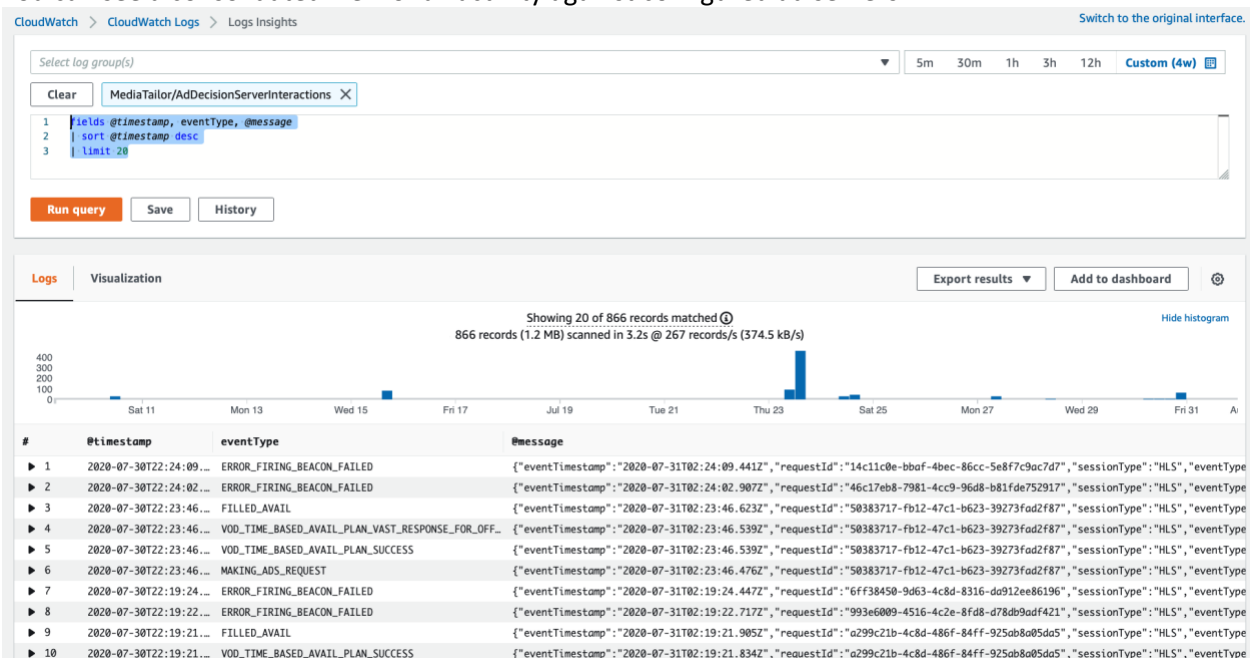
Log Insights Best Practice

One important note about logs - it is always best to pull them using Log Insights, as opposed to the raw CloudWatch streams. As MediaTailor is an elastic service, several instances could write to multiple log streams. Log Insights allows you to query all logs at once, simplifying the process of parsing the data.

Using the following query in Log Insights against the AdDecisionServerInteractions Log Group:

```
fields @timestamp, eventType, @message
| sort @timestamp desc
| limit 20
```

You can see a consolidated view of all activity against configured ad servers.



Automation

Custom metrics can be used to enhance the default metrics automatically published by MediaTailor. Custom alarms can be defined in CloudWatch based of customer requirements. CloudWatch APIs area available for CloudWatch to create automated monitoring workflows. The following are relevant links.

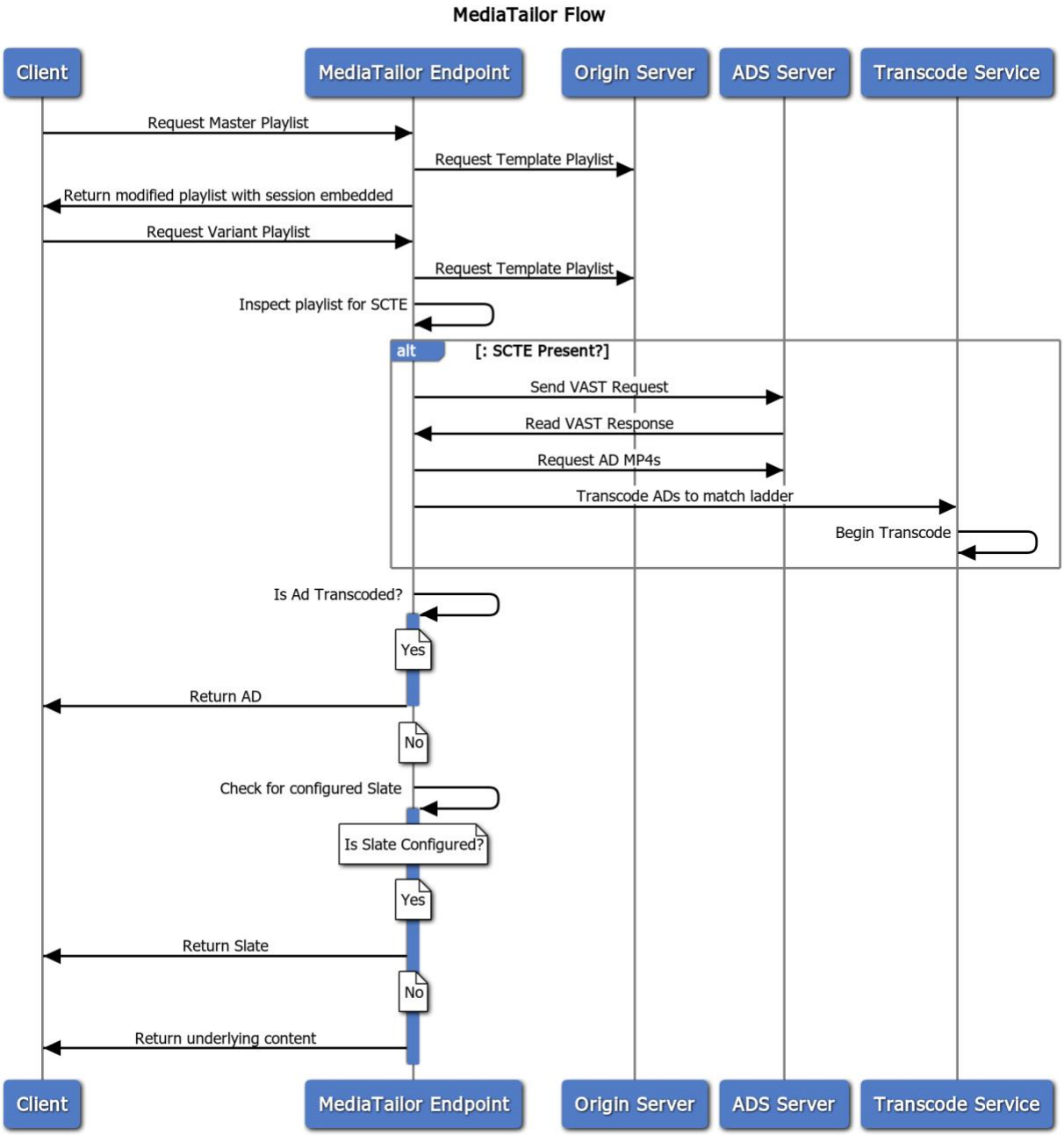
- CloudWatch APIs: https://docs.aws.amazon.com/AmazonCloudWatch/latest/APIReference/making-api-requests.html
- CloudWatch Publish Customer Metrics: https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/publishingMetrics.html
- CloudWatch Alarms: https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/AlarmThatSendsEmail.html
- CloudWatch Logs Insights Queries: https://docs.aws.amazon.com/mediatailor/latest/ug/monitor-cloudwatch-ads-logs.html

Debug Logging Intro

As you might have noticed, the ManifestService logs only report **errors** from the Origin or MediaTailor transformation service. What can be done to see logs of *all interactions* between the origin server and MediaTailor? If this data were available, it would be a huge help in initial setup of MediaTailor as well as during testing.

We certainly would not want this running at all times as the logs could become extremely large, so functionality is now available to selectively enable this logging on a per-session basis.

