# Table of Contents

What is AWS Elemental MediaPackage? ................................................................. 1  
Are you a first-time user of MediaPackage? .......................................................... 1  
Concepts and terminology .................................................................................... 1  
  Live components ................................................................................................. 2  
  VOD Components ............................................................................................... 3  
Supported inputs and outputs ................................................................................ 3  
  Live supported codecs and input types ............................................................... 3  
  Live-to-VOD supported codecs and input types .................................................. 5  
  VOD supported codecs and input types .............................................................. 6  
How AWS Elemental MediaPackage works ........................................................ 8  
  Live content processing ...................................................................................... 8  
  VOD Content Processing .................................................................................... 11  
  Live and VOD manifest reference ..................................................................... 12  
Features of AWS Elemental MediaPackage ......................................................... 13  
Related services .................................................................................................. 14  
Accessing MediaPackage .................................................................................... 15  
Pricing for MediaPackage ..................................................................................... 15  
Regions for MediaPackage .................................................................................. 15  
Setting up ............................................................................................................ 16  
  Signing up for AWS ............................................................................................ 16  
  Creating an admin IAM user .............................................................................. 16  
  Creating a non-admin IAM user ...................................................................... 17  
    Step 1: (Optional) create a policy for Amazon CloudFront ............................. 18  
    Step 2: (Optional) create a policy for AWS Elemental MediaPackage VOD ... 18  
    Step 3: Create users ....................................................................................... 20  
  Allowing AWS Elemental MediaPackage to access other AWS services ............ 21  
    Step 1: Create a policy .................................................................................. 21  
    Step 2: Create a role ..................................................................................... 25  
    Step 3: Modify the trust relationship .............................................................. 25  
  (Optional) setting up encryption ...................................................................... 26  
Getting started ..................................................................................................... 27  
  Live content delivery ....................................................................................... 27  
    Prerequisites ................................................................................................. 27  
    Step 1: Access AWS Elemental MediaPackage .............................................. 27  
    Step 2: Create a channel .............................................................................. 27  
    Step 3: Create endpoints .............................................................................. 28  
    (Optional) step 4: Monitor AWS Elemental MediaPackage activity .............. 29  
    Step 5: Clean up ............................................................................................ 29  
  Live-to-VOD content delivery ........................................................................... 29  
    Prerequisites ................................................................................................. 30  
    Step 1: Access AWS Elemental MediaPackage .............................................. 30  
    Step 2: Ingest live content ............................................................................ 30  
    Step 3: Extract a VOD asset ........................................................................... 32  
    Step 4: (Optional) output VOD content ........................................................ 32  
    (Optional) step 5: Monitor AWS Elemental MediaPackage activity .............. 34  
    Step 6: Clean up ............................................................................................ 35  
  VOD content delivery ....................................................................................... 36  
    Prerequisites ................................................................................................. 36  
    Step 1: Access AWS Elemental MediaPackage .............................................. 36  
    Step 2: Create a packaging group ................................................................. 36  
    Step 3: Create a packaging configuration .................................................... 37  
    Step 4: Create an asset .................................................................................. 37  
    Step 5: Provide playback URLs ...................................................................... 38  
    (Optional) step 6: Monitor AWS Elemental MediaPackage activity .............. 38
## Tagging resources

Working with manifest filters .............................................................. 106
Manifest filter query parameters .......................................................... 107
Manifest filtering examples ................................................................. 110
Special conditions for HLS and CMAF manifests ................................. 110
Error conditions ............................................................................... 110

## Monitoring with CloudWatch

Metadata passthrough ........................................................................ 111
ID3 metadata considerations ............................................................... 111
KLV metadata considerations .............................................................. 112

## Rendition groups

When to use rendition groups ............................................................. 113
When not to use rendition groups ........................................................ 113

## SCTE-35 messages

SCTE-35 settings in MediaPackage ....................................................... 114
How it works ................................................................................... 115
EXT-x-DATERANGE ad markers ......................................................... 116

## Time-shifted viewing

Rules for start and end parameters ..................................................... 119

## Security

Data protection .................................................................................. 121
Implementing DRM .......................................................................... 122
Implementing CDN authorization ....................................................... 122
Identity and Access Management ....................................................... 122
Audience ....................................................................................... 123
Authenticating with identities ............................................................ 123
Managing access using policies ........................................................ 125
How AWS Elemental MediaPackage Works with IAM ....................... 127
Identity-Based Policy Examples ........................................................ 129
Policy examples for secrets in AWS Secrets Manager ....................... 132
Troubleshooting ............................................................................ 133
Learn More .................................................................................... 135
Using Service-Linked Roles ............................................................... 135

## Logging and monitoring

Amazon CloudWatch alarms .............................................................. 137
AWS CloudTrail logs ....................................................................... 137
AWS Elemental MediaPackage access logs ....................................... 138
AWS Trusted Advisor ....................................................................... 138
Compliance validation ..................................................................... 138
Resilience ....................................................................................... 138
Infrastructure security ..................................................................... 139

## Monitoring with CloudWatch

Monitoring with CloudWatch metrics ............................................... 140
Live content metrics ....................................................................... 141
VOD content metrics ....................................................................... 146

## Access logging

Permissions to publish access logs to CloudWatch ............................... 158
Enable access logging ..................................................................... 158
Disable access logging .................................................................... 159
Access log format ........................................................................... 159
Read the access logs ....................................................................... 161

## Tagging resources

Tag restrictions ................................................................................ 163
What is AWS Elemental MediaPackage?

MediaPackage is a just-in-time video packaging and origination service that runs in the AWS Cloud. With MediaPackage, you can deliver highly secure, scalable, and reliable video streams to a wide variety of playback devices and content delivery networks (CDNs).

MediaPackage offers a broadcast-grade viewing experience for viewers, while allowing you the flexibility to control and protect your content. Additionally, the built-in resiliency and scalability of MediaPackage means that you have the right amount of resources at the right time, with no manual intervention required.

Topics
- Are you a first-time user of MediaPackage? (p. 1)
- Concepts and terminology (p. 1)
- Supported inputs and outputs (p. 3)
- How AWS Elemental MediaPackage works (p. 8)
- Features of AWS Elemental MediaPackage (p. 13)
- Related services (p. 14)
- Accessing MediaPackage (p. 15)
- Pricing for MediaPackage (p. 15)
- Regions for MediaPackage (p. 15)

Are you a first-time user of MediaPackage?

If you are a first-time user of MediaPackage, we recommend that you begin by reading the following sections:

- How AWS Elemental MediaPackage works (p. 8)
- Concepts and terminology (p. 1)
- Features of AWS Elemental MediaPackage (p. 13)
- Getting started with AWS Elemental MediaPackage (p. 27)

Concepts and terminology

AWS Elemental MediaPackage includes the following components:

Just-in-time packaging

MediaPackage performs just-in-time packaging (JITP). When a playback device requests content, MediaPackage dynamically customizes the live video streams and creates a manifest in a format that is compatible with the requesting device.
Origination service

MediaPackage is considered an origination service because it is the point of distribution for media content delivery.

Packager

A packager prepares output streams for access by different types of players. The packager type specifies the streaming format that MediaPackage delivers from the endpoint (either DASH-ISO, Microsoft Smooth Streaming, CMAF, or Apple HLS). Additional packager settings include buffer and update durations and manifest tag handling instructions.

A packager is a part of an endpoint. Each endpoint must have one, and only one, packager. To use different packager types for the same content, create multiple endpoints on the channel.

Source Content

Source contents are live streams and video files that MediaPackage ingests.

- For live video, source content comes from an upstream encoder, such as AWS Elemental MediaLive. MediaPackage supports HLS source content.
- For video on demand (VOD), source content resides in an Amazon S3 bucket in your AWS account. MediaPackage supports HLS and MP4 (.smil manifest format) content.

Stream

A stream refers to the content input and output of AWS Elemental MediaPackage.

For live workflows, an upstream encoder sends a live stream as an input to MediaPackage to the channel. When a downstream device requests playback of the content, MediaPackage dynamically packages the stream (including specifying the packager type, adding encryption, and configuring track outputs) and delivers it to the requesting device as an output of the endpoint. An endpoint can produce multiple streams.

For VOD workflows, MediaPackage pulls file-based content from Amazon S3. As with live workflows, when a downstream device requests playback of the content, MediaPackage dynamically packages the stream and delivers it to the requesting device as an output of the asset resource.

Track

Tracks make up the output content stream. MediaPackage includes selected video, audio, and subtitles or captions tracks in the output stream. The stream delivers the tracks to the player (either directly or through a CDN), and the player plays back the tracks based on player logic or network conditions (such as available bandwidth).

Live components

The following components apply to live workflows in MediaPackage:

Channel

A channel represents the entry point for a content stream into MediaPackage. Upstream encoders such as AWS Elemental MediaLive send content to the channel. When MediaPackage receives a content stream, it packages the content and outputs the stream from an endpoint that you create on the channel. There is one channel for each incoming set of ABR streams.

Endpoint

An endpoint is part of a channel and represents the packaging aspect of MediaPackage. When you create an endpoint on a channel, you indicate what streaming format, packaging parameters, and
features the output stream will use. Downstream devices request content from the endpoint. A channel can have multiple endpoints.

Harvest Job

A harvest job is a task that you create to extract a video on demand (VOD) asset from a live content stream. A harvest job defines the start and end times of the VOD asset, and where MediaPackage should store the asset. When the job runs, MediaPackage creates an HLS clip for the times that you indicated. This clip is stored as a VOD asset in an Amazon S3 bucket of your choosing. You can use the VOD functionality in MediaPackage to serve the asset to end users.

VOD Components

The following components apply to video on demand (VOD) workflows in MediaPackage:

Asset

An asset represents the entry point for file-based content into MediaPackage. MediaPackage uses the information in the asset to locate and ingest your source content from Amazon S3. When you create an asset in MediaPackage, you associate it with a packaging group, which holds one or more packaging configurations. Each asset and packaging configuration combination provides a URL for playback of repackaged content. Each asset is associated with all the packaging configurations within one packaging group.

Packaging Configuration

A packaging configuration defines how MediaPackage formats, encrypts, and delivers source content to viewers. The packaging configuration includes settings such as stream selection, encryption, segment duration and combining, as well as one or more HLS, DASH, CMAF, or MSS manifest definitions.

Packaging Group

A packaging group is a set of one or more packaging configurations. Because you can associate the group to more than one asset, the group provides an efficient way to associate multiple packaging configurations with multiple assets.

Supported inputs and outputs

This section describes the input types, input codecs, and output codecs that AWS Elemental MediaPackage supports for live and video on demand (VOD) content.

Topics

• Live supported codecs and input types (p. 3)
• Live-to-VOD supported codecs and input types (p. 5)
• VOD supported codecs and input types (p. 6)

Live supported codecs and input types

The following sections describe supported input types and codecs for live streaming content.

Supported input types

These are the input types that AWS Elemental MediaPackage supports for live content.
## Live supported codecs and input types

<table>
<thead>
<tr>
<th><strong>MediaPackage input type</strong></th>
<th><strong>Use case</strong></th>
</tr>
</thead>
</table>
| HLS                        | Push an HLS stream from an external source or encoder (such as AWS Elemental MediaLive) using the HTTPS protocol. Additional requirements:  
  • Inputs must be over WebDAV and with digest authentication.  
  • Media segments must not be encrypted.  
  • Streams can contain either muxed video and audio tracks, or unmuxed tracks.  
  • The input must contain at least one video track. MediaPackage does not support inputs that contain no video track. |

### Supported input codecs

These are the video, audio, and subtitles codecs that MediaPackage supports for source content streams.

<table>
<thead>
<tr>
<th><strong>Media container</strong></th>
<th><strong>Video codecs</strong></th>
<th><strong>Audio codecs</strong></th>
<th><strong>Subtitles format</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Video: TS</td>
<td>• H.264 (AVC)</td>
<td>• AAC</td>
<td>WebVTT</td>
</tr>
<tr>
<td>• Audio: TS, AAC, AC3, or EC3</td>
<td>• H.265 (HEVC) with HDR-10 support</td>
<td>• Dolby Digital</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Dolby Digital Plus</td>
<td></td>
</tr>
</tbody>
</table>

### Supported output codecs

These are the video, audio, and subtitles codecs that MediaPackage supports when delivering live content.

<table>
<thead>
<tr>
<th><strong>Endpoint type</strong></th>
<th><strong>Manifest format</strong></th>
<th><strong>Media container</strong></th>
<th><strong>Video codecs</strong></th>
<th><strong>Audio codecs</strong></th>
<th><strong>Subtitles format</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple HLS</td>
<td>HLS</td>
<td>HLS</td>
<td>• Video: TS</td>
<td>• AAC</td>
<td>WebVTT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Audio: TS or AAC</td>
<td>• Dolby Digital</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Dolby Digital Plus</td>
<td></td>
</tr>
<tr>
<td>Microsoft Smooth</td>
<td>MSS</td>
<td>MP4</td>
<td>• H.264 (AVC)</td>
<td>• AAC</td>
<td>DFXP</td>
</tr>
<tr>
<td>Iso</td>
<td>MPEG-DASH</td>
<td>MP4</td>
<td>• H.265 (HEVC) with HDR-10 support</td>
<td>• Dolby Digital</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Dolby Digital Plus</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• H.264 (AVC)</td>
<td>• AAC</td>
<td>EBU-TT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• H.265 (HEVC) with HDR-10 support</td>
<td>• Dolby Digital</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Dolby Digital Plus</td>
<td></td>
</tr>
</tbody>
</table>
Live-to-VOD supported codecs and input types

The following sections describe supported input types and codecs for live-to-VOD assets that are harvested from streaming content in AWS Elemental MediaPackage.

Supported input types

These are the input types that AWS Elemental MediaPackage supports for live-to-VOD assets.

<table>
<thead>
<tr>
<th>MediaPackage input type</th>
<th>Use case</th>
</tr>
</thead>
</table>
| HLS                     | Extract a portion of a live HLS endpoint in MediaPackage and save it as a live-to-VOD asset. Additional requirements:  
  • The endpoint must have a startover window defined. This window is the maximum length of live-to-VOD asset that you can harvest from the endpoint.  
  • Media segments must not be encrypted.  
  • Streams can contain either muxed video and audio tracks, or unmuxed tracks.  
  • The input must contain at least one video track. MediaPackage does not support inputs that contain no video track. |

Supported input codecs

These are the video, audio, and subtitles codecs that MediaPackage supports for live-to-VOD assets.

<table>
<thead>
<tr>
<th>Media container</th>
<th>Video codecs</th>
<th>Audio codecs</th>
<th>Subtitles format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video: TS</td>
<td>H.264 (AVC)</td>
<td>AAC</td>
<td>WebVTT</td>
</tr>
<tr>
<td>Audio: TS, AAC, AC3, or EC3</td>
<td>H.265 (HEVC) with HDR-10 support</td>
<td>Dolby Digital, Dolby Digital Plus</td>
<td></td>
</tr>
</tbody>
</table>

Supported output codecs

These are the video, audio, and subtitles codecs that MediaPackage supports when saving a live-to-VOD asset to an Amazon S3 bucket.
VOD supported codecs and input types

The following sections describe supported input types and codecs for file-based video on demand (VOD) content.

Supported input types

These are the input types that AWS Elemental MediaPackage supports for VOD content.

<table>
<thead>
<tr>
<th>MediaPackage input type</th>
<th>Use case</th>
</tr>
</thead>
<tbody>
<tr>
<td>HLS</td>
<td>Pull an HLS stream set from an Amazon Simple Storage Service bucket, with or without a secure connection.</td>
</tr>
<tr>
<td></td>
<td>Additional requirements:</td>
</tr>
<tr>
<td></td>
<td>• Media segments must not be encrypted.</td>
</tr>
<tr>
<td></td>
<td>• Streams can contain either muxed video and audio tracks, or unmuxed tracks.</td>
</tr>
<tr>
<td></td>
<td>• The input must contain at least one video track. MediaPackage does not support inputs that contain no video track.</td>
</tr>
<tr>
<td>SMIL</td>
<td>Pull an MP4 stream set referenced by a .smil manifest from an Amazon Simple Storage Service bucket, with or without a secure connection. For information about the .smil manifest, see Creating a SMIL file (p. 7).</td>
</tr>
<tr>
<td></td>
<td>Additional requirements:</td>
</tr>
<tr>
<td></td>
<td>• MP4 container must not be fragmented.</td>
</tr>
<tr>
<td></td>
<td>• Media segments must not be encrypted.</td>
</tr>
<tr>
<td></td>
<td>• Streams can contain either muxed video and audio tracks, or only video tracks.</td>
</tr>
<tr>
<td></td>
<td>• Streams must have an equal time base.</td>
</tr>
</tbody>
</table>

Supported input codecs

These are the video, audio, and subtitles codecs that MediaPackage supports for file-based source content.
VOD supported codecs and input types

<table>
<thead>
<tr>
<th>Input type</th>
<th>Media container</th>
<th>Video codecs</th>
<th>Audio codecs</th>
<th>Subtitles format</th>
</tr>
</thead>
<tbody>
<tr>
<td>HLS</td>
<td>• Video: TS &lt;br&gt;• Audio: TS, AAC, AC3, or EC3</td>
<td>• H.264 (AVC) &lt;br&gt;• H.265 (HEVC) with HDR-10 support</td>
<td>• AAC &lt;br&gt;• Dolby Digital &lt;br&gt;• Dolby Digital Plus</td>
<td>WebVTT</td>
</tr>
<tr>
<td>SMIL</td>
<td>MP4 (non-fragmented)</td>
<td>• H.264 (AVC) &lt;br&gt;• H.265 (HEVC) with HDR-10 support</td>
<td>• AAC &lt;br&gt;• Dolby Digital &lt;br&gt;• Dolby Digital Plus</td>
<td>SRT</td>
</tr>
</tbody>
</table>

Supported output codecs

These are the video, audio, and subtitles codecs that MediaPackage supports for delivering VOD content.