To better understand exactly what is being debugged, the following diagram illustrates the entire workflow:



More About MediaTailor Sessions

In order to individualize ads, MediaTailor (or the client) creates a session for each viewer. The session remains active during active viewing and ties the client to a particular set of content, keeping it unique per viewer. Understanding sessions is critical when debugging MediaTailor flows. This section will help you understand how sessions are created and used during playback.

Sessions are created either server-side or client-side (using the APIs) depending on whether you'd like to use server-side or client-side reporting.

To initialize a server-side reporting session, call MediaTailor so:

From the player, initialize a new MediaTailor playback session using a request in one of the following formats, according to your protocol:

- Example: HLS format

  GET <mediatailorURL>/v1/master/<hashed-account-id>/<origin-id>/<asset-id>?ads.<key-value-pairs-for-ads>&<key-value-pairs-for-origin-server>

- Example: DASH format

  GET <mediatailorURL>/v1/dash/<hashed-account-id>/<origin-id>/<asset-id>?ads.<key-value-pairs-for-ads>&<key-value-pairs-for-origin-server>

To initialize a client-side reporting session, call MediaTailor so:

```
POST <mediatailorURL>/v1/session/<hashed-account-id>/<origin-id>/<asset-id>
  {
    "adsParams": {
      "param1": "value1",
      "param2": "value2",
      "param3": "value3"
    }
    "originServerParam1": "originValue1",
    "originServerParam2": "originValue2"
  }
```

AWS Elemental MediaTailor responds to the request with two relative URLs, one for the manifest and one for the tracking endpoint:

Manifest – used to retrieve content manifests and ad segments

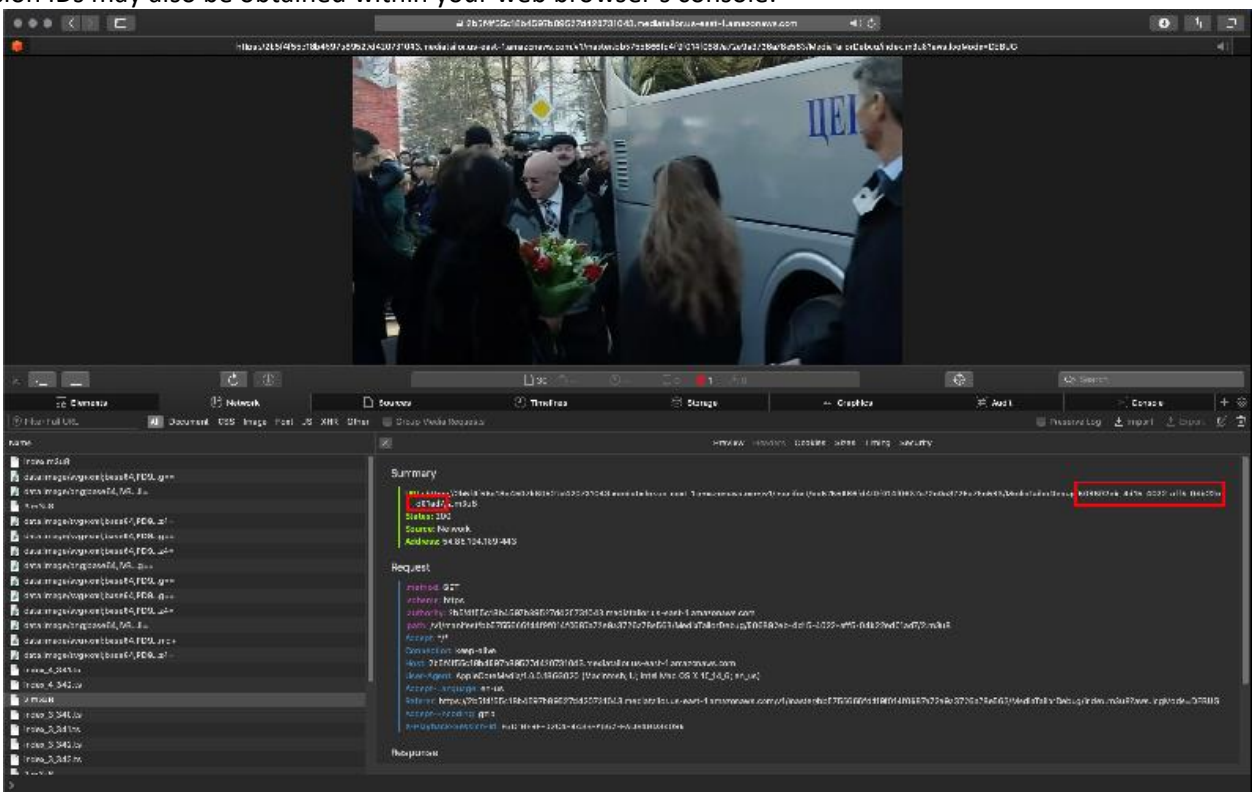
- Example: HLS

  /v1/master/<hashed-account-id>/<origin-id>/<asset-id>?aws.sessionId=<session>

- Example: DASH

  /v1/dash/<hashed-account-id>/<origin-id>/<asset-id>?aws.sessionId=<session>

Session IDs may also be obtained within your web browser's console:



Debug Logging Examples

As mentioned earlier, debugging is enabled on a per session basis. When you create a session, you must tag the request that debugging is desired. This is done in one of two ways.

Server-Side:

- Example: HLS format

  GET <mediatailorURL>/v1/master/<hashed-account-id>/<origin-id>/<asset-id>?ads.<key-value-pairs-for-ads>&<key-value-pairs-for-origin-server>**&aws.logMode=DEBUG**

- Example: DASH format

  GET <mediatailorURL>/v1/dash/<hashed-account-id>/<origin-id>/<asset-id>?ads.<key-value-pairs-for-ads>&<key-value-pairs-for-origin-server>**@aws.logMode=DEBUG**

To enable a server-side session with debugging, append aws.logMode=DEBUG to the manifest request's query string parameters. This is all that is needed to enable debugging.

Client-Side:

When initializing the session as seen above, add {'logMode': 'DEBUG'} to the payload in the POST request to the session endpoint.

Example:

```
curl -X POST -data '{'logMode': 'DEBUG'}' <mediatailorURL>/v1/session/<hashed-account-id>/<origin-id>/<asset-id>
```

Once debugging is enabled, we can now see the full lifecycle of a client's request including any sub-manifests delivered from MediaTailor.

Example

Normal Request:

```
curl "https://2b5f4f55c18b4597b89527d420731043.mediatailor.us-east-1.amazonaws.com/v1/master/bb5755666fd4f9f014f0687a72e9a3726a78e563/MediaTailorDebug/index.m3u8"

#EXTM3U
#EXT-X-VERSION:3
#EXT-X-INDEPENDENT-SEGMENTS
#EXT-X-STREAM-INF:CODECS="avc1.640028,mp4a.40.2",AVERAGE-BANDWIDTH=5711200,RESOLUTION=1920x1080,FRAME-RATE=30.0,BANDWIDTH=8968960
../../../manifest/bb5755666fd4f9f014f0687a72e9a3726a78e563/MediaTailorDebug/140cd622-a362-4646-9004-206e7c928900/0.m3u8
#EXT-X-STREAM-INF:CODECS="avc1.64001F,mp4a.40.2",AVERAGE-BANDWIDTH=3511200,RESOLUTION=1280x720,FRAME-RATE=30.0,BANDWIDTH=5448960
../../../manifest/bb5755666fd4f9f014f0687a72e9a3726a78e563/MediaTailorDebug/140cd622-a362-4646-9004-206e7c928900/1.m3u8
#EXT-X-STREAM-INF:CODECS="avc1.4D401E,mp4a.40.2",AVERAGE-BANDWIDTH=1790760,RESOLUTION=640x480,FRAME-RATE=30.0,BANDWIDTH=2752604
../../../manifest/bb5755666fd4f9f014f0687a72e9a3726a78e563/MediaTailorDebug/140cd622-a362-4646-9004-206e7c928900/2.m3u8
#EXT-X-STREAM-INF:CODECS="avc1.4D400D,mp4a.40.2",AVERAGE-BANDWIDTH=965742,RESOLUTION=320x240,FRAME-RATE=30.0,BANDWIDTH=1432591
../../../manifest/bb5755666fd4f9f014f0687a72e9a3726a78e563/MediaTailorDebug/140cd622-a362-4646-9004-206e7c928900/3.m3u8
```

Request with Debugging:

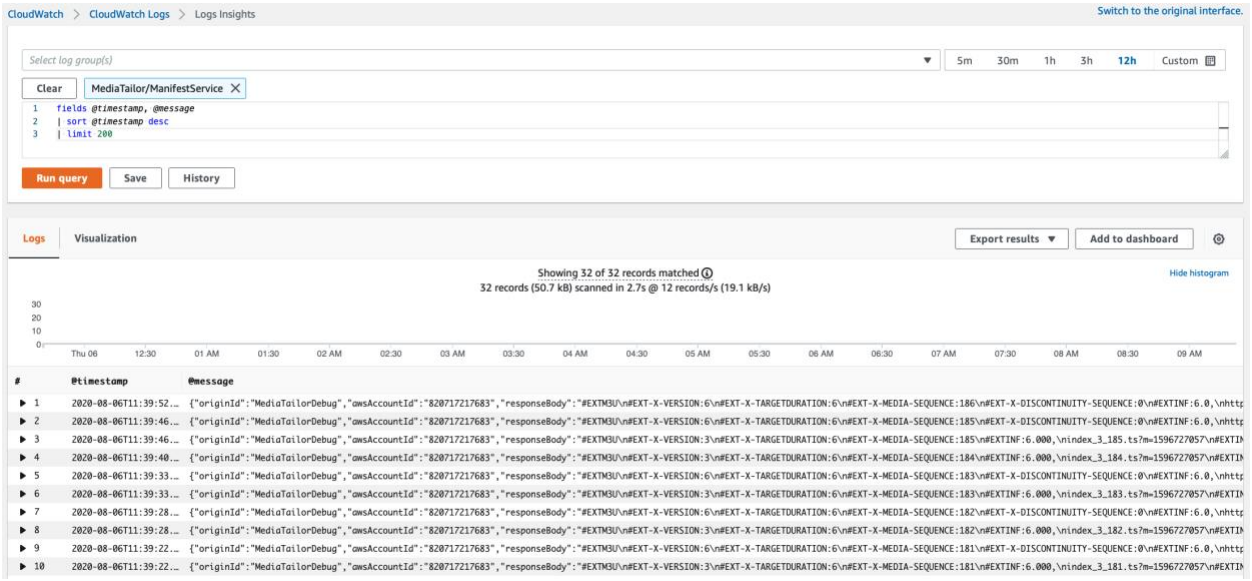
```
curl "https://2b5f4f55c18b4597b89527d420731043.mediatailor.us-east-1.amazonaws.com/v1/master/bb5755666fd4f9f014f0687a72e9a3726a78e563/MediaTailorDebug/index.m3u8?aws.logMode=DEBUG"

#EXTM3U
#EXT-X-VERSION:3
#EXT-X-INDEPENDENT-SEGMENTS
#EXT-X-STREAM-INF:CODECS="avc1.640028,mp4a.40.2",AVERAGE-BANDWIDTH=5711200,RESOLUTION=1920x1080,FRAME-RATE=30.0,BANDWIDTH=8968960
../../../manifest/bb5755666fd4f9f014f0687a72e9a3726a78e563/MediaTailorDebug/4878c173-3206-4998-bfe1-51ee778d5d7a/0.m3u8
#EXT-X-STREAM-INF:CODECS="avc1.64001F,mp4a.40.2",AVERAGE-BANDWIDTH=3511200,RESOLUTION=1280x720,FRAME-RATE=30.0,BANDWIDTH=5448960
../../../manifest/bb5755666fd4f9f014f0687a72e9a3726a78e563/MediaTailorDebug/4878c173-3206-4998-bfe1-51ee778d5d7a/1.m3u8
#EXT-X-STREAM-INF:CODECS="avc1.4D401E,mp4a.40.2",AVERAGE-BANDWIDTH=1790760,RESOLUTION=640x480,FRAME-RATE=30.0,BANDWIDTH=2752604
../../../manifest/bb5755666fd4f9f014f0687a72e9a3726a78e563/MediaTailorDebug/4878c173-3206-4998-bfe1-51ee778d5d7a/2.m3u8
#EXT-X-STREAM-INF:CODECS="avc1.4D400D,mp4a.40.2",AVERAGE-BANDWIDTH=965742,RESOLUTION=320x240,FRAME-RATE=30.0,BANDWIDTH=1432591
../../../manifest/bb5755666fd4f9f014f0687a72e9a3726a78e563/MediaTailorDebug/4878c173-3206-4998-bfe1-51ee778d5d7a/3.m3u8
```

Noting the manifest returned above, this client's session ID is included within the variant playlist URL. Note the session ID immediately before the playlist filename:

```
4878c173-3206-4998-bfe1-51ee778d5d7a
```

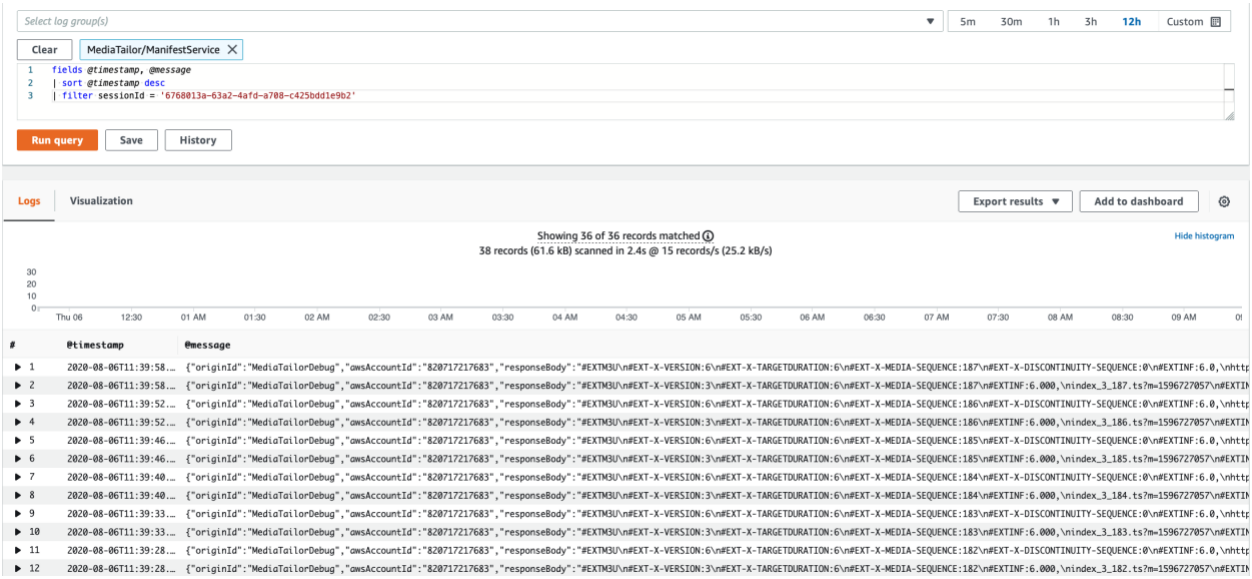
To see ALL debug logs available, the following query would return all sessions:

```
fields @timestamp, @message
| sort @timestamp desc
| limit 200 (Change limit based on preferences)
```



To see debug logs associated with a specific session, the following would return a single session:

```
fields @timestamp, @message
| sort @timestamp desc
| filter sessionId = '6768013a-63a2-4afd-a708-c425bdd1e9b2'
```



Example Origin Response:

```
fields @timestamp, @message
| sort @timestamp desc
| filter sessionId = '6768013a-63a2-4afd-a708-c425bdd1e9b2' and eventType ="ORIGIN_MANIFEST"
```

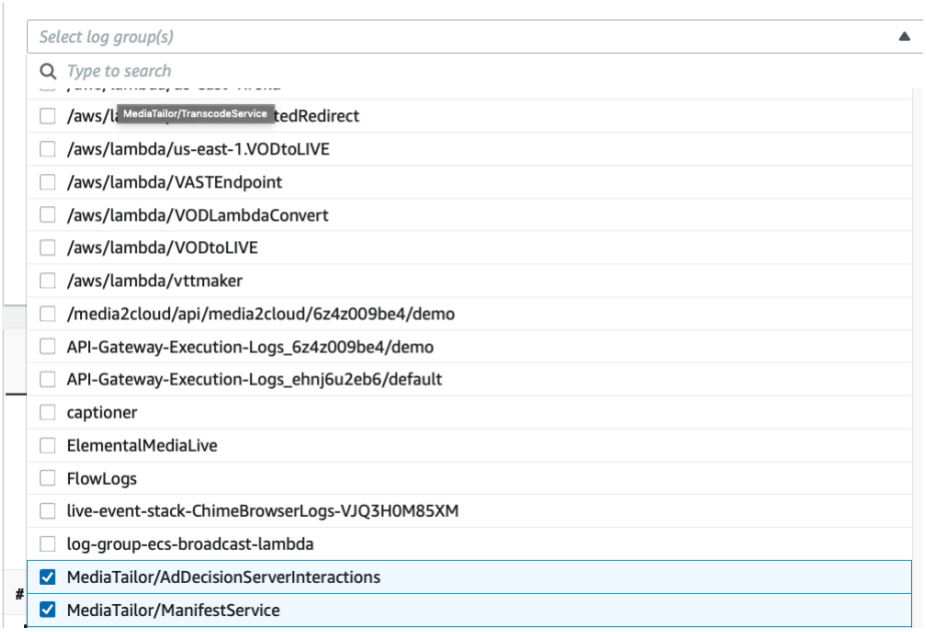
Field	Value
@ingestionTime	1596728321056
@log	820717217683:MediaTailor/ManifestService
@logStream	MediaTailorDebug-005ab4af79c5e5e02-31
@message	{"originId":"MediaTailorDebug","awsAccountId":"820717217683","responseBody":"#EXTM3U\n#EXT-X-VERSION:3\n#EXT-X-INDEPENDENT-SEGMENTS\n#EXT-X-STREAM-INF:BANDWIDTH=8968960,AVERAGE-BANDWIDTH=5711200,RESOLUTION=1920x1080,FRAME
@timestamp	1596728312001
awsAccountId	820717217683
customerId	bb5755666fd4f9f014f0687a72e9a3726a78e563
eventTimestamp	2020-08-06T15:38:32.001Z
eventType	ORIGIN_MANIFEST
mediaTailorPath	/v1/master/bb5755666fd4f9f014f0687a72e9a3726a78e563/MediaTailorDebug/index.m3u8?aws.logMode=DEBUG
originId	MediaTailorDebug
originRequestUrl	https://547f72e6652371c3.mediapackage.us-east-1.amazonaws.com/out/v1/eda3a5219a9e4d4ca750527b2b1835c8/index.m3u8
requestId	0fe480f0-8d38-42e7-bad5-32401b7b8dea
responseBody	#EXTM3U #EXT-X-VERSION:3 #EXT-X-INDEPENDENT-SEGMENTS #EXT-X-STREAM-INF:BANDWIDTH=8968960,AVERAGE-BANDWIDTH=5711200,RESOLUTION=1920x1080,FRAME-RATE=30.000,CODECS="avc1.640028,mp4a.40.2" index_1.m3u8 #EXT-X-STREAM-INF:BANDWIDTH=5448960,AVERAGE-BANDWIDTH=3511200,RESOLUTION=1280x720,FRAME-RATE=30.000,CODECS="avc1.64001F,mp4a.40.2" index_2.m3u8 #EXT-X-STREAM-INF:BANDWIDTH=2752604,AVERAGE-BANDWIDTH=1790760,RESOLUTION=640x480,FRAME-RATE=30.000,CODECS="avc1.4D401E,mp4a.40.2" index_3.m3u8 #EXT-X-STREAM-INF:BANDWIDTH=1432591,AVERAGE-BANDWIDTH=965742,RESOLUTION=320x240,FRAME-RATE=30.000,CODECS="avc1.4D400D,mp4a.40.2" index_4.m3u8
sessionId	6768013a-63a2-4afd-a708-c425bdd1e9b2
sessionType	HLS

Example MediaTailor Response:

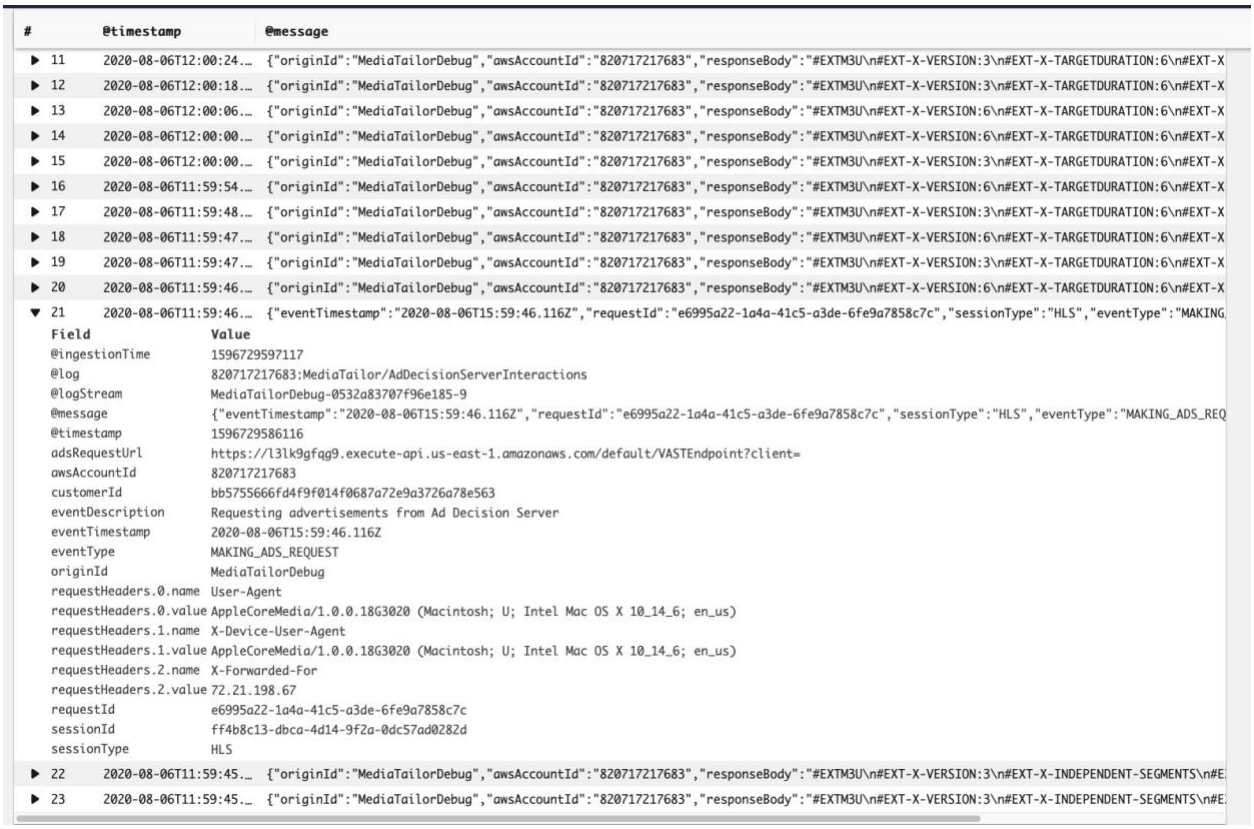
```
fields @timestamp, @message
| sort @timestamp desc
| filter sessionId = '6768013a-63a2-4afd-a708-c425bdd1e9b2' and eventType ='GENERATED_MANIFEST'
```