<table>
<thead>
<tr>
<th>Endpoint type</th>
<th>Manifest format</th>
<th>Media container</th>
<th>Video codecs</th>
<th>Audio codecs</th>
<th>Subtitles format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple HLS</td>
<td>HLS</td>
<td>• Video: TS &lt;br&gt;• Audio: TS, AAC, AC3, or EC3</td>
<td>• H.264 (AVC) &lt;br&gt;• H.265 (HEVC) with HDR-10 support</td>
<td>• AAC &lt;br&gt;• Dolby Digital &lt;br&gt;• Dolby Digital Plus</td>
<td>WebVTT</td>
</tr>
<tr>
<td>Microsoft Smooth MSS</td>
<td>MP4</td>
<td>• H.264 (AVC) &lt;br&gt;• H.265 (HEVC) with HDR-10 support</td>
<td>• AAC &lt;br&gt;• Dolby Digital &lt;br&gt;• Dolby Digital Plus</td>
<td>DFXP</td>
<td></td>
</tr>
<tr>
<td>DASH-ISO</td>
<td>MPEG-DASH</td>
<td>MP4</td>
<td>• H.264 (AVC) &lt;br&gt;• H.265 (HEVC) with HDR-10 support</td>
<td>• AAC &lt;br&gt;• Dolby Digital &lt;br&gt;• Dolby Digital Plus</td>
<td>EBU-TT</td>
</tr>
<tr>
<td>CMAF</td>
<td>HLS</td>
<td>CMAF</td>
<td>• H.264 (AVC) &lt;br&gt;• H.265 (HEVC) with HDR-10 support</td>
<td>• AAC &lt;br&gt;• Dolby Digital &lt;br&gt;• Dolby Digital Plus</td>
<td>WebVTT</td>
</tr>
</tbody>
</table>

Creating a SMIL file

When you send an VOD MP4 asset to AWS Elemental MediaPackage, you have to include a Synchronized Multimedia Integration Language (SMIL) file as well. This .smil file acts as a wrapper for all of the files that are part of the asset.

MediaPackage supports the following tags attributes in a .smil file.

**Attributes**
How AWS Elemental MediaPackage works

MediaPackage uses just-in-time format conversion to deliver over-the-top (OTT) video from a single source to a wide variety of playback devices or content delivery networks (CDNs).

The following sections describe how MediaPackage works.

Topics

- Live content processing (p. 8)
- VOD Content Processing (p. 11)
- Live and VOD manifest reference (p. 12)

Live content processing

In the processing flow for live content, encoders send live HLS streams to AWS Elemental MediaPackage. MediaPackage then packages the content, formatting it in response to playback requests from downstream devices.

The following sections describe the live processing flows.

Topics

- General AWS Elemental MediaPackage live processing flow (p. 9)
- Live input redundancy AWS Elemental MediaPackage processing flow (p. 10)
General AWS Elemental MediaPackage live processing flow

Here is the general processing flow for live content in MediaPackage:

1. An upstream encoder (such as AWS Elemental MediaLive) sends an HLS live stream with digest authentication over WebDAV to the MediaPackage channel input URL, and includes the channel's access credentials (as supplied in MediaPackage). If you're using input redundancy, the encoder sends two identical HLS live streams to MediaPackage, one to each input URL on the channel. MediaPackage uses the stream from one input URL as the source content. If MediaPackage stops receiving content on the active input URL, it automatically switches to the other input URL for source content. Additionally, AWS scales resources up and down to handle the incoming traffic.

   For more information, see Live input redundancy AWS Elemental MediaPackage processing flow (p. 10).

   **Note**
   To allow support for features like time-shifted viewing, MediaPackage stores all received content for a limited time. This stored content is only available for playback if it falls within the **startover window** that's defined on the endpoint. Stored content is not available for playback if it's outside the startover window, or if you haven't defined a window on the endpoint. For more information, see Time-shifted viewing reference in AWS Elemental MediaPackage (p. 118).

2. A downstream device requests content from MediaPackage through the endpoint output URL. A downstream device is either a video player or a content delivery network (CDN). The output URL is associated with an endpoint for a specific streaming format (either Apple HLS, DASH-ISO, CMAF, or Microsoft Smooth Streaming).

3. When MediaPackage receives the playback request from the downstream device, it dynamically packages the stream according to the settings that you specified on the endpoint. Packaging can include adding encryption and configuring audio, video, and subtitles or captions track outputs.

4. MediaPackage delivers the output stream over HTTPS to the requesting device. As with input, AWS scales resources up and down to handle changes in traffic.

5. MediaPackage logs activity through Amazon CloudWatch. You can view information like the number of content requests and amount of content that MediaPackage has received or delivered. For information about viewing MediaPackage metrics in CloudWatch, see Monitoring AWS Elemental MediaPackage with Amazon CloudWatch metrics (p. 140).

Throughout the content input and output processes, MediaPackage detects and mitigates potential infrastructure failures before they become a problem for viewers.

The following illustration shows the overall process.
Live input redundancy AWS Elemental MediaPackage processing flow

Achieve input redundancy in MediaPackage by sending two streams to separate input URLs on a channel in MediaPackage. One of the streams becomes the primary, active source of content for the endpoints, while the other continues to passively receive content. If MediaPackage stops receiving content from the active stream, it switches over to the other input stream so that content playback isn't interrupted.

If you use MediaPackage with AWS Elemental MediaLive (for example), here is the flow of input redundancy:

1. You create a channel in MediaPackage, as described in Creating a channel (p. 40). When MediaPackage provisions the channel, it creates two input URLs for the channel. If you're not using input redundancy, you can send a stream to either input URL. There is no requirement that you send content to both URLs.

   **Note**

   When input redundancy became available, MediaPackage added a second input URL to existing channels and updated the existing URL to a new format. You can use either the existing URL or the new URLs for content input.

2. You create an endpoint in MediaPackage as described in Creating an endpoint (p. 44).
Important
If you use short output segments, depending on your playback device, you might see
buffering when MediaPackage switches inputs. You can reduce buffering by using the time
delay feature on the endpoint. Be aware that using a time delay introduces latency to end-to-end
delivery of the content. For information about enabling time delay, see Creating an
endpoint (p. 44).

3. You create an input and channel in AWS Elemental MediaLive, and you add a MediaPackage output
group to the channel in MediaLive. For more information, see Creating a Channel from Scratch in the
AWS Elemental MediaLive User Guide.

If you use an HLS output group in AWS Elemental MediaLive, the input loss action on the HLS group’s
settings must be set to pause the output if the service doesn’t receive input. If MediaLive sends a black
frame or some other filler frame when it’s missing input, then MediaPackage can’t tell when segments
are missing, and subsequently can’t perform failover. For more information about setting the input
loss action in MediaLive, see Fields for the HLS Group in the AWS Elemental MediaLive User Guide.

Important
If you use a different encoder (not AWS Elemental MediaLive) and you send two separate
streams to the same channel in AWS Elemental MediaPackage, the streams must have
identical encoder settings. Otherwise, input redundancy might not work correctly and
playback could be interrupted if the inputs switch.

4. You start the channel in AWS Elemental MediaLive to send the streams to MediaPackage.

5. MediaPackage receives content on both of the input URLs, but only one of the streams is used
for source content at a time. If the active stream is missing any segments, then MediaPackage
automatically fails over to the other stream. MediaPackage continues to use this stream until failover
is needed again.

The formula that is used to determine if an input is missing segments is based on the segment lengths
on the inputs and the endpoints. If an input is missing segments and quickly recovers, an endpoint
with longer segment lengths won’t switch inputs. This might result in different endpoints on the
channel using different inputs (if one endpoint switches and the other doesn’t). This is expected
behavior and should not affect the content workflow.

VOD Content Processing

In the processing flow for video on demand (VOD) content, AWS Elemental MediaPackage ingests
file-based video content from Amazon S3. MediaPackage then packages the content, formatting it in
response to playback requests from downstream devices.

Here is the general processing flow for VOD content in AWS Elemental MediaPackage:

1. From the MediaPackage asset, you initiate ingest of the source content from an Amazon S3 bucket.
   This process can take several minutes. You receive an Amazon CloudWatch event when ingest is
   complete and the playback URLs are live.
2. A downstream device requests content from MediaPackage through the packaging configuration URL
   on the asset. A downstream device is either a video player or a content delivery network (CDN). The
   URL is associated with a configuration for a specific streaming format (either Apple HLS, DASH-ISO,
   CMAF, or Microsoft Smooth).
3. When MediaPackage receives the playback request from the downstream device, it dynamically
   packages the stream according to the settings that you specified in the packaging configuration.
   Packaging can include adding encryption and configuring audio, video, and subtitles or captions track
   outputs.
4. MediaPackage delivers the output stream over HTTPS to the requesting device. As with input, AWS
   scales resources up and down to handle changes in traffic.
MediaPackage logs activity through Amazon CloudWatch. You can view information like the number of content requests and amount of content that MediaPackage has delivered. For information about viewing MediaPackage VOD metrics in CloudWatch, see Monitoring AWS Elemental MediaPackage with Amazon CloudWatch metrics (p. 140).

Throughout the content input and output processes, MediaPackage detects and mitigates potential infrastructure failures before they become a problem for viewers.

## Live and VOD manifest reference

AWS Elemental MediaPackage delivers live and video on demand (VOD) manifests to requesting devices. A live manifest indicates that the content is not complete. New content continually becomes available through the playback endpoint. Alternatively, a VOD manifest indicates that the program is complete, or will be complete at a specified time in the future.

This section describes the differences in live and VOD manifests, and explains when MediaPackage delivers each manifest type.

### Manifest properties

These are the main properties in a manifest that determine if it's live or VOD:

- For HLS and CMAF VOD manifests, `EXT-X-ENDLIST` is at the end of the bitrate manifests. In live manifests, this tag isn't present.
- For MPEG-DASH VOD manifests, `type="static"` is in the MPD properties. In live manifests, the type is dynamic.
- For Microsoft Smooth VOD manifests, `IsLive=True` is in the SmoothStreamingMedia properties. In live manifests, the `IsLive` property isn't present.

For VOD, the scrub bar on playback devices also often shows that the program has a limited duration. This duration is equal to the length of the current manifest. If a playback request defines a specific playback window, this duration is equal to the length of that playback window.

### When a manifest is VOD

MediaPackage delivers a VOD manifest when the content of the program is complete. MediaPackage considers a program complete under the following conditions:

**There's an end parameter in the past.**

When a playback request includes an end parameter that's set in the past, the content is complete. No new content is added to it. MediaPackage delivers a static, VOD manifest to downstream devices.

For information about start and end parameters in playback requests, see Time-shifted viewing reference in AWS Elemental MediaPackage (p. 118).

**The manifest that the upstream encoder delivers to MediaPackage includes an `EXT-X-ENDLIST` tag.**

When you stop the output from your encoder, the manifest that it sends to MediaPackage includes an `EXT-X-ENDLIST` tag. This tag tells MediaPackage that the content is complete, and no new content will be added. MediaPackage delivers a static, VOD manifest to downstream devices.

**Note**

If you manually stop an AWS Elemental MediaLive channel when one or both pipelines to MediaPackage are stopped, MediaLive doesn't include `EXT-X-ENDLIST` in the HLS manifest to MediaPackage. MediaPackage continues to produce a live manifest.
Features of AWS Elemental MediaPackage

MediaPackage supports the following features:

Audio

MediaPackage supports multi-language audio inputs, as well as the following audio codecs:
- AAC stereo
- Dolby AC3 and E-AC3 (Dolby Digital and Dolby Digital+)

MediaPackage accepts these codecs from the input source and passes them through to the output stream.

Important
MediaPackage does not support audio-only inputs. The stream configuration from the encoder must include at least one video track.

Captions

Your embedded source captions can be CEA-608 captions, CEA-708 captions, or both CEA-608 and CEA-708.

Important
MediaPackage currently doesn't support manifest signaling for closed captions. The CEA-608 or CEA-708 stream will be present in the video.

CDN Authorization

MediaPackage supports content delivery network (CDN) authorization. For information, see CDN authorization in AWS Elemental MediaPackage (p. 89).

DRM

MediaPackage supports content protection through digital rights management (DRM). For information, see Content encryption and DRM in AWS Elemental MediaPackage (p. 93).

HLS Rendition Groups

MediaPackage supports rendition groups for incoming and outgoing HLS content. For information about output redundancy groups, see Rendition groups reference in AWS Elemental MediaPackage (p. 113).

Live to VOD

Use the harvest job resource to extract a live-to-VOD (video on demand) asset from a live content stream. MediaPackage creates the asset and stores it in an Amazon S3 bucket. You can use the VOD functionality in MediaPackage to deliver the asset to end users.

Input Redundancy

Input redundancy is available with only live workflows in MediaPackage.
MediaPackage creates two input URLs on every channel so that you can create input redundancy by sending two identical streams to the same channel. For information about how input redundancy works, see Live input redundancy AWS Elemental MediaPackage processing flow (p. 10).

Subtitles

MediaPackage supports input WebVTT text-based subtitles. MediaPackage translates the subtitles to the appropriate format based on the packager that is used on the endpoint:

- For HLS and CMAF: WebVTT is passed through
- For DASH: subtitles are translated to EBU-TT
- For Microsoft Smooth Streaming: subtitles are translated to DFXP

Time-shift Viewing

Time-shift viewing is available with only live workflows in MediaPackage.

MediaPackage allows playback of a stream at a time earlier than the current time. Start-over, catch-up TV, and time delay are all supported. For more information about setting up time-shift capabilities, see Time-shifted viewing reference in AWS Elemental MediaPackage (p. 118).

Video

MediaPackage supports the input H.264 video codec and passes it through to the output stream. Common Media Application Format (CMAF) endpoints in MediaPackage also support H.265/HEVC and HDR-10, following the Apple specification to applicable playback devices.

Important

MediaPackage requires at least one video track to be present in the stream configuration from the encoder. The service does not support audio-only ingest.

Whitelisting

Whitelisting is available with only live workflows in MediaPackage.

MediaPackage supports restricting network access to the endpoint. To take advantage of this feature, you must enter the allowed IP addresses on the endpoint. For more information about adding whitelisting information, see Access control fields (p. 48).

Related services

- **Amazon CloudFront** is a global content delivery network (CDN) service that securely delivers data and videos to your viewers. Use CloudFront to deliver content with the best possible performance. For more information, see Amazon CloudFront.
- **Amazon CloudWatch** is a monitoring service for AWS Cloud resources and the applications that you run on AWS. Use CloudWatch to track metrics such as content input and output request counts. For more information, see Amazon CloudWatch.
- **AWS Elemental MediaLive** is a live video processing service that encodes high-quality live video streams for broadcast television and multi-screen devices. Use MediaLive to encode content streams and send them to MediaPackage for packaging. For more information about how encoders (such as MediaLive) work with MediaPackage, see How AWS Elemental MediaPackage works (p. 8).
- **AWS Identity and Access Management (IAM)** is a web service that helps you securely control access to AWS resources for your users. Use IAM to control who can use your AWS resources (authentication) and what resources users can use in which ways (authorization). For more information, see Setting up (p. 16).
- **AWS Elemental MediaTailor** is a scalable ad insertion service that runs in the AWS Cloud. Use MediaTailor to serve targeted ads to viewers. For more information, see AWS Elemental MediaTailor.
- **Amazon Simple Storage Service** (Amazon S3) is a storage service. Pull video on demand (VOD) assets from Amazon S3, or store live-to-VOD assets in the bucket of your choice. For more information, see
Getting started with VOD content delivery in AWS Elemental MediaPackage (p. 36) and Getting started with live-to-VOD content delivery in AWS Elemental MediaPackage (p. 29).

Accessing MediaPackage

You can access MediaPackage through the console, AWS Command Line Interface (AWS CLI), or MediaPackage REST API.