Field	Value
@ingestionTime	1596728319404
@log	820717217683:MediaTailor/ManifestService
@logStream	MediaTailorDebug-005ab4af79c5e5e02-44
@message	{"originId":"MediaTailorDebug","awsAccountId":"820717217683","responseBody":"#EXTM3U\n#EXT-X-VERSION:3\n#EXT-X-INDEPENDENT-SEGMENTS\n#EXT-X-STREAM-INF:CODECS=\"avc1.640028,mp4a.40.2\",AVERAGE-BANDWIDTH=5711200,RESOLUTION
@timestamp	1596728312001
awsAccountId	820717217683
customerId	bb5755666fd4f9f014f0687a72e9a3726a78e563
eventTimestamp	2020-08-06T15:38:32.001Z
eventType	GENERATED_MANIFEST
mediaTailorPath	/v1/master/bb5755666fd4f9f014f0687a72e9a3726a78e563/MediaTailorDebug/index.m3u8?aws.logMode=DEBUG
originId	MediaTailorDebug
originRequestUrl	https://547f72e6652371c3.mediapackage.us-east-1.amazonaws.com/out/v1/eda3a5219a9e4d4ca750527b2b1835c8/index.m3u8
requestId	0fe480f0-8d38-42e7-bad5-32401b7b8dea
responseBody	#EXTM3U #EXT-X-VERSION:3 #EXT-X-INDEPENDENT-SEGMENTS #EXT-X-STREAM-INF:CODECS="avc1.640028,mp4a.40.2",AVERAGE-BANDWIDTH=5711200,RESOLUTION=1920x1080,FRAME-RATE=30.0,BANDWIDTH=8968960 ../../../manifest/bb5755666fd4f9f014f0687a72e9a3726a78e563/MediaTailorDebug/6768013a-63a2-4afd-a708-c425bdd1e9b2/0.m3u8 #EXT-X-STREAM-INF:CODECS="avc1.64001F,mp4a.40.2",AVERAGE-BANDWIDTH=3511200,RESOLUTION=1280x720,FRAME-RATE=30.0,BANDWIDTH=5448960 ../../../manifest/bb5755666fd4f9f014f0687a72e9a3726a78e563/MediaTailorDebug/6768013a-63a2-4afd-a708-c425bdd1e9b2/1.m3u8 #EXT-X-STREAM-INF:CODECS="avc1.4D401E,mp4a.40.2",AVERAGE-BANDWIDTH=1790760,RESOLUTION=640x480,FRAME-RATE=30.0,BANDWIDTH=2752604 ../../../manifest/bb5755666fd4f9f014f0687a72e9a3726a78e563/MediaTailorDebug/6768013a-63a2-4afd-a708-c425bdd1e9b2/2.m3u8 #EXT-X-STREAM-INF:CODECS="avc1.4D400D,mp4a.40.2",AVERAGE-BANDWIDTH=965742,RESOLUTION=320x240,FRAME-RATE=30.0,BANDWIDTH=1432591 ../../../manifest/bb5755666fd4f9f014f0687a72e9a3726a78e563/MediaTailorDebug/6768013a-63a2-4afd-a708-c425bdd1e9b2/3.m3u8
sessionId	6768013a-63a2-4afd-a708-c425bdd1e9b2
sessionType	HLS

Note that multiple log groups may be combined:



```
fields @timestamp, @message
| sort @timestamp desc
| limit 200 | filter sessionId = "ff4b8c13-dbca-4d14-9f2a-0dc57ad0282d"
```



#	@timestamp	@message
▶ 11	2020-08-06T12:00:24.…	{"originId":"MediaTailorDebug","awsAccountId":"820717217683","responseBody":"#EXTM3U\n#EXT-X-VERSION:3\n#EXT-X-TARGETDURATION:6\n#EXT-X
▶ 12	2020-08-06T12:00:18.…	{"originId":"MediaTailorDebug","awsAccountId":"820717217683","responseBody":"#EXTM3U\n#EXT-X-VERSION:3\n#EXT-X-TARGETDURATION:6\n#EXT-X
▶ 13	2020-08-06T12:00:06.…	{"originId":"MediaTailorDebug","awsAccountId":"820717217683","responseBody":"#EXTM3U\n#EXT-X-VERSION:6\n#EXT-X-TARGETDURATION:6\n#EXT-X
▶ 14	2020-08-06T12:00:00.…	{"originId":"MediaTailorDebug","awsAccountId":"820717217683","responseBody":"#EXTM3U\n#EXT-X-VERSION:6\n#EXT-X-TARGETDURATION:6\n#EXT-X
▶ 15	2020-08-06T12:00:00.…	{"originId":"MediaTailorDebug","awsAccountId":"820717217683","responseBody":"#EXTM3U\n#EXT-X-VERSION:3\n#EXT-X-TARGETDURATION:6\n#EXT-X
▶ 16	2020-08-06T11:59:54.…	{"originId":"MediaTailorDebug","awsAccountId":"820717217683","responseBody":"#EXTM3U\n#EXT-X-VERSION:6\n#EXT-X-TARGETDURATION:6\n#EXT-X
▶ 17	2020-08-06T11:59:48.…	{"originId":"MediaTailorDebug","awsAccountId":"820717217683","responseBody":"#EXTM3U\n#EXT-X-VERSION:3\n#EXT-X-TARGETDURATION:6\n#EXT-X
▶ 18	2020-08-06T11:59:47.…	{"originId":"MediaTailorDebug","awsAccountId":"820717217683","responseBody":"#EXTM3U\n#EXT-X-VERSION:6\n#EXT-X-TARGETDURATION:6\n#EXT-X
▶ 19	2020-08-06T11:59:47.…	{"originId":"MediaTailorDebug","awsAccountId":"820717217683","responseBody":"#EXTM3U\n#EXT-X-VERSION:3\n#EXT-X-TARGETDURATION:6\n#EXT-X
▶ 20	2020-08-06T11:59:46.…	{"originId":"MediaTailorDebug","awsAccountId":"820717217683","responseBody":"#EXTM3U\n#EXT-X-VERSION:6\n#EXT-X-TARGETDURATION:6\n#EXT-X
▼ 21	2020-08-06T11:59:46.…	{"eventTimestamp":"2020-08-06T15:59:46.116Z","requestId":"e6995a22-1a4a-41c5-a3de-6fe9a7858c7c","sessionType":"HLS","eventType":"MAKING

Field	Value
@ingestionTime	1596729597117
@log	820717217683:MediaTailor/AdDecisionServerInteractions
@logStream	MediaTailorDebug-0532a83707f96e185-9
@message	{"eventTimestamp":"2020-08-06T15:59:46.116Z","requestId":"e6995a22-1a4a-41c5-a3de-6fe9a7858c7c","sessionType":"HLS","eventType":"MAKING_ADS_REQ
@timestamp	1596729586116
adsRequestUrl	https://l3lk9gfqg9.execute-api.us-east-1.amazonaws.com/default/VASTEndpoint?client=
awsAccountId	820717217683
customerId	bb5755666fd4f9f014f0687a72e9a3726a78e563
eventDescription	Requesting advertisements from Ad Decision Server
eventTimestamp	2020-08-06T15:59:46.116Z
eventType	MAKING_ADS_REQUEST
originId	MediaTailorDebug
requestHeaders.0.name	User-Agent
requestHeaders.0.value	AppleCoreMedia/1.0.0.18G3020 (Macintosh; U; Intel Mac OS X 10_14_6; en_us)
requestHeaders.1.name	X-Device-User-Agent
requestHeaders.1.value	AppleCoreMedia/1.0.0.18G3020 (Macintosh; U; Intel Mac OS X 10_14_6; en_us)
requestHeaders.2.name	X-Forwarded-For
requestHeaders.2.value	72.21.198.67
requestId	e6995a22-1a4a-41c5-a3de-6fe9a7858c7c
sessionId	ff4b8c13-dbca-4d14-9f2a-0dc57ad0282d
sessionType	HLS

▶ 22	2020-08-06T11:59:45.…	{"originId":"MediaTailorDebug","awsAccountId":"820717217683","responseBody":"#EXTM3U\n#EXT-X-VERSION:3\n#EXT-X-INDEPENDENT-SEGMENTS\n#E
▶ 23	2020-08-06T11:59:45.…	{"originId":"MediaTailorDebug","awsAccountId":"820717217683","responseBody":"#EXTM3U\n#EXT-X-VERSION:3\n#EXT-X-INDEPENDENT-SEGMENTS\n#E

Tying it together

That was a lot of information. What sorts of things can be done with it?

Debugging ADS server parameters

During onboarding, an upstream ADS server might not be returning ads as expected. Using debugging, we can find exactly what is going on from client to server to pinpoint what might be happening. ADS servers rely on query strings and session information to determine the proper set of ads to return. Ensuring that all information arrives to the ADS server is critical.

To determine whether query strings are delivered end to end, we could use logs in the following way:

For this session:
https://2b5f4f55c18b4597b89527d420731043.mediatailor.us-east-1.amazonaws.com/v1/master/bb5755666fd4f9f014f0687a72e9a3726a78e563/MediaTailorDebug/index.m3u8?aws.logMode=DEBUG&ads.device=2

Note that the device query string is being added. Suppose that would like to see that this is passed all the way through to the ADS server.

First, ensure that MediaTailor receives the query string using the debug logs in the Manifest Service.

On the request for the master manifest, we see:

eventType	ORIGIN_MANIFEST
mediaTailorPath	/v1/master/bb5755666fd4f9f014f0687a72e9a3726a78e563/MediaTailorDebug/index.m3u8?aws.logMode=DEBUG&ads.device=2

eventType	GENERATED_MANIFEST
mediaTailorPath	/v1/master/bb5755666fd4f9f014f0687a72e9a3726a78e563/MediaTailorDebug/index.m3u8?aws.logMode=DEBUG&ads.device=2

Note that query strings are present on the request.

We then need to determine what is sent to the ADS server by looking into the AdDecisionServerInteractions log stream.

Search for MAKING_ADS_REQUEST.

adsRequestUrl	https://l3lk9gfqg9.execute-api.us-east-1.amazonaws.com/default/VASTEndpoint?client=2
eventDescription	Requesting advertisements from Ad Decision Server
eventTimestamp	2020-08-10T14:31:22.862Z
eventType	MAKING_ADS_REQUEST

Note that the proper query string is being sent to the ADS server. If ads are not displaying properly, but query strings are delivered to the ADS server, we now know that we should look into logs on the ADS server as MediaTailor is passing them correctly.

Debugging beacons

Now, let's assume that ads are being properly placed, but the ADS server is not receiving beacons. How would we figure out what the cause is? Note that beacons are fired to the same endpoint that the ads are returned from, except they use a POST method. The previous example used a custom ADS server that currently does not support beacons, so we can see beaconing errors easily by looking into logs. Inside the AdDecisionServerInteractions log group, not only are the VAST request/responses stored, but also the beaconing request/responses.

In the case that beacons cannot be delivered, a log similar to the following would appear:

Field	Value
@ingestionTime	1597069894523
@log	820717217683:MediaTailor/AdDecisionServerInteractions
@logStream	MediaTailorDebug-0f8ad0ce53fec324d-76
@message	{"eventTimestamp":"2020-08-10T14:31:23.418Z","requestId":"8da7aecf-350e-4cde-be5b-67d66d461478","sessionType":"HLS","eventType":"ERROR_FIRING_BEACON_FAILED","eventDescription":"Firing tracking beacon failed","awsAccountId":"820717217683","customerId":"bb5755666fd4f9f014f0687a72e9a3726a78e563","originId":"MediaTailorDebug","sessionId":"522d43ff-2f22-4bce-8e0d-1f03bd431f5a","error":"com.amazon.elemental.midas.common.io.Response.HttpClientException: Apache client had an unexpected error.","beaconInfo":{"beaconUri":"","headers":[{"name":"User-Agent","value":"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/13.1.2 Safari/605.1.15"},{"name":"X-Forwarded-For","value":"72.21.196.64"}],"trackingEvent":"impression"}}
@timestamp	1597069883419
beaconInfo.headers.0.name	User-Agent
beaconInfo.headers.0.value	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/13.1.2 Safari/605.1.15
beaconInfo.headers.1.name	X-Forwarded-For
beaconInfo.headers.1.value	72.21.196.64
beaconInfo.trackingEvent	impression
error	com.amazon.elemental.midas.common.io.Response.HttpClientException: Apache client had an unexpected error.

Field	Value
eventDescription	Firing tracking beacon failed
eventTimestamp	2020-08-10T14:31:23.418Z
eventType	ERROR_FIRING_BEACON_FAILED
originId	MediaTailorDebug
requestId	8da7aecf-350e-4cde-be5b-67d66d461478
sessionId	522d43ff-2f22-4bce-8e0d-1f03bd431f5a
sessionType	HLS

Notice that the endpoint had an “unexpected error”. This likely means that the response of the beacon did not conform to specs. If a timeout occurred or another error, that would be returned in the logs as well.

Again, if an error is returned from the ADS server in the MediaTailor logs, it is best to next investigate the issue on the ADS server’s logs to see what might be the issue.

Ensuring correct ads are returned

Assuming that MediaTailor and the upstream ADS server are communicating properly, but incorrect ads are returned, we can dive into the cause again with the available logs.

Upon parsing the VAST response, MediaTailor logs present how it will use the response to fill avails.

The following is an example:

Field	Value
@ingestionTime	1597069897892
@log	820717217683:MediaTailor/AdDecisionServerInteractions
@logStream	MediaTailorDebug-0a9a96180a9295bcb-26
@timestamp	1597069883228
avail.availId	57153
avail.creativeAds.0.adContent.adPlaylistUris.https://547f72e6652371c3.mediapackage.us-east-1.amazonaws.com/out/v1/eda3a5219a9e4d4ca750527b2b1835c8/index_1.m3u8	http://transcode-ad-manifests.us-east-1.566271146650.servicelookup.internal/tm/bb5755666fd4f9f014f0687a72e9a3726a78e563/jj6fqbl2ubahvk4prwllw3h3owrwtpns/asset_1080_8_3.m3u8
avail.creativeAds.0.adContent.adPlaylistUris.https://547f72e6652371c3.mediapackage.us-east-1.amazonaws.com/out/v1/eda3a5219a9e4d4ca750527b2b1835c8/index_2.m3u8	http://transcode-ad-manifests.us-east-1.566271146650.servicelookup.internal/tm/bb5755666fd4f9f014f0687a72e9a3726a78e563/jj6fqbl2ubahvk4prwllw3h3owrwtpns/asset_720_5_2.m3u8
avail.creativeAds.0.adContent.adPlaylistUris.https://547f72e6652371c3.mediapackage.us-east-1.amazonaws.com/out/v1/eda3a5219a9e4d4ca750527b2b1835c8/index_3.m3u8	http://transcode-ad-manifests.us-east-1.566271146650.servicelookup.internal/tm/bb5755666fd4f9f014f0687a72e9a3726a78e563/jj6fqbl2ubahvk4prwllw3h3owrwtpns/asset_480_2_0.m3u8
avail.creativeAds.0.adContent.adPlaylistUris.https://547f72e6652371c3.mediapackage.us-east-1.amazonaws.com/out/v1/eda3a5219a9e4d4ca750527b2b1835c8/index_4.m3u8	http://transcode-ad-manifests.us-east-1.566271146650.servicelookup.internal/tm/bb5755666fd4f9f014f0687a72e9a3726a78e563/jj6fqbl2ubahvk4prwllw3h3owrwtpns/asset_240_1_1.m3u8
avail.creativeAds.0.transcodedAdDuration	15.067
avail.creativeAds.0.uri	https://mediatailorads.s3.amazonaws.com/2_15.mp4
avail.creativeAds.0.vastDuration	15.0
avail.creativeAds.1.adContent.adPlaylistUris.https://547f72e6652371c3.mediapackage.us-east-1.amazonaws.com/out/v1/eda3a5219a9e4d4ca750527b2b1835c8/index_1.m3u8	http://transcode-ad-manifests.us-east-1.566271146650.servicelookup.internal/tm/bb5755666fd4f9f014f0687a72e9a3726a78e563/45jnw2fxoraqzb2m4voesybcrhc2prm4/asset_1080_8_3.m3u8
avail.creativeAds.1.adContent.adPlaylistUris.https://547f72e6652371c3.mediapackage.us-east-1.amazonaws.com/out/v1/eda3a5219a9e4d4ca750527b2b1835c8/index_2.m3u8	http://transcode-ad-manifests.us-east-1.566271146650.servicelookup.internal/tm/bb5755666fd4f9f014f0687a72e9a3726a78e563/45jnw2fxoraqzb2m4voesybcrhc2prm4/asset_720_5_2.m3u8
avail.creativeAds.1.adContent.adPlaylistUris.https://547f72e6652371c3.mediapackage.us-east-1.amazonaws.com/out/v1/eda3a5219a9e4d4ca750527b2b1835c8/index_3.m3u8	http://transcode-ad-manifests.us-east-1.566271146650.servicelookup.internal/tm/bb5755666fd4f9f014f0687a72e9a3726a78e563/45jnw2fxoraqzb2m4voesybcrhc2prm4/asset_480_2_0.m3u8
avail.creativeAds.1.adContent.adPlaylistUris.https://547f72e6652371c3.mediapackage.us-east-1.amazonaws.com/out/v1/eda3a5219a9e4d4ca750527b2b1835c8/index_4.m3u8	http://transcode-ad-manifests.us-east-1.566271146650.servicelookup.internal/tm/bb5755666fd4f9f014f0687a72e9a3726a78e563/45jnw2fxoraqzb2m4voesybcrhc2prm4/asset_240_1_1.m3u8
avail.creativeAds.1.transcodedAdDuration	30.033