- Console access:
  
  https://console.aws.amazon.com/mediapackage/

- AWS CLI endpoint:
  
  aws mediapackage

- MediaPackage REST API endpoint:
  
  https://mediapackage.<region>.amazonaws.com

Pricing for MediaPackage

As with other AWS products, there are no contracts or minimum commitments for using MediaPackage. You are charged only for AWS resources that your account uses. Pricing is pay-as-you-go and consists of the following:

- A per GB charge for received content
- A per GB charge for content that is streamed out of MediaPackage

Content that is cached and served from a content delivery network (CDN) does not incur this per GB charge.

For detailed pricing information, see MediaPackage Pricing.

Regions for MediaPackage

To reduce latency in your applications, MediaPackage offers a regional endpoint for your requests. To view the list of AWS Regions where MediaPackage is available, see MediaPackage Regions.
Setting up AWS Elemental MediaPackage

Before you use AWS Elemental MediaPackage for the first time, you must sign up for AWS if you don't already have an AWS account. Next, you must create AWS Identity and Access Management (IAM) users to allow access to MediaPackage.

Topics

• Signing up for AWS (p. 16)
• Creating an admin IAM user (p. 16)
• Creating a non-admin IAM user (p. 17)
• Allowing AWS Elemental MediaPackage to access other AWS services (p. 21)
• (Optional) setting up encryption (p. 26)

Signing up for AWS

If you do not have an AWS account, complete the following steps to create one.

To sign up for an AWS account

2. Follow the online instructions.

Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

Creating an admin IAM user

When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account root user and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

In this procedure, you use the AWS account root user to create your first IAM user. You add this IAM user to an Administrators group, to ensure that you have access to all services and their resources in your account. The next time that you access your AWS account, you should sign in with the credentials for this IAM user.

To create users with limited permissions, see Creating a non-admin IAM user (p. 17).

To create an administrator user for yourself and add the user to an administrators group (console)

1. Sign in to the IAM console as the account owner by choosing Root user and entering your AWS account email address. On the next page, enter your password.
Creating a non-admin IAM user

Users in the Administrators group for an account have access to all AWS services and resources in that account. This section describes how to create users with permissions that are limited to AWS Elemental MediaPackage.

Topics
- Step 1: (Optional) create a policy for Amazon CloudFront (p. 18)
- Step 2: (Optional) create a policy for AWS Elemental MediaPackage VOD (p. 18)
- Step 3: Create users (p. 20)

Note
We strongly recommend that you adhere to the best practice of using the Administrator IAM user that follows and securely lock away the root user credentials. Sign in as the root user only to perform a few account and service management tasks.

2. In the navigation pane, choose Users and then choose Add user.
3. For User name, enter Administrator.
4. Select the check box next to AWS Management Console access. Then select Custom password, and then enter your new password in the text box.
5. (Optional) By default, AWS requires the new user to create a new password when first signing in. You can clear the check box next to User must create a new password at next sign-in to allow the new user to reset their password after they sign in.
6. Choose Next: Permissions.
7. Under Set permissions, choose Add user to group.
8. Choose Create group.
9. In the Create group dialog box, for Group name enter Administrators.
10. Choose Filter policies, and then select AWS managed - job function to filter the table contents.
11. In the policy list, select the check box for AdministratorAccess. Then choose Create group.

Note
You must activate IAM user and role access to Billing before you can use the AdministratorAccess permissions to access the AWS Billing and Cost Management console. To do this, follow the instructions in step 1 of the tutorial about delegating access to the billing console.

12. Back in the list of groups, select the check box for your new group. Choose Refresh if necessary to see the group in the list.
13. Choose Next: Tags.
14. (Optional) Add metadata to the user by attaching tags as key-value pairs. For more information about using tags in IAM, see Tagging IAM entities in the IAM User Guide.
15. Choose Next: Review to see the list of group memberships to be added to the new user. When you are ready to proceed, choose Create user.

You can use this same process to create more groups and users and to give your users access to your AWS account resources. To learn about using policies that restrict user permissions to specific AWS resources, see Access management and Example policies.

For information about creating users with limited permissions, see Creating a non-admin IAM user (p. 17).
Step 1: (Optional) create a policy for Amazon CloudFront

If you or your users will create Amazon CloudFront distributions from the AWS Elemental MediaPackage live console, create a policy that allows access to CloudFront.

For more information about using CloudFront with AWS Elemental MediaPackage, see Working with Content Delivery Networks (CDNs) (p. 164).

To use the JSON policy editor to create a policy

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation column on the left, choose Policies.
   
   If this is your first time choosing Policies, the Welcome to Managed Policies page appears. Choose Get Started.
3. At the top of the page, choose Create policy.
4. Choose the JSON tab.
5. Enter the following JSON policy document:

   ```json
   {
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Action": [
           "cloudfront:GetDistribution",
           "cloudfront:CreateDistributionWithTags",
           "cloudfront:UpdateDistribution",
           "cloudfront:CreateDistribution",
           "cloudfront:TagResource",
           "tag:GetResources"
         ],
         "Resource": "*"
       }
     ]
   }
   ```


   Note
   
   You can switch between the Visual editor and JSON tabs any time. However, if you make changes or choose Review policy in the Visual editor tab, IAM might restructure your policy to optimize it for the visual editor. For more information, see Policy restructuring in the IAM User Guide.

7. On the Review policy page, enter a Name and an optional Description for the policy that you are creating. Review the policy Summary to see the permissions that are granted by your policy. Then choose Create policy to save your work.

Step 2: (Optional) create a policy for AWS Elemental MediaPackage VOD

If you or your users will be using video on demand (VOD) functionality in MediaPackage, create a policy that allows access to resources for the mediapackage-vod service.
The following sections describe how to create a policy that allows all actions, and one that allows read-only rights. You can customize the policies by adding or removing actions to fit your workflows.

### Policy for full VOD access

This policy allows the user to perform all actions on all VOD resources.

**To use the JSON policy editor to create a policy**

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation column on the left, choose **Policies**.
   
   If this is your first time choosing **Policies**, the Welcome to Managed Policies page appears. Choose Get Started.
3. At the top of the page, choose **Create policy**.
4. Choose the **JSON** tab.
5. Enter the following JSON policy document:

   ```json
   {
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Action": "mediapackage-vod:*",
         "Resource": "*"
       }
     ]
   }
   ```
6. Choose **Review policy**.

   **Note**
   You can switch between the Visual editor and JSON tabs any time. However, if you make changes or choose Review policy in the Visual editor tab, IAM might restructure your policy to optimize it for the visual editor. For more information, see Policy restructuring in the IAM User Guide.
7. On the Review policy page, enter a **Name** and an optional **Description** for the policy that you are creating. Review the policy Summary to see the permissions that are granted by your policy. Then choose **Create policy** to save your work.

### Policy for read-only VOD access

This policy allows the user to view all VOD resources.

**To use the JSON policy editor to create a policy**

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation column on the left, choose **Policies**.
   
   If this is your first time choosing **Policies**, the Welcome to Managed Policies page appears. Choose Get Started.
3. At the top of the page, choose **Create policy**.
4. Choose the **JSON** tab.
5. Enter the following JSON policy document:

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "mediapackage-vod:List*",
                "mediapackage-vod:Describe*"
            ],
            "Resource": "*"
        }
    ]
}
```


   **Note**

   You can switch between the Visual editor and JSON tabs any time. However, if you make changes or choose Review policy in the Visual editor tab, IAM might restructure your policy to optimize it for the visual editor. For more information, see Policy restructuring in the IAM User Guide.

7. On the Review policy page, enter a Name and an optional Description for the policy that you are creating. Review the policy Summary to see the permissions that are granted by your policy. Then choose Create policy to save your work.

---

### Step 3: Create users

Create IAM users for the individuals who require access to AWS Elemental MediaPackage. Add the appropriate policies to users to ensure that they have the right level of permissions.

#### To create users who can access MediaPackage

1. In the navigation pane of the IAM console, choose Users, and then choose Add user.
2. For User name, enter the name that the user will use to sign in to MediaPackage.
3. Select the check box next to AWS Management Console access, select Custom password, and then enter the new user's password in the box. You can optionally select Require password reset to force the user to create a password the next time the user signs in.
4. Choose Next: Permissions.
5. On the Set permissions for user page, choose Attach existing policies directly.
6. In the policy list, search for and add the policy with the appropriate MediaPackage permissions level.

   - For access to live functionality:
     - Use **AWSElementalMediaPackageFullAccess** to allow the user to perform all actions on all live resources in MediaPackage.
     - Use **AWSElementalMediaPackageReadOnly** to provide the user read-only rights for all live resources in MediaPackage.
   - For access to video on demand (VOD) functionality, use the policy that you created in Step 2: (Optional) create a policy for AWS Elemental MediaPackage VOD (p. 18).
7. Add policies to allow the MediaPackage console to make calls to Amazon CloudWatch on the user's behalf. Without these policies, the user is able to use the service's API only (not the console). Choose one of these options:

   - Use **ReadOnlyAccess** to allow MediaPackage to communicate with CloudWatch, and also provide the user read-only access to all AWS services on your account.
Allowing AWS Elemental MediaPackage to access other AWS services

Some features require you to allow MediaPackage to access other AWS services, such as Amazon Simple Storage Service (Amazon S3) and Secrets Manager. To allow this access, create an IAM role and policy with the appropriate permissions. The following steps describe how to create roles and policies for MediaPackage features.

Topics

• Step 1: Create a policy (p. 21)
• Step 2: Create a role (p. 25)
• Step 3: Modify the trust relationship (p. 25)

Step 1: Create a policy

The IAM policy defines the permissions that AWS Elemental MediaPackage requires to access other services.

• For video on demand (VOD) workflows, create a policy that allows MediaPackage to read from the Amazon S3 bucket, verify the billing method, and retrieve content. For the billing method, MediaPackage must verify that the bucket does not require the requester to pay for requests. If the bucket has requestPayment enabled, MediaPackage can't ingest content from that bucket.
• For live-to-VOD workflows, create a policy that allows MediaPackage to read from the Amazon S3 bucket and store the live-to-VOD asset in it.
• For content delivery network (CDN) authorization, create a policy that allows MediaPackage to read from a secret in Secrets Manager.

The following sections describe how to create these policies.

Topics

• Amazon S3 access for VOD workflows (p. 21)
• Policy for live-to-VOD workflows (p. 22)
• Secrets Manager access for CDN authorization (p. 23)

Amazon S3 access for VOD workflows

If you're using MediaPackage to ingest a VOD asset from an Amazon S3 bucket and to package and deliver that asset, you need a policy that allows you to do these things in Amazon S3:

• Use CloudWatchReadOnlyAccess, CloudWatchEventsReadOnlyAccess, and CloudWatchLogsReadOnlyAccess to allow MediaPackage to communicate with CloudWatch, and limit the user's read-only access to CloudWatch.

8. (Optional) If this user will create Amazon CloudFront distributions from the MediaPackage console, attach the policy that you created in Step 1: (Optional) create a policy for Amazon CloudFront (p. 18).
9. Choose Next: Review to see the list of policies to be added to the new user. When you are ready to proceed, choose Create user.
Step 1: Create a policy

- **GetObject**: MediaPackage can retrieve the VOD asset from the bucket.
- **GetBucketLocation**: MediaPackage can retrieve the Region for the bucket. The bucket must be in the same region as the MediaPackage VOD resources.
- **GetBucketRequestPayment**: MediaPackage can retrieve the payment request information. MediaPackage uses this information to verify that the bucket doesn't require the requester to pay for the content requests.

If you also use MediaPackage for live-to-VOD asset harvesting, add the **PutObject** action to the policy. For more information the required policy for live-to-VOD workflows, see [Policy for live-to-VOD workflows](#).

**To use the JSON policy editor to create a policy**

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation column on the left, choose **Policies**.
   
   If this is your first time choosing **Policies**, the **Welcome to Managed Policies** page appears. Choose **Get Started**.
3. At the top of the page, choose **Create policy**.
4. Choose the **JSON** tab.
5. Enter the following JSON policy document:

   ```json
   {
     "Version": "2012-10-17",
     "Statement": [
       {
         "Action": [
           "s3:GetObject",
           "s3:GetBucketLocation",
           "s3:GetBucketRequestPayment",
           "s3:ListBucket"
         ],
         "Resource": [
           "arn:aws:s3:::{bucket_name}/*",
           "arn:aws:s3:::{bucket_name}"
         ],
         "Effect": "Allow"
       }
     ]
   }
   ```
6. Choose **Review policy**.

   **Note**
   
   You can switch between the **Visual editor** and **JSON** tabs any time. However, if you make changes or choose **Review policy** in the **Visual editor** tab, IAM might restructure your policy to optimize it for the visual editor. For more information, see [Policy restructuring](#) in the IAM User Guide.

7. On the **Review policy** page, enter a **Name** and an optional **Description** for the policy that you are creating. Review the policy **Summary** to see the permissions that are granted by your policy. Then choose **Create policy** to save your work.

**Policy for live-to-VOD workflows**

If you use MediaPackage to harvest a live-to-VOD asset from a live stream, you need a policy that allows you to do these things in Amazon S3:
• **PutObject**: MediaPackage can save the VOD asset in the bucket.
• **GetBucketLocation**: MediaPackage can retrieve the Region for the bucket. The bucket must be in the same AWS Region as the MediaPackage VOD resources.

If you also use MediaPackage for VOD asset delivery, add these actions to the policy: **GetObject** and **GetBucketRequestPayment**. For more information about the required policy for VOD workflows, see [Amazon S3 access for VOD workflows](p. 21)

**To use the JSON policy editor to create a policy**
1. Sign in to the AWS Management Console and open the IAM console at `https://console.aws.amazon.com/iam/`.
2. In the navigation column on the left, choose **Policies**.
3. At the top of the page, choose **Create policy**.
4. Choose the **JSON** tab.
5. Enter the following JSON policy document:

```
{
   "Version": "2012-10-17",
   "Statement": [
   {
      "Action": [
        "s3:PutObject",
        "s3:ListBucket",
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3:::{bucket_name}/*",
        "arn:aws:s3:::{bucket_name}"
      ],
      "Effect": "Allow"
   }
   ]
}
```
6. Choose **Review policy**.

**Note**
You can switch between the **Visual editor** and **JSON** tabs any time. However, if you make changes or choose **Review policy** in the **Visual editor** tab, IAM might restructure your policy to optimize it for the visual editor. For more information, see [Policy restructuring](in the IAM User Guide).

7. On the **Review policy** page, enter a **Name** and an optional **Description** for the policy that you are creating. Review the policy **Summary** to see the permissions that are granted by your policy. Then choose **Create policy** to save your work.

**Secrets Manager access for CDN authorization**

If you use content delivery network (CDN) authorization headers to restrict access to your endpoints in MediaPackage, you need a policy that allows you to do these things in Secrets Manager:

• **GetSecretValue**: MediaPackage can retrieve the encrypted authorization code from a version of the secret.
• **DescribeSecret**: MediaPackage can retrieve the details of the secret, excluding encrypted fields.
• **ListSecrets**: MediaPackage can retrieve a list of secrets in the AWS account.
• **ListSecretVersionIds**: MediaPackage can retrieve all of the versions that are attached to the specified secret.

**Note**
You don't need a separate policy for each secret that you store in Secrets Manager. If you create a policy like the one described in the following procedure, MediaPackage can access all secrets in your account in this Region.

**To use the JSON policy editor to create a policy**

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation column on the left, choose **Policies**.
   
   If this is your first time choosing **Policies**, the **Welcome to Managed Policies** page appears. Choose **Get Started**.
3. At the top of the page, choose **Create policy**.
4. Choose the **JSON** tab.
5. Enter the following JSON policy document:

   ```json
   {
   "Version": "2012-10-17",
   "Statement": [
   {
   "Effect": "Allow",
   "Action": [
   "secretsmanager:GetSecretValue",
   "secretsmanager:DescribeSecret",
   "secretsmanager:ListSecrets",
   "secretsmanager:ListSecretVersionIds"
   ],
   "Resource": ["*"]
   },
   {
   "Effect": "Allow",
   "Action": [
   "iam:GetRole",
   "iam:PassRole"
   ],
   "Resource": "*
   }
   ]
   }
   ```

6. Choose **Review policy**.

   **Note**
   You can switch between the **Visual editor** and **JSON** tabs any time. However, if you make changes or choose **Review policy** in the **Visual editor** tab, IAM might restructure your policy to optimize it for the visual editor. For more information, see **Policy restructuring** in the **IAM User Guide**.

7. On the **Review policy** page, enter a **Name** and an optional **Description** for the policy that you are creating. Review the policy **Summary** to see the permissions that are granted by your policy. Then choose **Create policy** to save your work.
Step 2: Create a role

An IAM role is an IAM identity that you can create in your account that has specific permissions. An IAM role is similar to an IAM user in that it is an AWS identity with permissions policies that determine what the identity can and cannot do in AWS. However, instead of being uniquely associated with one person, a role is intended to be assumable by anyone who needs it. Also, a role does not have standard long-term credentials such as a password or access keys associated with it. Instead, when you assume a role, it provides you with temporary security credentials for your role session. Create a role that AWS Elemental MediaPackage assumes when ingesting source content from Amazon S3.

When you create the role, you choose EC2 as the trusted entity that can assume the role because AWS Elemental MediaPackage isn't available for selection. In Step 3: Modify the trust relationship (p. 25), you change the trusted entity to MediaPackage.

To create the service role for an EC2 trusted entity (IAM console)

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane of the IAM console, choose Roles, and then choose Create role.
3. Choose the AWS service role type, and then choose EC2 trusted entity.
4. Choose the EC2 use case. Then choose Next: Permissions.
5. On the Attach permissions policies page, search for and choose the policy that you created in Step 1: Create a policy (p. 21). Then choose Next: Tags and Next: Review.
6. (Optional) Set a permissions boundary. This is an advanced feature that is available for service roles, but not service-linked roles.
   Expand the Set permissions boundary section and choose Use a permissions boundary to control the maximum role permissions. IAM includes a list of the AWS managed and customer managed policies in your account. Select the policy to use for the permissions boundary or choose Create policy to open a new browser tab and create a new policy from scratch. For more information, see step 4 in the procedure Creating IAM policies in the IAM User Guide. After you create the policy, close that tab and return to your original tab to select the policy to use for the permissions boundary.
7. Choose Next: Tags.
8. (Optional) Add metadata to the user by attaching tags as key-value pairs. For more information about using tags in IAM, see Tagging IAM entities in the IAM User Guide.
9. Choose Next: Review.
10. If possible, enter a role name or role name suffix to help you identify the purpose of this role. Role names must be unique within your AWS account. They are not distinguished by case. For example, you cannot create roles named both PRODROLE and prodrole. Because various entities might reference the role, you cannot edit the name of the role after it has been created.
11. (Optional) For Role description, enter a description for the new role.
12. Review the role and then choose Create role.

Step 3: Modify the trust relationship

The trust relationship defines what entities can assume the role that you created in the section called “Step 2: Create a role” (p. 25). When you created the role and established the trusted relationship, you chose EC2 as the trusted entity. Modify the role so that the trusted relationship is between your AWS account and AWS Elemental MediaPackage.

To change the trust relationship to MediaPackage

1. Access the role that you created in Step 2: Create a role (p. 25).
If you're not already displaying the role, in the navigation pane of the IAM console, choose **Roles**. Search for and choose the role that you created.

2. On the **Summary** page for the role, choose **Trust relationships**.

3. Choose **Edit trust relationship**.


The policy document should now look like this:

```json
{
"Version": "2012-10-17",
"Statement": [
{
  "Sid": "",
  "Effect": "Allow",
  "Principal": {
    "Service": "mediapackage.amazonaws.com"
  },
  "Action": "sts:AssumeRole"
}
]
}
```

5. Choose **Update Trust Policy**.

6. On the **Summary** page, make a note of the value in **Role ARN**. You use this ARN when you ingest source content for video on demand (VOD) workflows. The ARN looks like this:

```
arn:aws:iam::111122223333:role/RoleName
```

In the example, `111122223333` is your AWS account number.

---

**(Optional) setting up encryption**

Protect your content from unauthorized use through content encryption and digital rights management (DRM). AWS Elemental MediaPackage uses the **AWS Secure Package and Encoder Key Exchange (SPEKE) API** to facilitate content encryption and decryption by a DRM provider. Using SPEKE, the DRM provider supplies encryption keys to AWS Elemental MediaPackage through the SPEKE API. The DRM provider also supplies licenses to supported media players for decryption. For more information about how SPEKE is used with services and features running in the cloud, see **AWS cloud-based architecture** in the **Secure Packager and Encoder Key Exchange API Specification guide**.

To encrypt content, you must have a DRM solution provider and be set up to use encryption. For more information, see the section called "Content encryption and DRM" (p. 93).
Getting started with AWS Elemental MediaPackage

The following sections describe how to quickly get started receiving and sending content with AWS Elemental MediaPackage.

Topics
- Getting started with live content delivery in AWS Elemental MediaPackage (p. 27)
- Getting started with live-to-VOD content delivery in AWS Elemental MediaPackage (p. 29)
- Getting started with VOD content delivery in AWS Elemental MediaPackage (p. 36)

Getting started with live content delivery in AWS Elemental MediaPackage

This Getting Started tutorial shows you how to use the AWS Elemental MediaPackage console to create a channel and endpoints for streaming live videos.

Topics
- Prerequisites (p. 27)
- Step 1: Access AWS Elemental MediaPackage (p. 27)
- Step 2: Create a channel (p. 27)
- Step 3: Create endpoints (p. 28)
- (Optional) step 4: Monitor AWS Elemental MediaPackage activity (p. 29)
- Step 5: Clean up (p. 29)

Prerequisites

Before you can use AWS Elemental MediaPackage, you need an AWS account and the appropriate permissions to access, view, and edit MediaPackage components. Make sure that your system administrator has completed the steps in Setting up (p. 16), and then return to this tutorial.

For supported live inputs and codecs, see Live supported codecs and input types (p. 3).

Step 1: Access AWS Elemental MediaPackage

Using your IAM credentials, sign in to the AWS Elemental MediaPackage console:

https://console.aws.amazon.com/mediapackage/

Step 2: Create a channel

The channel is the first component in AWS Elemental MediaPackage. It represents the input to MediaPackage for incoming live content from an encoder such as AWS Elemental MediaLive.
MediaPackage does not require that you supply any customer data. There are no fields in channels where there is an expectation that you will provide customer data.

To create a channel

1. On the MediaPackage Channels page, choose Create channel.
2. For ID, enter a name that describes the channel, such as channelHLS1. The ID is the primary identifier for the channel, and must be unique for your account in the AWS Region. Supported characters are letters, numbers, underscore (_), and dash (-). You can't use spaces in the ID.
3. Keep the defaults for the remaining fields, and then choose Create channel.

MediaPackage displays the new channel's details page.
4. On the channel's details page, note the values for Input URL, Username, and Password. If you're using input redundancy, you need this information for both input URLs. If you're sending only one stream to the channel, you can note the information for either input URL.

MediaPackage securely generates the WebDAV user names and passwords when it creates the channel. If you need to change these credentials, see Rotating credentials on an input URL (p. 42).

Provide the information from these fields to the person in charge of the upstream encoder. In the stream configuration in the encoder, this person must enter the destination as the input URL, and the WebDAV credentials as the channel's user name and password. The upstream encoder must use digest authentication and push WebDAV over HTTPS to MediaPackage, and include these credentials. If you're using input redundancy, the input streams to this channel must have identical encoder settings. For more information about setting up source streams for input redundancy, see Live input redundancy AWS Elemental MediaPackage processing flow (p. 10).

Step 3: Create endpoints

The endpoint is attached to a channel, and represents the output of the live content. You can associate multiple endpoints to a single channel. Each endpoint gives players and downstream CDNs (such as Amazon CloudFront) access to the content for playback.

AWS Elemental MediaPackage does not require that you supply any customer data. There are no fields in endpoints where there is an expectation that you will provide customer data.

To create an endpoint

1. On the Channels page, choose the channel that the endpoint will be associated with.
2. On the details page for the channel, choose either Add and edit endpoint or Add endpoints if there are no existing endpoints.
3. For ID, enter a name that describes the endpoint, such as HLSendpoint1. The ID is the primary identifier for the endpoint, and must be unique for your account in the AWS Region. Supported characters are letters, numbers, underscore (_), and dash (-). You can't use spaces in the ID.
4. Keep the defaults for the remaining fields, and then choose Save endpoints.

MediaPackage displays the channel's details page, including the endpoint that you just created.
5. On the channel's details page, note the value in the URL field for the endpoint. Provide this information to the person in charge of the downstream device (CDN or player). In the downstream device, this person must enter the request destination as the endpoint's URL.
(Optional) step 4: Monitor AWS Elemental MediaPackage activity

Use Amazon CloudWatch to track AWS Elemental MediaPackage activity, such as the counts of bytes that MediaPackage has received and sent, response times, and request counts. Metrics are grouped first by the service namespace, and then by the various dimension combinations within each namespace.

To view metrics using the CloudWatch console
2. In the navigation pane, choose Metrics.
3. Under All metrics, choose the AWS/MediaPackage namespace.
4. Select the metric dimension to view the metrics (for example, choose channel to view metrics per channel).

For a list of AWS Elemental MediaPackage metrics, see AWS Elemental MediaPackage live content metrics (p. 141).

Step 5: Clean up

To avoid extraneous charges, be sure to delete all unnecessary channels and endpoints. You must delete all endpoints on a channel before the channel can be deleted.

To delete an endpoint
1. On the Channels page, choose the channel that the endpoint is associated with.
2. On the channel details page, choose the name of the endpoint that you want to delete.
3. On the endpoint details page, choose Delete endpoint.
4. On the Delete Endpoints page, choose Save all.

To delete a channel
1. On the Channels page, choose the channel using one the following methods:
   - Choose the channel name
   - Select the check box next to the channel name
2. Choose Delete selected or Delete channel.
3. In the confirmation dialog box, choose Delete.

AWS Elemental MediaPackage removes the channel and all associated endpoints.

Getting started with live-to-VOD content delivery in AWS Elemental MediaPackage

This Getting Started tutorial shows you how to use the AWS Elemental MediaPackage console to create a live-to-VOD (video on demand) asset and make it available for playback.

To deliver live-to-VOD content, you do these three main things:
• Ingest a live HLS content stream into MediaPackage
• Extract a VOD asset from the stream
• Make the asset available for playback

Note
You’re not required to use MediaPackage to deliver your live-to-VOD asset to viewers. This tutorial is meant as an illustration of how you can use MediaPackage to complete the live-to-VOD workflow.

The following sections are a guided tutorial for becoming familiar with these three things and other supporting actions.

Topics
• Prerequisites (p. 30)
• Step 1: Access AWS Elemental MediaPackage (p. 30)
• Step 2: Ingest live content (p. 30)
• Step 3: Extract a VOD asset (p. 32)
• Step 4: (Optional) output VOD content (p. 32)
• (Optional) step 5: Monitor AWS Elemental MediaPackage activity (p. 34)
• Step 6: Clean up (p. 35)

Prerequisites

Before you can use AWS Elemental MediaPackage, you need an AWS account and the appropriate permissions to access, view, and edit MediaPackage components. Make sure that your system administrator has completed the following steps in Setting up (p. 16), and then return to this tutorial:

• To create an AWS account, see Signing up for AWS (p. 16).
• To allow non-administrative users access to MediaPackage, see Creating a non-admin IAM user (p. 17).
• To allow MediaPackage to access your Amazon S3 bucket to save and retrieve the live-to-VOD asset, see Allowing AWS Elemental MediaPackage to access other AWS services (p. 21).

Step 1: Access AWS Elemental MediaPackage

Using your IAM credentials, sign in to the AWS Elemental MediaPackage console:

https://console.aws.amazon.com/mediapackage/

Step 2: Ingest live content

To ingest a live content stream into MediaPackage and extract a video on demand (VOD) asset from it, create a channel and endpoint. The channel is the entry point to MediaPackage, and the endpoint provides MediaPackage access to the stream so that it can extract the VOD asset. The following sections describe how to use the MediaPackage console to create a channel and endpoint.

Create a channel

The channel is the first component in AWS Elemental MediaPackage. It represents the input to MediaPackage for incoming live content from an encoder such as AWS Elemental MediaLive.
MediaPackage does not require that you supply any customer data. There are no fields in channels where there is an expectation that you will provide customer data.

To create a channel

1. On the MediaPackage **Channels** page, choose **Create channel**.
2. For **ID**, enter a name that describes the channel, such as **channelHLS1**. The ID is the primary identifier for the channel, and must be unique for your account in the AWS Region. Supported characters are letters, numbers, underscore (_), and dash (-). You can't use spaces in the ID.
3. Keep the defaults for the remaining fields, and then choose **Create channel**.

MediaPackage displays the new channel's details page.

4. On the channel's details page, note the values for **Input URL**, **Username**, and **Password**. If you're using input redundancy, you need this information for both input URLs. If you're sending only one stream to the channel, you can note the information for either input URL.

MediaPackage securely generates the WebDAV user names and passwords when it creates the channel. If you need to change these credentials, see Rotating credentials on an input URL (p. 42).

Provide the information from these fields to the person in charge of the upstream encoder. In the stream configuration in the encoder, this person must enter the destination as the input URL, and the WebDAV credentials as the channel's user name and password. The upstream encoder must use digest authentication and push WebDAV over HTTPS to MediaPackage, and include these credentials. If you're using input redundancy, the input streams to this channel must have identical encoder settings. For more information about setting up source streams for input redundancy, see Live input redundancy AWS Elemental MediaPackage processing flow (p. 10).

Create an endpoint

The endpoint is attached to a channel, and represents the output of the live content. When you create a harvest job to extract a VOD asset from the live content, you have to indicate what endpoint you're extracting from. You can harvest assets from clear (unencrypted) or encrypted HLS and DASH endpoints, and the endpoint must have a startover window defined. If you have only encrypted endpoints, see the Creating live-to-VOD assets with AWS Elemental MediaPackage (p. 84) feature reference.

AWS Elemental MediaPackage does not require that you supply any customer data. There are no fields in endpoints where there is an expectation that you will provide customer data.

To create an endpoint

1. On the **Channels page**, choose the channel that the endpoint will be associated with.
2. On the details page for the channel, choose either **Add and edit endpoint** or **Add endpoints** if there are no existing endpoints.
3. For **ID**, enter a name that describes the endpoint, such as **HLSendpoint1**. The ID is the primary identifier for the endpoint, and must be unique for your account in the AWS Region. Supported characters are letters, numbers, underscore (_), and dash (-). You can't use spaces in the ID.
4. Keep the defaults for the remaining fields, and then choose **Save endpoints**.

MediaPackage displays the channel's details page, including the endpoint that you just created.

5. On the channel's details page, note the value in the **URL** field for the endpoint. Provide this information to the person in charge of the downstream device (CDN or player). In the downstream device, this person must enter the request destination as the endpoint's URL.
Step 3: Extract a VOD asset

To extract a live-to-VOD asset from a live content stream, create a harvest job. The harvest job identifies what endpoint the asset is being harvested from, the start and end of the asset, and where MediaPackage saves the asset after it's been harvested.

To create a harvest job

1. On the MediaPackage Harvest jobs page, choose Create job.
2. For ID, enter a name that describes the harvest job, such as gamehighlights. The ID is the primary identifier for the job. You can reuse the ID after the harvest job expires from your account. Supported characters are letters, numbers, underscore (_), and dash (-). You can't use spaces in the ID.
3. For Origin endpoint, select the endpoint for the live content stream that you're extracting a VOD asset from. The endpoint must serve clear (unencrypted) or encrypted DASH or HLS content. If you want to extract from encrypted live content, see Creating live-to-VOD assets with AWS Elemental MediaPackage (p. 84).
4. For Date and time format, keep the default.
5. For Start date and time and End date and time, enter the start and end dates and times for the extracted VOD asset. The start time must be after the live stream has started and before the current time ("now"). The end time must be in the past.
   
   Note
   "Now" is the current time according to the program date time (PDT), when it's present in the source content from the encoder.

6. For IAM role, enter the IAM role that allows MediaPackage to write to Amazon S3. For help with the role, see Allowing AWS Elemental MediaPackage to access other AWS services (p. 21).
7. For S3 bucket name, select the bucket where you want MediaPackage to store the live-to-VOD asset.
8. For Manifest key, enter the path within the Amazon S3 bucket and file name for the master manifest of the live-to-VOD asset. MediaPackage creates a directory based on the path that you enter.
   
   Important
   The manifest key must be unique. When you use the same manifest key for multiple harvest jobs, the newest playlist for the asset overwrites existing playlists. The only time you should reuse a manifest key is when you are harvesting the same content, such as if there was a problem with a previous harvest of the content.

9. Choose Create job.

When MediaPackage processes the harvest job, it sends a CloudWatch event when the job fails or succeeds. The event includes the details of the harvest job. If the job fails, the event includes information about why. This information is available only in the CloudWatch event. For example events, see Harvest job notification events (p. 153).

Step 4: (Optional) output VOD content

To use MediaPackage to make the live-to-VOD asset available for playback, create a packaging group, packaging configuration, and asset resource. The asset ingests the live-to-VOD asset from the Amazon S3 bucket. A packaging group holds one or more packaging configurations, which define the output format and settings.
Create a packaging group

A packaging group holds one or more packaging configurations. The packaging configurations enable you to define what kind of VOD outputs you want. To apply these output definitions, associate a packaging group to multiple assets.

Example

You have 15 pieces of source content. You want to serve them all as DASH, HLS, and encrypted HLS outputs. To do this, you define one packaging group with DASH, HLS, and encrypted HLS packaging configurations. You then associate that group to the asset resources that represent these pieces of content. You don't have to create new configurations for each asset.

AWS Elemental MediaPackage does not require that you supply any customer data. There are no fields in packaging groups where there is an expectation that you will provide customer data.

To create a packaging group

1. On the AWS Elemental MediaPackage Packaging groups page, choose Create.
2. For ID, enter a name that describes the group, such as gamehighlights. The ID is the primary identifier for the group, and must be unique for your account in this AWS Region. Supported characters are letters, numbers, underscore (_), and dash (-). You can't use spaces in the ID.
3. Choose Create.

Create a packaging configuration

A packaging configuration specifies how the output manifest is configured, such as stream selection limitations and ordering.

AWS Elemental MediaPackage does not require that you supply any customer data. There are no fields in packaging configurations where there is an expectation that you will provide customer data.

To create a packaging configuration

1. On the AWS Elemental MediaPackage Packaging groups page, choose the group that you just created.
2. On the details page for the packaging group, choose either Add or remove configuration or Add configuration if there are no existing packaging configurations.
3. On the Add packaging configurations page, choose Add, and then choose New configuration.
4. For ID, enter a name that describes the configuration, such as hls_highlights. The ID is the primary identifier for the configuration, and must be unique for your account in this AWS Region. Supported characters are letters, numbers, underscore (_), and dash (-). You can't use spaces in the ID.
5. Keep the defaults for the remaining fields, and then choose Save.

Create an asset

An asset resource is how MediaPackage ingests, packages, and serves VOD content. The asset is associated with one or more packaging configurations. Downstream devices send playback requests to specific packaging configurations on the asset.

AWS Elemental MediaPackage does not require that you supply any customer data. There are no fields in assets where there is an expectation that you will provide customer data.
To create an asset and ingest source content

1. From your Amazon S3 buckets, determine what file you're using as source content. Make note of the following:
   - The name of the Amazon S3 bucket where the file is stored
   - The full path for the file, such as S3://bucket/path/source-file-name
   - The IAM role that allows AWS Elemental MediaPackage to read from Amazon S3
2. On the AWS Elemental MediaPackage console, go to the Assets page, and then choose Ingest asset.
3. For S3 bucket name, select the bucket where your source content is stored.
4. For IAM role, choose Use existing role and select the IAM role that allows AWS Elemental MediaPackage to read from Amazon S3.
5. For Filename, enter the path within the Amazon S3 bucket and name for the source content.
6. For Packaging group, select the group that you created in Step 2: Create a packaging group (p. 36).
7. Choose Ingest assets.

Provide playback URLs

After creating the asset resource, AWS Elemental MediaPackage prepares to serve the packaged manifests to viewers. This happens in the background and might take some time depending on the size and complexity of the source content, but is usually less than a few minutes. The URLs of the manifests are available immediately on the asset's details page, but content is not yet available for playback.

After the processing for each manifest is complete, MediaPackage sends an Amazon CloudWatch event to your account.

On the asset, MediaPackage provides a URL for each packaging configuration. This URL is how downstream devices (CDN or playback device) request VOD content from MediaPackage.

To get playback URLs

1. On the MediaPackage console, go to the Assets page and choose the ID of the asset that you created in Step 4: Create an asset (p. 37).
2. On the asset's detail page, get the URL for each packaging configuration.
3. Provide the URLs to the person in charge of the downstream device (CDN or player). In the downstream device, this person must enter the request destination as the URL from the corresponding packaging configuration.

Each URL is stable. It never changes during the lifetime of the combination of this asset and packaging configuration. Provide the URL to the person in charge of the downstream device (CDN or player). In the downstream device, this person must use the asset's URL as the request destination.

(Optional) step 5: Monitor AWS Elemental MediaPackage activity

Use Amazon CloudWatch to track AWS Elemental MediaPackage activity, such as the counts of bytes that MediaPackage has received and sent, response times, and request counts. Metrics are grouped first by the service namespace, and then by the various dimension combinations within each namespace.

To view metrics using the CloudWatch console

2. In the navigation pane, choose **Metrics**.
3. Under **All metrics**, choose the **AWS/MediaPackage** namespace.
4. Select the metric dimension to view the metrics (for example, choose `channel` to view metrics per channel).

For a list of AWS Elemental MediaPackage metrics, see *Monitoring AWS Elemental MediaPackage with Amazon CloudWatch metrics (p. 140).*

**Step 6: Clean up**

To avoid incurring extra charges, delete the resources that you're no longer using.

**Note**

Harvest jobs automatically expire off your account and can't be manually deleted.

**Delete live resources**

When you're done ingesting, serving, and harvesting from live content, delete the channel and endpoint. You must delete all endpoints on a channel before you can delete the channel.

**To delete an endpoint**

1. On the **Channels page**, choose the channel that the endpoint is associated with.
2. On the channel details page, choose the name of the endpoint that you want to delete.
3. On the endpoint details page, choose **Delete endpoint**.
4. On the **Delete Endpoints** page, choose **Save all**.

**To delete a channel**

1. On the **Channels** page, choose the channel using one the following methods:
   - Choose the channel name
   - Select the check box next to the channel name
2. Choose **Delete selected** or **Delete channel**.
3. In the confirmation dialog box, choose **Delete**.

AWS Elemental MediaPackage removes the channel and all associated endpoints.

**Delete VOD resources**

When you're done ingesting and serving VOD content, delete the extra resources. If you want to make a specific output unavailable, delete the packaging configuration from the packaging group. If you want to make an asset no longer available for playback from any outputs, delete the asset.

**To delete an asset**

1. On the AWS Elemental MediaPackage console, go to the **Assets** page, and then choose the **ID** of the asset.
2. On the asset's details page, choose **Delete**.
3. In the confirmation dialog box, choose **Delete**.
To delete a packaging configuration

1. On the AWS Elemental MediaPackage console, go to the Packaging groups page.
2. Choose the ID of the group that has the configuration that you want to delete.
3. On the packaging group's details page, in the Packaging configurations section, locate the configuration and choose its ID.
4. On the packaging configuration's details page, choose Delete.
5. In the confirmation dialog box, choose Delete.

Getting started with VOD content delivery in AWS Elemental MediaPackage

This Getting Started tutorial shows you how to use the AWS Elemental MediaPackage console to ingest video on demand (VOD) content and make it available for playback.

Topics

- Prerequisites (p. 36)
- Step 1: Access AWS Elemental MediaPackage (p. 36)
- Step 2: Create a packaging group (p. 36)
- Step 3: Create a packaging configuration (p. 37)
- Step 4: Create an asset (p. 37)
- Step 5: Provide playback URLs (p. 38)
- (Optional) step 6: Monitor AWS Elemental MediaPackage activity (p. 38)
- Step 7: Clean up (p. 39)

Prerequisites

Before you can use AWS Elemental MediaPackage VOD capability, you must meet the following conditions:

- You have an AWS account and the appropriate permissions to access, view, and edit MediaPackage components. Make sure that your system administrator has completed the steps in Setting up (p. 16), and then return to this tutorial.
- You have file-based source content in one or more Amazon S3 buckets.

For supported VOD inputs and codecs, see VOD supported codecs and input types (p. 6).

Step 1: Access AWS Elemental MediaPackage

Using your IAM credentials, sign in to the AWS Elemental MediaPackage console:

https://<region>.console.aws.amazon.com/mediapackage/home

Step 2: Create a packaging group

A packaging group holds one or more packaging configurations. The packaging configurations enable you to define what kind of VOD outputs you want. To apply these output definitions, associate a packaging group to multiple assets.
Example

You have 15 pieces of source content. You want to serve them all as DASH, HLS, and encrypted HLS outputs. To do this, you define one packaging group with DASH, HLS, and encrypted HLS packaging configurations. You then associate that group to the asset resources that represent these pieces of content. You don’t have to create new configurations for each asset.

AWS Elemental MediaPackage does not require that you supply any customer data. There are no fields in packaging groups where there is an expectation that you will provide customer data.

To create a packaging group

1. On the AWS Elemental MediaPackage Packaging groups page, choose Create.
2. For ID, enter a name that describes the group, such as gamehighlights. The ID is the primary identifier for the group, and must be unique for your account in this AWS Region. Supported characters are letters, numbers, underscore (_), and dash (-). You can’t use spaces in the ID.
3. Choose Create.

Step 3: Create a packaging configuration

A packaging configuration specifies how the output manifest is configured, such as stream selection limitations and ordering.

AWS Elemental MediaPackage does not require that you supply any customer data. There are no fields in packaging configurations where there is an expectation that you will provide customer data.

To create a packaging configuration

1. On the Packaging groups page, choose the group that you just created.
2. On the details page for the packaging group, choose either Add or remove configuration or Add configuration if there are no existing packaging configurations.
3. On the Add packaging configurations page, choose Add, and then choose New configuration.