Field	Value
avail.creativeAds.1.uri	https://mediatailorads.s3.amazonaws.com/2_30.mp4
avail.creativeAds.1.vastDuration	30.0
avail.filledDuration	45.1
avail.fillRate	0.7516666666666667
avail.numAds	2
avail.originAvailDuration	60.0
avail.slateAd.adContent.adPlaylistUris.https://547f72e6652371c3.mediapackage.us-east-1.amazonaws.com/out/v1/eda3a5219a9e4d4ca750527b2b1835c8/index_1.m3u8	http://transcode-ad-manifests.us-east-1.566271146650.servicelookup.internal/tm/bb5755666fd4f9f014f0687a72e9a3726a78e563/h7qhj2cuqfcajlp46qroe6o42f67dbh2/asset_1080_8_3.m3u8
avail.slateAd.adContent.adPlaylistUris.https://547f72e6652371c3.mediapackage.us-east-1.amazonaws.com/out/v1/eda3a5219a9e4d4ca750527b2b1835c8/index_2.m3u8	http://transcode-ad-manifests.us-east-1.566271146650.servicelookup.internal/tm/bb5755666fd4f9f014f0687a72e9a3726a78e563/h7qhj2cuqfcajlp46qroe6o42f67dbh2/asset_720_5_2.m3u8
avail.slateAd.adContent.adPlaylistUris.https://547f72e6652371c3.mediapackage.us-east-1.amazonaws.com/out/v1/eda3a5219a9e4d4ca750527b2b1835c8/index_3.m3u8	http://transcode-ad-manifests.us-east-1.566271146650.servicelookup.internal/tm/bb5755666fd4f9f014f0687a72e9a3726a78e563/h7qhj2cuqfcajlp46qroe6o42f67dbh2/asset_480_2_0.m3u8
avail.slateAd.adContent.adPlaylistUris.https://547f72e6652371c3.mediapackage.us-east-1.amazonaws.com/out/v1/eda3a5219a9e4d4ca750527b2b1835c8/index_4.m3u8	http://transcode-ad-manifests.us-east-1.566271146650.servicelookup.internal/tm/bb5755666fd4f9f014f0687a72e9a3726a78e563/h7qhj2cuqfcajlp46qroe6o42f67dbh2/asset_240_1_1.m3u8
avail.slateAd.transcodedAdDuration	120.133
avail.slateAd.uri	https://testtestads.s3-sa-east-1.amazonaws.com/halloween.mp4
awsAccountId	820717217683
customerId	bb5755666fd4f9f014f0687a72e9a3726a78e563
eventDescription	Successfully filled avail
eventTimestamp	2020-08-10T14:31:23.228Z
eventType	FILLED_AVAIL
originId	MediaTailorDebug
requestId	3d49a6b6-149b-497e-b736-24d313be3127
sessionId	522d43ff-2f22-4bce-8e0d-1f03bd431f5a
sessionType	HLS

Note that we see the plan for filling the space. We had a 60 second break with 2 ads returned from the ADS server. It used those two ads:

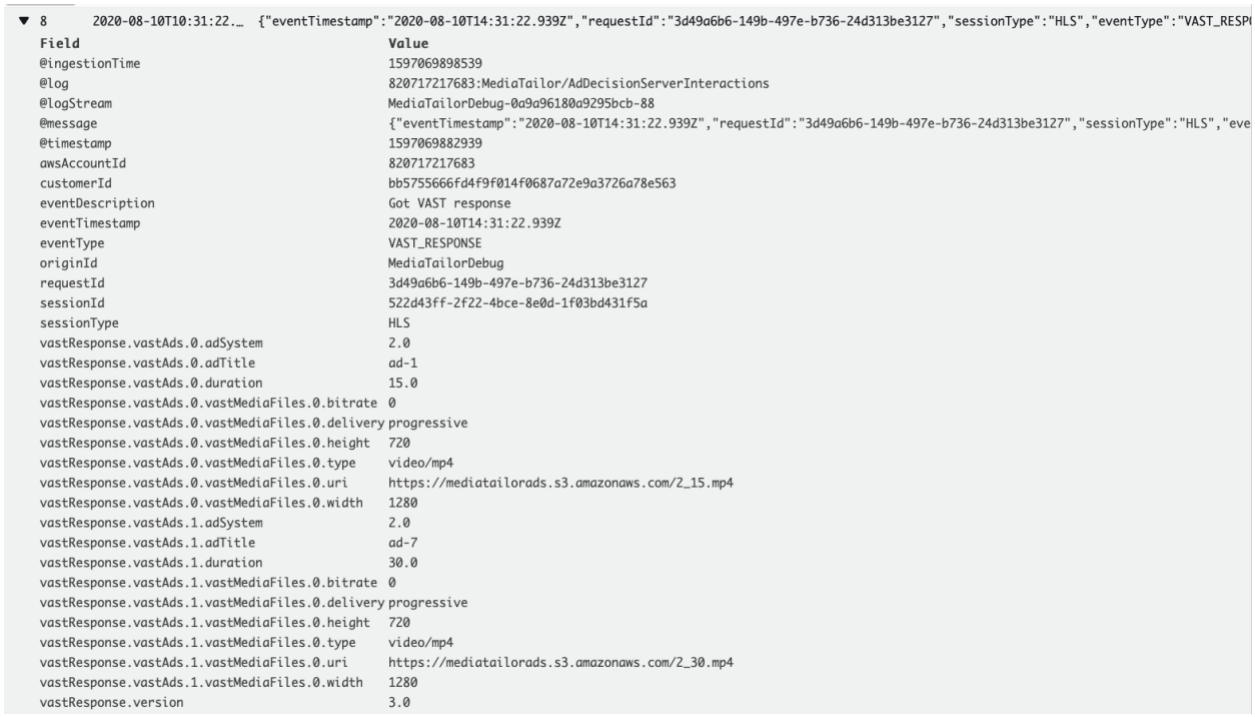
2_15.mp4

2_30.mp4

Since the break was 60 seconds, the remainder was filled by the configured slate ad.

halloween.mp4

The previous VAST response log line show what was parsed from the ADS server:

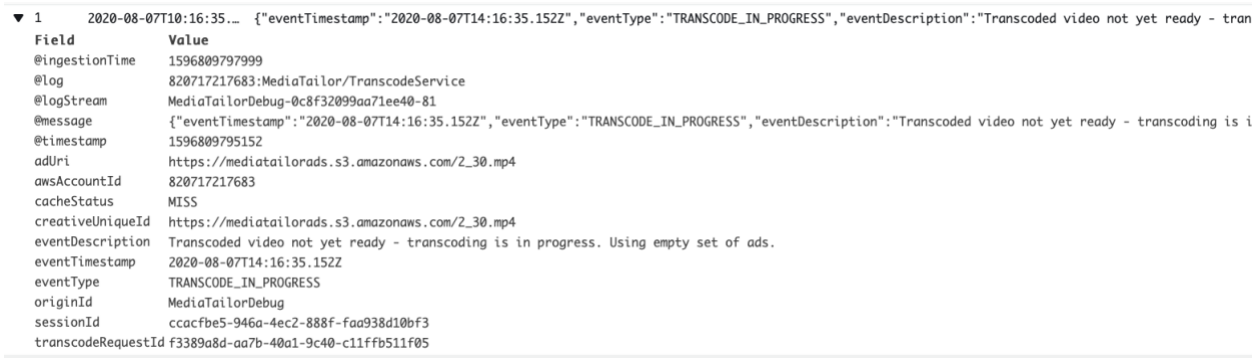


Field	Value
@ingestionTime	1597069898539
@log	820717217683:MediaTailor/AdDecisionServerInteractions
@logStream	MediaTailorDebug-0a9a96180a9295bcb-88
@message	{"eventTimestamp":"2020-08-10T14:31:22.939Z","requestId":"3d49a6b6-149b-497e-b736-24d313be3127","sessionType":"HLS","eve
@timestamp	1597069882939
awsAccountId	820717217683
customerId	bb5755666fd4f9f014f0687a72e9a3726a78e563
eventDescription	Got VAST response
eventTimestamp	2020-08-10T14:31:22.939Z
eventType	VAST_RESPONSE
originId	MediaTailorDebug
requestId	3d49a6b6-149b-497e-b736-24d313be3127
sessionId	522d43ff-2f22-4bce-8e0d-1f03bd431f5a
sessionType	HLS
vastResponse.vastAds.0.adSystem	2.0
vastResponse.vastAds.0.adTitle	ad-1
vastResponse.vastAds.0.duration	15.0
vastResponse.vastAds.0.vastMediaFiles.0.bitrate	0
vastResponse.vastAds.0.vastMediaFiles.0.delivery	progressive
vastResponse.vastAds.0.vastMediaFiles.0.height	720
vastResponse.vastAds.0.vastMediaFiles.0.type	video/mp4
vastResponse.vastAds.0.vastMediaFiles.0.uri	https://mediatailorads.s3.amazonaws.com/2_15.mp4
vastResponse.vastAds.0.vastMediaFiles.0.width	1280
vastResponse.vastAds.1.adSystem	2.0
vastResponse.vastAds.1.adTitle	ad-7
vastResponse.vastAds.1.duration	30.0
vastResponse.vastAds.1.vastMediaFiles.0.bitrate	0
vastResponse.vastAds.1.vastMediaFiles.0.delivery	progressive
vastResponse.vastAds.1.vastMediaFiles.0.height	720
vastResponse.vastAds.1.vastMediaFiles.0.type	video/mp4
vastResponse.vastAds.1.vastMediaFiles.0.uri	https://mediatailorads.s3.amazonaws.com/2_30.mp4
vastResponse.vastAds.1.vastMediaFiles.0.width	1280
vastResponse.version	3.0

In the case that an ad is returned, but the slate is displaying instead, it is possible that the returned ad as not been transcoded yet.

MediaTailor must transcode the ad returned from the ADS server to match the bitrates delivered to the client. This happens on the first pull of the asset from the ADS server. Subsequent views will not require a transcode.

In the following, the transcode job is running, so slate ads are displayed instead:



Field	Value
@ingestionTime	1596809797999
@log	820717217683:MediaTailor/TranscodeService
@logStream	MediaTailorDebug-0c8f32099aa71ee40-81
@message	{"eventTimestamp":"2020-08-07T14:16:35.152Z","eventType":"TRANSCODE_IN_PROGRESS","eventDescription":"Transcoded video not yet ready - transcoding is i
@timestamp	1596809795152
adUri	https://mediatailorads.s3.amazonaws.com/2_30.mp4
awsAccountId	820717217683
cacheStatus	MISS
creativeUniqueId	https://mediatailorads.s3.amazonaws.com/2_30.mp4
eventDescription	Transcoded video not yet ready - transcoding is in progress. Using empty set of ads.
eventTimestamp	2020-08-07T14:16:35.152Z
eventType	TRANSCODE_IN_PROGRESS
originId	MediaTailorDebug
sessionId	ccacfbe5-946a-4ec2-888f-faa938d10bf3
transcodeRequestId	f3389a8d-aa7b-40a1-9c40-c11ffb511f05

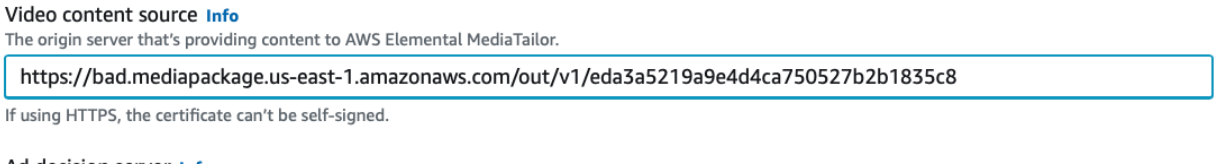
Diagnosing Origin Issues

As MediaTailor must connect to a downstream origin to retrieve the template playlist, we'd like to be able to diagnose any issues with that flow.

Normally (If debug is enabled), you would see MediaTailor communicating with the origin server normally, printing the content from the returned manifest that it will use to transform if needed.

Field	Value
@ingestionTime	1597071927427
@log	820717217683:MediaTailor/ManifestService
@logStream	MediaTailorDebug-084a03f902d2b12da-56
@message	{"originId":"MediaTailorDebug","awsAccountId":"820717217683","responseBody":"#EXTM3U\n#EXT-X-VERSION:3\n#EXT-X-TARGETDURATION:6\n#EXT-X-MEDIA-SEQUENCE
@timestamp	1597071922413
awsAccountId	820717217683
customerId	bb5755666fd4f9f014f0687a72e9a3726a78e563
eventTimestamp	2020-08-10T15:05:22.413Z
eventType	ORIGIN_MANIFEST
mediaTailorPath	/v1/manifest/bb5755666fd4f9f014f0687a72e9a3726a78e563/MediaTailorDebug/522d43ff-2f22-4bce-8e0d-1f03bd431f5a/2.m3u8
originId	MediaTailorDebug
originRequestUrl	https://547f72e6652371c3.mediapackage.us-east-1.amazonaws.com/out/v1/eda3a5219a9e4d4ca750527b2b1835c8/index_3.m3u8
requestId	ff1f697c-1601-40fd-a706-80db121f740c
responseBody	#EXTM3U #EXT-X-VERSION:3 #EXT-X-TARGETDURATION:6 #EXT-X-MEDIA-SEQUENCE:57486 #EXTINF:6.000, index_3_57486.ts?m=1596809448 #EXTINF:6.000, index_3_57487.ts?m=1596809448 #EXTINF:6.000, index_3_57488.ts?m=1596809448 #EXTINF:6.000, index_3_57489.ts?m=1596809448 #EXTINF:6.000, index_3_57490.ts?m=1596809448 #EXTINF:6.000, index_3_57491.ts?m=1596809448 #EXTINF:6.000, index_3_57492.ts?m=1596809448 #EXTINF:6.000, index_3_57493.ts?m=1596809448 #EXTINF:6.000, index_3_57494.ts?m=1596809448 #EXTINF:6.000, index_3_57495.ts?m=1596809448
sessionId	522d43ff-2f22-4bce-8e0d-1f03bd431f5a
sessionType	HLS

To simulate an origin failure, we can change the origin domain name in the configuration and look at the logs.



First off, when visiting the stream from a player, the message will be displayed directly to the browser:

In the logs, a corresponding message will appear:

Field	Value
@ingestionTime	1597072391053
@log	820717217683:MediaTailor/ManifestService
@logStream	MediaTailorDebug-02cffcd83df38020c-40
@message	{"eventTimestamp":"2020-08-10T15:13:08.204Z","eventType":"UNKNOWN_HOST","eventDescription":"Host is not known","awsAccountId":"820717217683","originId":"MediaTailorDebug","sessionId":"5e6a45a1-9239-48c6-a31c-544b86425d01","assetPath":"index.m3u8","originFullUrl":"https://bad.mediapackage.us-east-1.amazonaws.com/out/v1/eda3a5219a9e4d4ca750527b2b1835c8/index.m3u8"}
@timestamp	1597072388204
assetPath	index.m3u8
eventDescription	Host is not known
eventTimestamp	2020-08-10T15:13:08.204Z
eventType	UNKNOWN_HOST

Field	Value
originFullUrl	https://bad.mediapackage.us-east-1.amazonaws.com/out/v1/eda3a5219a9e4d4ca750527b2b1835c8/index.m3u8
originId	MediaTailorDebug
sessionId	5e6a45a1-9239-48c6-a31c-544b86425d01

Possible reasons for being unable to communicate to an origin are:

- IP Access Control List on the Origin
- Tokenization enabled on the origin
- Incorrect hostname configured

Conclusion

Ad Insertion workflows can become complicated very quickly, so this document provides deeper insights into the backend workflow. Combining standard logging with use of debug logging can provide the detailed information needed to ensure that all pieces of a workflow are working as designed. Many customers are utilizing this new functionality today to gather insights into multi-origin DAI workflows, as well as new customers just jumping into Ad Insertion wishing for more insight during the development process.