4. For ID, enter a name that describes the configuration, such as hls_highlights. The ID is the primary identifier for the configuration, and must be unique for your account in this AWS Region. Supported characters are letters, numbers, underscore (_), and dash (-). You can’t use spaces in the ID.
5. Keep the defaults for the remaining fields, and then choose Save.

Step 4: Create an asset

An asset resource is how MediaPackage ingests, packages, and serves VOD content. The asset is associated with one or more packaging configurations. Downstream devices send playback requests to specific packaging configurations on the asset.

AWS Elemental MediaPackage does not require that you supply any customer data. There are no fields in assets where there is an expectation that you will provide customer data.

To create an asset and ingest source content

1. From your Amazon S3 buckets, determine what file you’re using as source content. Make note of the following:
   - The name of the Amazon S3 bucket where the file is stored
   - The full path for the file, such as S3://bucket/path/source-file-name
- The IAM role that allows AWS Elemental MediaPackage to read from Amazon S3
- On the AWS Elemental MediaPackage console, go to the **Assets** page, and then choose **Ingest asset**.
- For **S3 bucket name**, select the bucket where your source content is stored.
- For **IAM role**, choose **Use existing role** and select the IAM role that allows AWS Elemental MediaPackage to read from Amazon S3.
- For **Filename**, enter the path within the Amazon S3 bucket and name for the source content.
- For **Packaging group**, select the group that you created in Step 2: Create a packaging group (p. 36).
- Choose **Ingest assets**.

### Step 5: Provide playback URLs

After creating the asset resource, AWS Elemental MediaPackage prepares to serve the packaged manifests to viewers. This happens in the background and might take some time depending on the size and complexity of the source content, but is usually less than a few minutes. The URLs of the manifests are available immediately on the asset's details page, but content is not yet available for playback.

After the processing for each manifest is complete, MediaPackage sends an Amazon CloudWatch event to your account.

On the asset, MediaPackage provides a URL for each packaging configuration. This URL is how downstream devices (CDN or playback device) request VOD content from MediaPackage.

#### To get playback URLs

1. On the MediaPackage console, go to the **Assets** page and choose the **ID** of the asset that you created in Step 4: Create an asset (p. 37).
2. On the asset's detail page, get the URL for each packaging configuration.
3. Provide the URLs to the person in charge of the downstream device (CDN or player). In the downstream device, this person must enter the request destination as the URL from the corresponding packaging configuration.

Each URL is stable. It never changes during the lifetime of the combination of this asset and packaging configuration. Provide the URL to the person in charge of the downstream device (CDN or player). In the downstream device, this person must use the asset's URL as the request destination.

### (Optional) step 6: Monitor AWS Elemental MediaPackage activity

Use Amazon CloudWatch to track AWS Elemental MediaPackage activity, such as the counts of bytes that MediaPackage has received and sent, response times, and request counts. Metrics are grouped first by the service namespace, and then by the various dimension combinations within each namespace.

#### To view metrics using the CloudWatch console

2. In the navigation pane, choose **Metrics**.
3. Under **All metrics**, choose the **AWS/MediaPackage** namespace.
4. Select the metric dimension to view the metrics (for example, choose **channel** to view metrics per channel).
For a list of AWS Elemental MediaPackage metrics, see AWS Elemental MediaPackage VOD content metrics (p. 146).

Step 7: Clean up

To avoid incurring extra charges, delete your VOD resources. If you want to make a specific output unavailable, delete the packaging configuration from the packaging group. If you want to make an asset no longer available for playback from any outputs, delete the asset.

To delete an asset

1. On the AWS Elemental MediaPackage console, go to the Assets page, and then choose the ID of the asset.
2. On the asset's details page, choose Delete.
3. In the confirmation dialog box, choose Delete.

To delete a packaging configuration

1. On the AWS Elemental MediaPackage console, go to the Packaging groups page.
2. Choose the ID of the group that has the configuration that you want to delete.
3. On the packaging group's details page, in the Packaging configurations section, locate the configuration and choose its ID.
4. On the packaging configuration's details page, choose Delete.
5. In the confirmation dialog box, choose Delete.
Delivering live content from AWS Elemental MediaPackage

AWS Elemental MediaPackage uses the following resources for live content:

- **Channels** are the entry point for your live streams from upstream encoders.
  
  For supported live inputs and codecs, see [Live supported codecs and input types](p. 3).

- **Endpoints** tell AWS Elemental MediaPackage how to package outbound content. Endpoints are associated with channels and hold encryption, stream, and packaging settings.

The following sections describe how to use these resources to manage live content in AWS Elemental MediaPackage.

**Topics**

- Working with channels in AWS Elemental MediaPackage (p. 40)
- Working with endpoints in AWS Elemental MediaPackage (p. 44)

Working with channels in AWS Elemental MediaPackage

A channel holds all the information that AWS Elemental MediaPackage requires to receive a live content stream from a source such as AWS Elemental MediaLive or another encoder. The channel receives content, and after packaging it, outputs it through an endpoint to downstream devices (such as video players or CDNs) that request the content.

After you create a channel, AWS Elemental MediaPackage provides a pair of input URLs that are fixed for the lifetime of the channel, regardless of any failures or upgrades that might happen over time. The output of the upstream encoder points to the URLs for stream delivery to MediaPackage.

For supported live inputs and codecs, see [Live supported codecs and input types](p. 3).

**Topics**

- Creating a channel (p. 40)
- Viewing channel details (p. 41)
- Editing a channel (p. 42)
- Rotating credentials on an input URL (p. 42)
- Deleting a channel (p. 43)
- Adding an endpoint to a channel (p. 43)

Creating a channel

Create a channel to start receiving content streams. Later, you add an endpoint to the channel. This endpoint is the access point for content playback requests.
You can use the AWS Elemental MediaPackage console, the AWS CLI, or the MediaPackage API to create a channel. For information about creating a channel through the AWS CLI or MediaPackage API, see the AWS Elemental MediaPackage API Reference.

When you're creating a channel, do not put sensitive identifying information like customer account numbers into free-form fields such as the Name field. This includes when you work with AWS Elemental MediaPackage using the console, REST API, AWS CLI, or AWS SDKs. Any data that you enter into MediaPackage might get picked up for inclusion in diagnostic logs or Amazon CloudWatch Events.

**To create a channel (console)**

1. Open the MediaPackage console at https://console.aws.amazon.com/mediapackage/.
2. On the Channels page, choose Create channel.
3. For ID, type a name that describes the channel. The ID is the primary identifier for the channel, and must be unique for your account in the region.
4. (Optional) For Description, type any descriptive text that helps you to identify the channel.
5. For Input type, choose Apple HLS.
6. Choose Create channel.

AWS Elemental MediaPackage displays the new channel's details page.

The channel is active and can start receiving content as soon as it's created. AWS Elemental MediaPackage scales resources up and down to allow the right amount of capacity for your traffic. If you're using input redundancy and one of the inputs stops sending content, then MediaPackage automatically switches to the other input for the source content. For more information about how input redundancy works, see Live input redundancy AWS Elemental MediaPackage processing flow (p. 10).

When you're creating a channel, you will receive an error if you exceed the quotas on the account. An error similar to Too many requests, please try again. Resource limit exceeded means that either you have exceeded the API request quotas, or you have already reached the maximum number of channels allowed on your account. If this is your first channel, or if you think you received this error wrongfully, use the Service Quotas console to request quota increases. For more information about quotas in AWS Elemental MediaPackage, see Quotas in AWS Elemental MediaPackage (p. 168).

**Viewing channel details**

View all channels that are configured in AWS Elemental MediaPackage, or view the details of a specific channel, including the endpoints that are associated with it.

You can use the AWS Elemental MediaPackage console, the AWS CLI, or the MediaPackage API to view channel details. For information about viewing details about a channel through the AWS CLI or MediaPackage API, see the AWS Elemental MediaPackage API Reference.

**To view channels (console)**

1. Open the MediaPackage console at https://console.aws.amazon.com/mediapackage/.
2. If the Channels page doesn't appear, on the AWS Elemental MediaPackage home page, choose Skip and go to console.
   
   All existing channels are displayed on the console.
3. (Optional) To adjust your viewing preferences (such as page size and properties that are displayed), choose Preferences.
4. To view more information about a specific channel, choose the name of the channel that you want to view.
AWS Elemental MediaPackage displays important information such as the values for **Input URL** and the **WebDAV Username** and **Password** for each input URL. Provide this information for the upstream encoder stream destination settings. If you're using input redundancy, provide the information for both input URLs. If you're sending only one stream to the channel, you can provide the information for either input URL. For information about how input redundancy works, see Live input redundancy AWS Elemental MediaPackage processing flow (p. 10).

**Note**
All channels have two input URLs. For channels that existed before input redundancy, AWS Elemental MediaPackage created two new input URLs. You can use either the old or new URLs for inputs to the channel.

If you created an Amazon CloudFront distribution from the AWS Elemental MediaPackage console, you will also see the high-level distribution information (such as status and ID) from the channel. When you add an endpoint in MediaPackage, an origin is also added to the distribution, and you will see the CloudFront CDN URL from the channel's details page as well.

**Editing a channel**

Edit a channel's description for easier identification later.

You can edit the description on a channel or enable Amazon CloudFront distribution creation from the AWS Elemental MediaPackage console. For information about creating a distribution from AWS Elemental MediaPackage, see Creating a Distribution from AWS Elemental MediaPackage (p. 165).

**Note**
To make changes to an existing distribution (even if it was created from AWS Elemental MediaPackage), go to the Amazon CloudFront console.

You can use the AWS Elemental MediaPackage console, the AWS CLI, or the MediaPackage API to edit a channel. For information about editing a channel through the AWS CLI or MediaPackage API, see the AWS Elemental MediaPackage API Reference.

When you're editing a channel, do not put sensitive identifying information like customer account numbers into free-form fields such as the **Name** field. This includes when you work with AWS Elemental MediaPackage using the console, REST API, AWS CLI, or AWS SDKs. Any data that you enter into MediaPackage might get picked up for inclusion in diagnostic logs or Amazon CloudWatch Events.

**To edit a channel (console)**

1. Open the MediaPackage console at https://console.aws.amazon.com/mediapackage/.
2. If the **Channels** page doesn't appear, on the AWS Elemental MediaPackage home page, choose **Skip and go to console**.
3. On the **Channels** page, choose the name of the channel that you want to edit.
4. On the channel's details page, choose **Edit channel**.
5. Make the changes that you want.
6. Choose **Save changes**.

**Rotating credentials on an input URL**

Rotate credentials on an input URL to generate a new WebDAV user name and password.

You can use the AWS Elemental MediaPackage console or the MediaPackage API to rotate credentials. For information about rotating credentials through the MediaPackage API, see the AWS Elemental MediaPackage API Reference.
To rotate credentials (console)

1. Open the MediaPackage console at https://console.aws.amazon.com/mediapackage/.
2. If the Channels page doesn't appear, on the AWS Elemental MediaPackage home page, choose Skip and go to console.
3. On the Channels page, choose the name of the channel that holds the input URL that you're rotating the credentials for.
4. On the channel's details page, choose the input URL that you're rotating credentials for, and then choose Rotate credentials.
5. To confirm that you want to generate a new user name and password, choose Rotate.

AWS Elemental MediaPackage displays the new credentials.

Deleting a channel

Delete a channel to stop AWS Elemental MediaPackage from receiving further content. You must delete the channel's endpoints (as described in Deleting an endpoint (p. 63)) before you can delete the channel.

You can use the AWS Elemental MediaPackage console, the AWS CLI, or the MediaPackage API to delete a channel. For information about deleting a channel through the AWS CLI or MediaPackage API, see the AWS Elemental MediaPackage API Reference.

To delete a channel (console)

1. Open the MediaPackage console at https://console.aws.amazon.com/mediapackage/.
2. If the Channels page doesn't appear, on the AWS Elemental MediaPackage home page, choose Skip and go to console.
3. On the Channels page, choose the channel using one the following methods:
   - Choose the channel name
   - Select the check box next to the channel name
4. Choose Delete selected or Delete channel.

If there is an Amazon CloudFront distribution associated with the channel, select the CloudFront link in the confirmation dialog box to go to the CloudFront console to delete the distribution. AWS Elemental MediaPackage will not delete the distribution when the channel is deleted. For help deleting in CloudFront, see Deleting a distribution in the Amazon CloudFront Developer Guide.
5. In the confirmation dialog box in AWS Elemental MediaPackage, choose Delete to proceed with the channel deletion.

Adding an endpoint to a channel

Add an endpoint to a channel to allow downstream video players and content distribution networks (CDNs) to start requesting content playback.

You can use the AWS Elemental MediaPackage console, the AWS CLI, or the MediaPackage API to add an endpoint to a channel. For information about adding through the AWS CLI or MediaPackage API, see the AWS Elemental MediaPackage API Reference.

For instructions on adding endpoints to a channel from the AWS Elemental MediaPackage console, see the section called “Working with endpoints” (p. 44).
Working with endpoints in AWS Elemental MediaPackage

An endpoint defines a single delivery point of a channel. The endpoint holds all the information that is needed for AWS Elemental MediaPackage to integrate with a player or content distribution network (CDN) such as Amazon CloudFront. Configure the endpoint to output content in one of the available stream formats:

- Apple HLS – packages content to Apple HTTP Live Streaming (HLS)
- Microsoft Smooth – packages content for Microsoft Smooth Streaming players
- CMAF – packages content to devices that support Apple HLS fragmented MP4 (fMP4)
- DASH-ISO – packages content for the DASH-ISO ABR streaming protocol

Additionally, the endpoint holds information about digital rights management (DRM) and encryption integration, stream bitrate presentation order, and more.

Topics
- Creating an endpoint (p. 44)
- Viewing all endpoints associated with a channel (p. 62)
- Viewing a single endpoint (p. 62)
- Editing an endpoint (p. 63)
- Deleting an endpoint (p. 63)
- Previewing an endpoint (p. 64)

Creating an endpoint

Create an endpoint on a channel to define how AWS Elemental MediaPackage prepares content for delivery. Content can’t be served from a channel until it has an endpoint. If you’re using input redundancy, each endpoint receives content from one input URL at a time. If MediaPackage performs a failover on the inputs for one input URL, the endpoints automatically start receiving content from the other input URL. For more information about input redundancy and failover, see Live input redundancy AWS Elemental MediaPackage processing flow (p. 10).

When you create an endpoint, AWS Elemental MediaPackage assigns it a public URL that is fixed for the lifetime of the endpoint, regardless of any failures or upgrades that might happen over time. This URL is how the player or CDN accesses the stream from the endpoint.

You can use the AWS Elemental MediaPackage console, the AWS CLI, or the MediaPackage API to create an endpoint. For information about creating an endpoint through the AWS CLI or MediaPackage API, see the AWS Elemental MediaPackage API Reference.

AWS Elemental MediaPackage does not require that you supply any customer data. There are no fields in endpoints where there is an expectation that you will provide customer data.

Topics
- Creating an HLS endpoint (p. 45)
- Creating a Microsoft smooth streaming endpoint (p. 49)
- Creating a common media application format (CMAF) endpoint (p. 52)
- Creating a DASH endpoint (p. 57)
Creating an HLS endpoint

Create an endpoint that formats content for devices that support Apple HLS.

To create an apple HLS endpoint (console)

1. Access the channel that the endpoint will be associated with, as described in Viewing channel details (p. 41).
2. On the details page for the channel, choose either Add and edit endpoint or Add endpoints if there are no existing endpoints.
3. Complete the fields as described in the following topics:
   - New endpoint fields (p. 45)
   - Packager settings fields (p. 46)
   - Encryption fields (p. 47)
   - Access control fields (p. 48)
   - Streams to include fields (p. 49)
4. Choose Save endpoints.

If you enabled Amazon CloudFront distribution creation from the AWS Elemental MediaPackage console and this is your first endpoint on the channel, MediaPackage adds an origin to the distribution. You can view the CloudFront CDN URL and endpoint information in the endpoints section of the channel's details page.

The endpoint is active and can deliver content as soon as requests are sent to its URL endpoints. AWS Elemental MediaPackage scales resources up and down to allow the right amount of capacity for your traffic.

When you're creating an endpoint, you will receive an error if you exceed the quotas on the account. An error similar to Too many requests, please try again. Resource limit exceeded means that either you have exceeded the API request quotas, or you have already reached the maximum number of endpoints allowed on this channel. If you think you received this error wrongfully, use the Service Quotas console to request quota increases. For more information about quotas in AWS Elemental MediaPackage, see Quotas in AWS Elemental MediaPackage (p. 168).

New endpoint fields

When you're creating an endpoint, do not put sensitive identifying information like customer account numbers into free-form fields such as the Name field. This includes when you work with AWS Elemental MediaPackage using the console, REST API, AWS CLI, or AWS SDKs. Any data that you enter into MediaPackage might get picked up for inclusion in diagnostic logs or Amazon CloudWatch Events.

1. For ID, enter a name that describes the endpoint. The ID is the primary identifier for the endpoint, and must be unique for your account in the AWS Region.
2. (Optional) For Description, enter any descriptive text that helps you to identify the endpoint later.
3. For Manifest name, enter a short string that will be appended to the end of the endpoint URL. The manifest name helps to create a unique path to this endpoint.
4. (Optional) To create a window of the live stream that's available for on-demand viewing, select Startover window and enter the size of the window (in seconds). Viewers can start-over or catch-up on content that falls within the window. For more information about implementing start-over and catch-up TV, see Time-shifted viewing reference in AWS Elemental MediaPackage (p. 118).
5. (Optional) To delay when content is available to players, enter the duration (in seconds) for the delay in Time delay. The minimum time is 5 seconds. The maximum time is 86,400 seconds (24 hours).
Use time delay to redefine the live point and make content available at a time that equals "now" minus the delay specified. With a 60-second time delay, content that AWS Elemental MediaPackage receives at 12:20 isn’t available until 12:21. Requests for playback at 12:20 will be served with content from 12:19. Likewise, if you’re serving content across time zones, you can set a time delay equal to the time zone difference to make content available at, for example, 8:00 local time.

When you use time delay in conjunction with a startover window, the time delay duration must be less than the startover window duration.

**Tip**
Use a time delay to help reduce buffering during input switching when you’re using input redundancy with short output segments. Note that the delay can increase latency in content playback.

### Packager settings fields

1. For **Type**, choose **Apple HLS**.
2. (Optional) For **Segment duration**, type the duration (in seconds) of each segment. If the value that you type here is different from the input segment size, AWS Elemental MediaPackage rounds segments to the nearest multiple of the input segment duration.
3. (Optional) For **Playlist window duration**, type the total duration (in seconds) of the manifest.
4. (Optional) To group all audio tracks into a single HLS rendition group, select **Use audio rendition group**. For more information about rendition groups, see **Rendition groups reference in AWS Elemental MediaPackage** (p. 113).
5. (Optional) In stream sets with a single video track, to include an additional I-frame only stream along with the other tracks in the manifest, select **Include IFrame only stream**. MediaPackage inserts EXT-I-FRAMES-ONLY tags in the manifest, and then compiles and includes an I-frames only playlist in the stream. This playlist enables player functionality like fast forward and rewind.
6. (Optional) To include EXT-X-PROGRAM-DATE-TIME tags in the output manifest, select **Program date/time interval**, and then type the interval for MediaPackage to insert the tags in the manifest.

The EXT-X-PROGRAM-DATE-TIME tag holds the time of the segment. When program date time (PDT) information is available in the source content, MediaPackage uses this same information on the output content. Otherwise, MediaPackage uses Coordinated Universal Time (UTC) for the PDT.

The PDT information helps downstream players to synchronize the stream to the wall clock, enabling functionality like viewer seek in the playback timeline and time display on the player.

### SCTE-35 Options

The following fields dictate how MediaPackage processes SCTE-35 messages from the input stream. For more information, see **SCTE-35 message options in AWS Elemental MediaPackage** (p. 114).

1. (Optional) In **Ad markers**, choose how ad markers are included in the packaged content.

   Choose from the following:
   - **None** – Omit all SCTE-35 ad markers from the output.
   - **SCTE-35 enhanced** – Generate ad markers and blackout tags based on the SCTE-35 input messages from the input source.
   - **Passthrough** – Copy the SCTE-35 ad markers directly from the input HLS input manifest to the output manifest.

2. (Optional) To add or remove SCTE-35 message types that MediaPackage treats as ads, choose **Customize ad triggers**. If you don’t make a selection here, MediaPackage inserts ad markers in the output manifest based on these message types:
Creating an endpoint

- Splice insert
- Provider advertisement
- Distributor advertisement
- Provider placement opportunity
- Distributor placement opportunity

3. (Optional) To change what ad insertion action MediaPackage takes based on delivery restriction flags in the segmentation descriptors of SCTE-35 messages, choose Ads on delivery restrictions. These are the available options:
   - **None** – MediaPackage doesn't insert any ad markers in the output manifest.
   - **Restricted** – MediaPackage inserts ad markers when there are delivery restrictions in the SCTE-35 message types that you indicated in Customize ad triggers.
   - **Unrestricted** – MediaPackage inserts ad markers when there aren't delivery restrictions in the SCTE-35 message types that you indicated in Customize ad triggers.
   - **Both** – MediaPackage inserts ad markers whether or not there are delivery restrictions in the SCTE-35 message types that you indicated in Customize ad triggers.

Encryption fields

Protect your content from unauthorized use through content encryption and digital rights management (DRM). AWS Elemental MediaPackage uses the AWS Secure Package and Encoder Key Exchange (SPEKE) API to facilitate content encryption and decryption by a DRM provider. Using SPEKE, the DRM provider supplies encryption keys to AWS Elemental MediaPackage through the SPEKE API. The DRM provider also supplies licenses to supported media players for decryption. For more information about how SPEKE is used with services and features running in the cloud, see AWS cloud-based architecture in the Secure Packager and Encoder Key Exchange API Specification guide.

**Note**
To encrypt content, you must have a DRM provider, and be set up to use encryption. For information, see the section called “Content encryption and DRM” (p. 93).

1. To serve content without copyright protection, keep **No encryption** selected.
2. To serve content with copyright protection, choose **Encrypt content** and complete the additional fields as follows:
   a. For **Resource ID**, enter an identifier for the content. The service sends this to the key server to identify the current endpoint. How unique you make this depends on how fine-grained you want access controls to be. The service does not allow you to use the same ID for two simultaneous encryption processes. The resource ID is also known as the content ID.

   The following example shows a resource ID.
   
   MovieNight20171126093045

   b. For **System IDs**, enter unique identifiers for your streaming protocol and DRM system. Provide one system ID. If you do not know your ID, ask your DRM provider.

   c. For **URL**, enter the URL of the API Gateway proxy that you set up to talk to your key server. The API Gateway proxy must reside in the same AWS Region as MediaPackage.

   The following example shows a URL.
   
   https://1wm2dx1f33.execute-api.us-west-2.amazonaws.com/SpekeSample/copyProtection

   d. For **Role ARN**, enter the Amazon Resource Name (ARN) of the IAM role that provides you access to send your requests through API Gateway. Get this from your DRM solution provider.
The following example shows a role ARN.

```
arn:aws:iam::444455556666:role/SpekeAccess
```

e. **Certificate ARN** – (Optional) Enter a 2048 RSA certificate ARN to use for content key encryption. Use this option only if your DRM key provider supports content key encryption. If you use this and your key provider doesn't support it, the event fails.

To enter a certificate ARN here, you must have already imported the corresponding certificate into AWS Certificate Manager. Then enter the certificate ARN from ACM here.

For information about content key encryption, see Preparing and managing certificates for use with content keys (p. 95).

f. For **Encryption method**, choose **Sample-AES** for Apple HLS FairPlay or choose **AES-128** for Apple HLS AES-128.

g. (Optional) For **Constant initialization vector** enter a 128-bit, 16-byte hex value represented by a 32-character string, to be used with the key for encrypting content.

h. (Optional) For **Key rotation interval**, enter the frequency, in seconds, of key changes for live workflows, in which content is streamed real time. The service retrieves content keys before the live content begins streaming, and then retrieves them as needed over the lifetime of the workflow. By default, key rotation is set to 60 seconds, which is equivalent to setting it to 60. To disable key rotation, set this interval to 0 (zero).

The following example setting causes the service to rotate keys every thirty minutes.

```
1800
```

For information about key rotation, see Understanding key rotation behavior (p. 95).

i. (Optional) Select **Repeat EXT-X-KEY** if you want the service to repeat the key before every segment of the manifest. By default, the key is written just once, after the header and before the segments. If you select **Repeat EXT-X-KEY**, the manifest is written as header, key, segment, key, segment, key, and so on, with every segment preceded by the key. Set this according to the needs of the player. Selecting this option might result in an increase in client requests to the DRM server.

**Access control fields**

1. To enable this endpoint to serve content to requesting devices, select **Allow origination**. It is uncommon to disallow origination on an endpoint.

   Typically, the only reason that you won't allow an endpoint to serve content is if it's only being used to harvest VOD content from the live stream. For more information, see Creating live-to-VOD assets with AWS Elemental MediaPackage (p. 84).

2. To serve content to all requesting IP addresses, choose **Allow all incoming clients**. To limit the IP addresses that this endpoint serves, use these fields:

   **Note**
   
   Only IPv4 addresses are allowed.

   a. Select **Restrict by IP address**.

   b. In **Whitelist**, type the IP addresses that this endpoint serves content to.

3. To require that content requests to this endpoint include a valid authorization code, select **Use authorization**. Complete the remaining fields:

   a. In **Secrets role ARN**, enter the ARN for the IAM role that grants MediaPackage access to AWS Secrets Manager.
b. In **CDN identifier secret**, enter the ARN for the authorization code secret in Secrets Manager.

For information about how this authorization works, see [CDN authorization in AWS Elemental MediaPackage](p. 89).

### Streams to include fields

1. (Optional) For **Bitrate order**, choose from the following:
   - **Original** to sort the output streams in the same order that the incoming source uses.
   - **Ascending** to sort the output streams starting with the lowest bitrate and ending with the highest.
   - **Descending** to sort the output streams starting with the highest bitrate and ending with the lowest.

2. To make all incoming streams available for playback from this endpoint, select **Include all incoming streams**.

3. To limit which incoming streams are available for playback from this endpoint, select **Filter incoming streams** and enter filter criteria:
   - (Optional) For **Min video bitrate**, type the minimum bitrate threshold that video tracks must meet to be available for playback from this endpoint.
   - (Optional) For **Max video bitrate**, type the maximum bitrate that video tracks can have to be available for playback from this endpoint.

   The minimum and maximum values take into account only the video bitrates. If the video bitrate is *below the minimum* specified rate, it is not included in the output, regardless of the sum of the bitrates for other tracks. Likewise, if the video bitrate is *below the maximum* specified rate, it is included in the output, regardless of the sum of the bitrates for other tracks.

### Creating a Microsoft smooth streaming endpoint

Create an endpoint that formats content for devices that support Microsoft Smooth Streaming.

**To create a Microsoft smooth endpoint (console)**

1. Access the channel that the endpoint will be associated with, as described in [Viewing channel details](p. 41).

2. On the details page for the channel, choose either **Add and edit endpoint** or **Add endpoints** if there are no existing endpoints.

3. Complete the fields as described in the following topics:
   - **New endpoint fields** (p. 50)
   - **Packager settings fields** (p. 50)
   - **Encryption fields** (p. 50)
   - **Access control fields** (p. 51)
   - **Streams to include fields** (p. 52)

4. Choose **Save endpoints**.

   If you enabled Amazon CloudFront distribution creation from the AWS Elemental MediaPackage console and this is your first endpoint on the channel, MediaPackage adds an origin to the distribution. You can view the CloudFront CDN URL and endpoint information in the endpoints section of the channel's details page.

   The endpoint is active and can deliver content as soon as requests are sent to its URL endpoints. AWS Elemental MediaPackage scales resources up and down to allow the right amount of capacity for your traffic.
When you're creating an endpoint, you will receive an error if you exceed the quotas on the account. An error similar to Too many requests, please try again. Resource limit exceeded means that either you have exceeded the API request quotas, or you have already reached the maximum number of endpoints allowed on this channel. If you think you received this error wrongfully, use the Service Quotas console to request quota increases. For more information about quotas in AWS Elemental MediaPackage, see Quotas in AWS Elemental MediaPackage (p. 168).

New endpoint fields

When you're creating an endpoint, do not put sensitive identifying information like customer account numbers into free-form fields such as the Name field. This includes when you work with AWS Elemental MediaPackage using the console, REST API, AWS CLI, or AWS SDKs. Any data that you enter into MediaPackage might get picked up for inclusion in diagnostic logs or Amazon CloudWatch Events.

1. For ID, enter a name that describes the endpoint. The ID is the primary identifier for the endpoint, and must be unique for your account in the AWS Region.
2. (Optional) For Description, enter any descriptive text that helps you to identify the endpoint later.
3. For Manifest name, enter a short string that will be appended to the end of the endpoint URL. The manifest name helps to create a unique path to this endpoint.
4. (Optional) To create a window of the live stream that's available for on-demand viewing, select Startover window and enter the size of the window (in seconds). Viewers can start-over or catch-up on content that falls within the window. For more information about implementing start-over and catch-up TV, see Time-shifted viewing reference in AWS Elemental MediaPackage (p. 118).
5. (Optional) To delay when content is available to players, enter the duration (in seconds) for the delay in Time delay. The minimum time is 5 seconds. The maximum time is 86,400 seconds (24 hours).

Use time delay to redefine the live point and make content available at a time that equals "now" minus the delay specified. With a 60-second time delay, content that AWS Elemental MediaPackage receives at 12:20 isn't available until 12:21. Requests for playback at 12:20 will be served with content from 12:19. Likewise, if you're serving content across time zones, you can set a time delay equal to the time zone difference to make content available at, for example, 8:00 local time.

When you use time delay in conjunction with a startover window, the time delay duration must be less than the startover window duration.

Tip

Use a time delay to help reduce buffering during input switching when you're using input redundancy with short output segments. Note that the delay can increase latency in content playback.

Packager settings fields

1. For Type, choose Microsoft Smooth.
2. (Optional) For Segment duration, type the duration (in seconds) of each segment. If the value that you type here is different from the input segment size, AWS Elemental MediaPackage rounds segments to the nearest multiple of the input segment duration.
3. (Optional) For Manifest window duration, type the total duration (in seconds) of the manifest.

Encryption fields

Protect your content from unauthorized use through content encryption and digital rights management (DRM). AWS Elemental MediaPackage uses the AWS Secure Package and Encoder Key Exchange (SPEKE) API to facilitate content encryption and decryption by a DRM provider. Using SPEKE, the DRM provider
supplies encryption keys to AWS Elemental MediaPackage through the SPEKE API. The DRM provider also supplies licenses to supported media players for decryption. For more information about how SPEKE is used with services and features running in the cloud, see AWS cloud-based architecture in the Secure Packager and Encoder Key Exchange API Specification guide.

**Note**
To encrypt content, you must have a DRM solution provider and be set up to use encryption. For information, see the section called “Content encryption and DRM” (p. 93).

1. To serve content without copyright protection, keep **No encryption** selected.
2. To serve content with copyright protection, choose **Encrypt content** and complete the additional fields as follows:
   a. For **Resource ID**, enter an identifier for the content. The service sends this to the key server to identify the current endpoint. How unique you make this depends on how fine-grained you want access controls to be. The service does not allow you to use the same ID for two simultaneous encryption processes. The resource ID is also known as the content ID.

   The following example shows a resource ID.

   MovieNight20171126093045

   b. For **System IDs**, enter unique identifiers for your streaming protocol and DRM system. Provide up to one system ID. If you do not know your ID, ask your DRM solution provider.

   c. For **URL**, enter the URL of the API Gateway proxy that you set up to talk to your key server. The API Gateway proxy must reside in the same AWS Region as MediaPackage.

   The following example shows a URL.

   https://1wm2dx1f33.execute-api.us-west-2.amazonaws.com/SpekeSample/copyProtection

   d. For **Role ARN**, enter the Amazon Resource Name (ARN) of the IAM role that provides you access to send your requests through API Gateway. Get this from your DRM solution provider.

   The following example shows a role ARN.

   arn:aws:iam::444455556666:role/SpekeAccess

   e. **Certificate ARN** – (Optional) Enter a 2048 RSA certificate ARN to use for content key encryption. Use this option only if your DRM key provider supports content key encryption. If you use this and your key provider doesn't support it, the event fails.

   To enter a certificate ARN here, you must have already imported the corresponding certificate into AWS Certificate Manager. Then enter the certificate ARN from ACM here.

   For information about key encryption, see Preparing and managing certificates for use with content keys (p. 95).

**Access control fields**

1. To enable this endpoint to serve content to requesting devices, select **Allow origination**. It is uncommon to disallow origination on an endpoint.

   Typically, the only reason that you won't allow an endpoint to serve content is if it's only being used to harvest VOD content from the live stream. For more information, see Creating live-to-VOD assets with AWS Elemental MediaPackage (p. 84).

2. To serve content to all requesting IP address, choose **Allow all incoming clients**. To limit the IP addresses that this endpoint serves, use these fields:
Creating an endpoint

1. Select **Restrict by IP address**.
2. In **Whitelist**, type the IP addresses that this endpoint serves content to.
3. To require that content requests to this endpoint include a valid authorization code, select **Use authorization**. Complete the remaining fields:
   a. In **Secrets role ARN**, enter the ARN for the IAM role that grants MediaPackage access to AWS Secrets Manager.
   b. In **CDN identifier secret**, enter the ARN for the authorization code secret in Secrets Manager.

   For information about how this authorization works, see [CDN authorization in AWS Elemental MediaPackage](p. 89).

Streams to include fields

1. (Optional) For **Bitrate order**, choose from the following:
   - **Original** to sort the output streams in the same order that the incoming source uses.
   - **Ascending** to sort the output streams starting with the lowest bitrate and ending with the highest.
   - **Descending** to sort the output streams starting with the highest bitrate and ending with the lowest.
2. To make all incoming streams available for playback from this endpoint, select **Include all incoming streams**.
3. To limit which incoming streams are available for playback from this endpoint, select **Filter incoming streams** and enter filter criteria:
   - (Optional) For **Min video bitrate**, type the minimum bitrate threshold that video tracks must meet to be available for playback from this endpoint.
   - (Optional) For **Max video bitrate**, type the maximum bitrate that video tracks can have to be available for playback from this endpoint.

   The minimum and maximum values take into account only the video bitrates. If the video bitrate is below the minimum specified rate, it is not included in the output, regardless of the sum of the bitrates for other tracks. Likewise, if the video bitrate is below the maximum specified rate, it is included in the output, regardless of the sum of the bitrates for other tracks.

Creating a common media application format (CMAF) endpoint

Create an endpoint that formats content for devices that support Apple HLS fragmented MP4 (fMP4).

To create a CMAF endpoint (console)

1. Access the channel that the endpoint will be associated with, as described in [Viewing channel details](p. 41).
2. On the details page for the channel, choose either **Add and edit endpoint** or **Add endpoints** if there are no existing endpoints.
3. Complete the fields as described in the following topics:
   - **New endpoint fields** (p. 53)
   - **Packager settings fields** (p. 53)
   - **HLS manifest fields** (p. 54)
   - **Encryption fields** (p. 55)
   - **Access control fields** (p. 56)
   - **Streams to include fields** (p. 57)
4. Choose **Save endpoints**.
If you enabled Amazon CloudFront distribution creation from the AWS Elemental MediaPackage console and this is your first endpoint on the channel, MediaPackage adds an origin to the distribution. You can view the CloudFront CDN URL and endpoint information in the endpoints section of the channel’s details page.

The endpoint is active and can deliver content as soon as requests are sent to its URL endpoints. AWS Elemental MediaPackage scales resources up and down to allow the right amount of capacity for your traffic.

When you’re creating an endpoint, you will receive an error if you exceed the quotas on the account. An error similar to Too many requests, please try again. Resource limit exceeded means that either you have exceeded the API request quotas, or you have already reached the maximum number of endpoints allowed on this channel. If you think you received this error wrongfully, use the Service Quotas console to request quota increases. For more information about quotas in AWS Elemental MediaPackage, see Quotas in AWS Elemental MediaPackage (p. 168).

**New endpoint fields**

When you're creating an endpoint, do not put sensitive identifying information like customer account numbers into free-form fields such as the Name field. This includes when you work with AWS Elemental MediaPackage using the console, REST API, AWS CLI, or AWS SDKs. Any data that you enter into MediaPackage might get picked up for inclusion in diagnostic logs or Amazon CloudWatch Events.

1. For **ID**, enter a name that describes the endpoint. The ID is the primary identifier for the endpoint, and must be unique for your account in the AWS Region.

2. (Optional) For **Description**, enter any descriptive text that helps you to identify the endpoint later.

3. For **Manifest name**, enter a short string that will be appended to the end of the endpoint URL. The manifest name helps to create a unique path to this endpoint.

4. (Optional) To create a window of the live stream that’s available for on-demand viewing, select **Startover window** and enter the size of the window (in seconds). Viewers can start-over or catch-up on content that falls within the window. For more information about implementing start-over and catch-up TV, see Time-shifted viewing reference in AWS Elemental MediaPackage (p. 118).

5. (Optional) To delay when content is available to players, enter the duration (in seconds) for the delay in **Time delay**. The minimum time is 5 seconds. The maximum time is 86,400 seconds (24 hours).

Use time delay to redefine the live point and make content available at a time that equals "now" minus the delay specified. With a 60-second time delay, content that AWS Elemental MediaPackage receives at 12:20 isn't available until 12:21. Requests for playback at 12:20 will be served with content from 12:19. Likewise, if you’re serving content across time zones, you can set a time delay equal to the time zone difference to make content available at, for example, 8:00 local time.

When you use time delay in conjunction with a startover window, the time delay duration must be less than the startover window duration.

**Tip**

Use a time delay to help reduce buffering during input switching when you’re using input redundancy with short output segments. Note that the delay can increase latency in content playback.

**Packager settings fields**

The Packager settings fields hold general information about the endpoint.

1. For **Type**, choose Common Media Application Format (CMAF).
2. (Optional) For **Segment duration**, type the duration (in seconds) of each segment. If the value that you type here is different from the input segment size, AWS Elemental MediaPackage rounds segments to the nearest multiple of the input segment duration.

3. (Optional) For **Segment prefix**, type a custom name for the segments in the HLS child manifest. The segment prefix is prepended to the segment name to create a unique identifier for each segment.

   **Example**

   If the segment prefix is `movie`, a segment from the child manifest is `movie_1_2.ts`.

**HLS manifest fields**

The HLS manifest fields allow you to define values for this manifest.

1. For **ID**, type a name that describes this manifest. The ID is the primary identifier for the manifest, and must be unique for this endpoint.

2. For **Manifest name**, type a string that will be appended to the end of the endpoint URL. The manifest name helps to create a unique path to this manifest on this endpoint. The HLS manifest name overrides the manifest name that you provided in the New Endpoint **Manifest name** field (if applicable).

3. (Optional) For **Playlist window duration**, type the total duration (in seconds) of the manifest.

4. (Optional) Use the following fields to dictate how MediaPackage processes SCTE-35 messages from the input stream. For more information, see **SCTE-35 message options in AWS Elemental MediaPackage** (p. 114).

   a. (Optional) In **Ad markers**, choose how ad markers are included in the packaged content.

      Choose from the following:
      - **None** – Omit all SCTE-35 ad markers from the output.
      - **SCTE-35 enhanced** – Generate ad markers and blackout tags based on the SCTE-35 input messages from the input source.
      - **Passthrough** – Copy the SCTE-35 ad markers directly from the input HLS input manifest to the output manifest.

   b. (Optional) To add or remove SCTE-35 message types that MediaPackage treats as ads, choose **Customize ad triggers**. If you don't make a selection here, MediaPackage inserts ad markers in the output manifest based on these message types:

      - Splice insert
      - Provider advertisement
      - Distributor advertisement
      - Provider placement opportunity
      - Distributor placement opportunity

   c. (Optional) To change what ad insertion action MediaPackage takes based on delivery restriction flags in the segmentation descriptors of SCTE-35 messages, choose **Ads on delivery restrictions**. These are the available options:

      - **None** – MediaPackage doesn't insert any ad markers in the output manifest.
      - **Restricted** – MediaPackage inserts ad markers when there are delivery restrictions in the SCTE-35 message types that you indicated in **Customize ad triggers**.
      - **Unrestricted** – MediaPackage inserts ad markers when there aren't delivery restrictions in the SCTE-35 message types that you indicated in **Customize ad triggers**.
      - **Both** – MediaPackage inserts ad markers whether or not there are delivery restrictions in the SCTE-35 message types that you indicated in **Customize ad triggers**.

5. (Optional) To include **EXT-X-PROGRAM-DATE-TIME** tags in the output manifest, select **Program date/time interval**, and then type the interval for MediaPackage to insert the tags in the manifest.
The `EXT-X-PROGRAM-DATE-TIME` tag holds the time of the segment. When program date time (PDT) information is available in the source content, MediaPackage uses this same information on the output content. Otherwise, MediaPackage uses Coordinated Universal Time (UTC) for the PDT.

The PDT information helps downstream players to synchronize the stream to the wall clock, enabling functionality like viewer seek in the playback timeline and time display on the player.

6. (Optional) In stream sets with a single video track, to include an additional I-frame only stream along with the other tracks in the manifest, select **Include IFrame only stream**. AWS Elemental MediaPackage inserts `EXT-I-FRAMES-ONLY` tags in the manifest, and then compiles and includes an I-frames only playlist in the stream. This playlist enables player functionality like fast forward and rewind.

**Encryption fields**

Protect your content from unauthorized use through content encryption and digital rights management (DRM). AWS Elemental MediaPackage uses the AWS Secure Package and Encoder Key Exchange (SPEKE) API to facilitate content encryption and decryption by a DRM provider. Using SPEKE, the DRM provider supplies encryption keys to AWS Elemental MediaPackage through the SPEKE API. The DRM provider also supplies licenses to supported media players for decryption. For more information about how SPEKE is used with services and features running in the cloud, see AWS cloud-based architecture in the Secure Packager and Encoder Key Exchange API Specification guide.

**Important**

To encrypt content you must have a DRM provider and use a version of AWS SPEKE. For more information about how to use encryption for MediaPackage, see Content encryption and DRM in AWS Elemental MediaPackage.

1. To serve content without copyright protection, keep **No encryption** selected.
2. To serve content with copyright protection, choose **Encrypt content** and complete the additional fields as follows:
   a. For **Resource ID**, enter an identifier for the content. The service sends this to the key server to identify the current endpoint. How unique you make this depends on how fine-grained you want access controls to be. The service does not allow you to use the same ID for two simultaneous encryption processes. The resource ID is also known as the content ID. The following example shows a resource ID.

   MovieNight20171126093045

   b. For **System IDs**, enter a unique identifier for your streaming protocol and DRM system. Provide up to three IDs. If you provide more than one system ID, enter one per line and choose **Add**. If you do not know your IDs, ask your system provider.

c. For **URL**, enter the URL of the API Gateway proxy that you set up to talk to your key server. The API Gateway proxy must reside in the same AWS Region as MediaPackage. The following example shows a URL.

   https://1wm2dx1f33.execute-api.us-west-2.amazonaws.com/SpekeSample/copyProtection

d. For **Role ARN**, enter the Amazon Resource Name (ARN) of the IAM role that provides you access to send your requests through API Gateway. Get this from your DRM solution provider. The following example shows a role ARN.

   arn:aws:iam::444455556666:role/SpekeAccess
e. **SPEKE version** – (Optional) Select the SPEKE version that you'd like to use for encryption. SPEKE 1.0 is the legacy version that uses CPIX 2.0, and supports single key encryption. SPEKE 2.0 uses CPIX 2.3, and supports multiple key encryption. For more information about using SPEKE with AWS Elemental MediaPackage, see [Content encryption and DRM in AWS Elemental MediaPackage](#).

If you select **SPEKE Version 2.0**, then also choose a **Video encryption preset** and an **Audio encryption preset**. The video and audio presets determine which content keys MediaPackage uses to encrypt the audio and video tracks in your stream.

**Note**
When using SPEKE Version 2.0, MediaPackage disables key rotation.

f. **Certificate ARN** – (Optional) Enter a 2048 RSA certificate ARN to use for content key encryption. Use this option only if your DRM key provider supports content key encryption. If you use this and your key provider doesn't support it, the event fails.

To enter a certificate ARN here, you must have already imported the corresponding certificate into AWS Certificate Manager. Then enter the certificate ARN from ACM here.

For information about content key encryption, see [Preparing and managing certificates for use with content keys](#) (p. 95).

g. (Optional) For **Constant initialization vector** enter a 128-bit, 16-byte hex value represented by a 32-character string, to be used with the key for encrypting content.

h. (Optional) For **Key rotation interval**, enter the frequency, in seconds, of key changes for live workflows, in which content is streamed real time. The service retrieves content keys before the live content begins streaming, and then retrieves them as needed over the lifetime of the workflow. By default, key rotation is set to 60 seconds, which is equivalent to setting it to 60. To disable key rotation, set this interval to 0 (zero).

   The following example setting causes the service to rotate keys every thirty minutes.

   ```
   1800
   ```

   For information about key rotation, see [Understanding key rotation behavior](#) (p. 95).

**Access control fields**

1. To enable this endpoint to serve content to requesting devices, select **Allow origination**. It is uncommon to disallow origination on an endpoint.

   Typically, the only reason that you won't allow an endpoint to serve content is if it's only being used to harvest VOD content from the live stream. For more information, see [Creating live-to-VOD assets with AWS Elemental MediaPackage](#) (p. 84).

2. To serve content to all requesting IP address, choose **Allow all incoming clients**. To limit the IP addresses that this endpoint serves, use these fields:

   a. Select **Restrict by IP address**.

   b. In **Whitelist**, type the IP addresses that this endpoint serves content to.

3. To require that content requests to this endpoint include a valid authorization code, select **Use authorization**. Complete the remaining fields:

   a. In **Secrets role ARN**, enter the ARN for the IAM role that grants MediaPackage access to AWS Secrets Manager.

   b. In **CDN identifier secret**, enter the ARN for the authorization code secret in Secrets Manager.

   For information about how this authorization works, see [CDN authorization in AWS Elemental MediaPackage](#) (p. 89).
Streams to include fields

1. (Optional) For Bitrate order, choose from the following:
   - Original to sort the output streams in the same order that the incoming source uses.
   - Ascending to sort the output streams starting with the lowest bitrate and ending with the highest.
   - Descending to sort the output streams starting with the highest bitrate and ending with the lowest.

2. To make all incoming streams available for playback from this endpoint, select Include all incoming streams.

3. To limit which incoming streams are available for playback from this endpoint, select Filter incoming streams and enter filter criteria:
   - (Optional) For Min video bitrate, type the minimum bitrate threshold that video tracks must meet to be available for playback from this endpoint.
   - (Optional) For Max video bitrate, type the maximum bitrate that video tracks can have to be available for playback from this endpoint.

   The minimum and maximum values take into account only the video bitrates. If the video bitrate is below the minimum specified rate, it is not included in the output, regardless of the sum of the bitrates for other tracks. Likewise, if the video bitrate is below the maximum specified rate, it is included in the output, regardless of the sum of the bitrates for other tracks.

Creating a DASH endpoint

Create an endpoint that formats content for devices that support MPEG-DASH.

To create an MPEG-DASH endpoint (console)

1. Access the channel that the endpoint will be associated with, as described in Viewing channel details (p. 41).

2. On the details page for the channel, choose either Add and edit endpoint or Add endpoints if there are no existing endpoints.

3. Complete the fields as described in the following topics:
   - New endpoint fields (p. 58)
   - Packager settings fields (p. 58)
   - Encryption fields (p. 60)
   - Access control fields (p. 61)
   - Streams to include fields (p. 62)

4. Choose Save endpoints.

   If you enabled Amazon CloudFront distribution creation from the AWS Elemental MediaPackage console and this is your first endpoint on the channel, MediaPackage adds an origin to the distribution. You can view the CloudFront CDN URL and endpoint information in the endpoints section of the channel's details page.

   The endpoint is active and can deliver content as soon as requests are sent to its URL endpoints. AWS Elemental MediaPackage scales resources up and down to allow the right amount of capacity for your traffic.

   When you're creating an endpoint, you will receive an error if you exceed the quotas on the account. An error similar to Too many requests, please try again. Resource limit exceeded means that either you have exceeded the API request quotas, or you have already reached the maximum number of endpoints allowed on this channel. If you think you received this error wrongfully, use the Service
Quotas console to request quota increases. For more information about quotas in AWS Elemental MediaPackage, see Quotas in AWS Elemental MediaPackage (p. 168).

New endpoint fields

When you're creating an endpoint, do not put sensitive identifying information like customer account numbers into free-form fields such as the Name field. This includes when you work with AWS Elemental MediaPackage using the console, REST API, AWS CLI, or AWS SDKs. Any data that you enter into MediaPackage might get picked up for inclusion in diagnostic logs or Amazon CloudWatch Events.

1. For ID, enter a name that describes the endpoint. The ID is the primary identifier for the endpoint, and must be unique for your account in the AWS Region.

2. (Optional) For Description, enter any descriptive text that helps you to identify the endpoint later.

3. For Manifest name, enter a short string that will be appended to the end of the endpoint URL. The manifest name helps to create a unique path to this endpoint.

4. (Optional) To create a window of the live stream that's available for on-demand viewing, select Startover window and enter the size of the window (in seconds). Viewers can start-over or catch-up on content that falls within the window. For more information about implementing start-over and catch-up TV, see Time-shifted viewing reference in AWS Elemental MediaPackage (p. 118).

5. (Optional) To delay when content is available to players, enter the duration (in seconds) for the delay in Time delay. The minimum time is 5 seconds. The maximum time is 86,400 seconds (24 hours).

Use time delay to redefine the live point and make content available at a time that equals "now" minus the delay specified. With a 60-second time delay, content that AWS Elemental MediaPackage receives at 12:20 isn't available until 12:21. Requests for playback at 12:20 will be served with content from 12:19. Likewise, if you're serving content across time zones, you can set a time delay equal to the time zone difference to make content available at, for example, 8:00 local time.

When you use time delay in conjunction with a startover window, the time delay duration must be less than the startover window duration.

Tip
Use a time delay to help reduce buffering during input switching when you're using input redundancy with short output segments. Note that the delay can increase latency in content playback.

Packager settings fields

1. For Type, choose DASH-ISO.

2. (Optional) For Segment duration, type the duration (in seconds) of each segment. If the value that you type here is different from the input segment size, AWS Elemental MediaPackage rounds segments to the nearest multiple of the input segment duration.

Important
If you enable Number with duration in Segment template format, you can't change the segment duration after you've created the endpoint.

3. (Optional) For Manifest window duration, type the total duration (in seconds) of the manifest.

4. (Optional) In Profile, specify a DASH profile, like HbbTV.

Choose from the following:

- None – the output doesn't use a DASH profile
- Hbbtv 1.5 – the output is HbbTV-compliant

5. (Optional) In Manifest layout, choose if you want AWS Elemental MediaPackage to serve a full or compact manifest in response to playback requests.
Creating an endpoint

For more information about the manifest layout options, see Compacted DASH manifests (p. 100).

6. (Optional) For **Min update period**, type the minimum amount of time (in seconds) that the player should wait before requesting manifest updates. A lower value means that manifests are updated more frequently, but a lower value also contributes to request and response network traffic.

7. (Optional) For **Min buffer time**, type the minimum amount of time (in seconds) that a player must keep in the buffer. If network conditions interrupt playback, the player will have additional buffered content before playback fails, allowing for recovery time before the viewer's experience is affected.

8. (Optional) For **Suggested presentation delay**, enter the amount of time (in seconds) that the player should be from the end of the manifest. This sets the content start point back x seconds from the end of the manifest (the point where content is live). For example, with a 35-second presentation delay, requests at 5:30 receive content from 5:29:25. When used with time delay, AWS Elemental MediaPackage adds the suggested presentation delay to the time delay duration.

9. (Optional) In **Segment template format**, choose how AWS Elemental MediaPackage and playback requests refer to each segment.
   - If you choose **Number with timeline**, MediaPackage uses the $Number$ variable to refer to the segment in the media attribute of the SegmentTemplate tag. The value of the variable is the sequential number of the segment. SegmentTimeline is included in each segment template.
   - If you choose **Number with duration**, MediaPackage uses the $Number$ variable and replaces the SegmentTimeline objects with a duration attribute in the segment template.

   **Note**
   This option isn't supported in combination with multi-period DASH.
   - If you choose **Time with timeline**, MediaPackage uses the $Time$ variable to refer to the segment. The value of the variable is the timestamp of when on the manifest timeline the segment starts. SegmentTimeline is included in each segment template.

   For more information about the formatting options of the SegmentTemplate tag, see DASH manifest segment template format (p. 102).

10. For **Period triggers**, choose how AWS Elemental MediaPackage creates media presentation description (MPD) periods in the DASH output manifest. Choose from the following:
   - **None** – MediaPackage doesn't create additional periods. It formats the manifest as a single period and doesn't include SCTE-35 markers in the segments.
   - **Trigger new periods on ads** – MediaPackage creates and inserts in the manifest multiple periods based on SCTE-35 ad markers from the input content. These periods separate portions of the content, such as setting boundaries between the main content and ad content. For more information about how AWS Elemental MediaPackage configures periods in the manifest, see DASH manifest options in AWS Elemental MediaPackage (p. 97).

   **Important**
   Multiple periods are required if you use AWS Elemental MediaTailor for personalized ad insertion in DASH content. For more information about this service, see the AWS Elemental MediaTailor User Guide.

**SCTE-35 Options**

The following fields dictate how MediaPackage processes SCTE-35 messages from the input stream. For more information, see SCTE-35 message options in AWS Elemental MediaPackage (p. 114).
1. (Optional) To add or remove SCTE-35 message types that MediaPackage treats as ads, choose **Customize ad triggers.** If you don’t make a selection here, MediaPackage inserts ad markers in the output manifest based on these message types:
   - Splice insert
   - Provider advertisement
   - Distributor advertisement
   - Provider placement opportunity
   - Distributor placement opportunity

2. (Optional) To change what ad insertion action MediaPackage takes based on delivery restriction flags in the segmentation descriptors of SCTE-35 messages, choose **Ads on delivery restrictions.** These are the available options:
   - **None** – MediaPackage doesn’t insert any ad markers in the output manifest.
   - **Restricted** – MediaPackage inserts ad markers when there are delivery restrictions in the SCTE-35 message types that you indicated in **Customize ad triggers.**
   - **Unrestricted** – MediaPackage inserts ad markers when there aren’t delivery restrictions in the SCTE-35 message types that you indicated in **Customize ad triggers.**
   - **Both** – MediaPackage inserts ad markers whether or not there are delivery restrictions in the SCTE-35 message types that you indicated in **Customize ad triggers.**

If you choose not to insert ad markers, MediaPackage also won’t create periods. The output manifest is contained in a single period.

### Encryption fields

Protect your content from unauthorized use through content encryption and digital rights management (DRM). AWS Elemental MediaPackage uses the **AWS Secure Package and Encoder Key Exchange (SPEKE) API** to facilitate content encryption and decryption by a DRM provider. Using SPEKE, the DRM provider supplies encryption keys to AWS Elemental MediaPackage through the SPEKE API. The DRM provider also supplies licenses to supported media players for decryption. For more information about how SPEKE is used with services and features running in the cloud, see **AWS cloud-based architecture in the Secure Packager and Encoder Key Exchange API Specification guide.**

**Note**

To encrypt content, you must have a DRM solution provider and be set up to use encryption. For information, see the section called “Content encryption and DRM” (p. 93).

1. To serve content without copyright protection, keep **No encryption** selected.
2. To serve content with copyright protection, choose **Encrypt content** and complete the additional fields as follows:
   a. For **Resource ID**, enter an identifier for the content. The service sends this to the key server to identify the current endpoint. How unique you make this depends on how fine-grained you want access controls to be. The service does not allow you to use the same ID for two simultaneous encryption processes. The resource ID is also known as the content ID.

      The following example shows a resource ID.

      MovieNight20171126093045

   b. For **System IDs**, enter unique identifiers for your streaming protocol and DRM system. Provide up to two IDs. If you provide more than one system ID, enter one per line and choose **Add.** For a list of common system IDs, see **DASH-IF System IDs.** If you do not know your IDs, ask your DRM solution provider.
   c. For **URL**, enter the URL of the API Gateway proxy that you set up to talk to your key server. The API Gateway proxy must reside in the same AWS Region as MediaPackage.
The following example shows a URL.

```
https://1wm2dx1f33.execute-api.us-west-2.amazonaws.com/SpekeSample/copyProtection
```

d. **Role ARN**, enter the Amazon Resource Name (ARN) of the IAM role that provides you access to send your requests through API Gateway. Get this from your DRM solution provider.

The following example shows a role ARN.

```
arn:aws:iam::444455556666:role/SpekeAccess
```

e. **SPEKE version** – (Optional) Select the SPEKE version that you'd like to use for encryption. SPEKE 1.0 is the legacy version that uses CPIX 2.0, and supports single key encryption. SPEKE 2.0 uses CPIX 2.3, and supports multiple key encryption. For more information about using SPEKE with AWS Elemental MediaPackage, see [Content encryption and DRM in AWS Elemental MediaPackage](#).

If you select **SPEKE Version 2.0**, then also choose a [Video encryption preset](#) and an [Audio encryption preset](#). The video and audio presets determine which content keys MediaPackage uses to encrypt the audio and video tracks in your stream.

**Note**
When using SPEKE Version 2.0, MediaPackage disables key rotation.

f. **Certificate ARN** – (Optional) Enter a 2048 RSA certificate ARN to use for content key encryption. Use this option only if your DRM key provider supports content key encryption. If you use this and your key provider doesn't support it, the event fails.

To enter a certificate ARN here, you must have already imported the corresponding certificate into AWS Certificate Manager. Then enter the certificate ARN from ACM here.

For information about content key encryption, see [Preparing and managing certificates for use with content keys](#).

g. (Optional) For **Key rotation interval**, enter the frequency, in seconds, of key changes for live workflows, in which content is streamed real time. The service retrieves content keys before the live content begins streaming, and then retrieves them as needed over the lifetime of the workflow. By default, key rotation is set to 60 seconds, which is equivalent to setting it to 60. To disable key rotation, set this interval to 0 (zero).

The following example setting causes the service to rotate keys every thirty minutes.

```
1800
```

For information about key rotation, see [Understanding key rotation behavior](#).

**Access control fields**

1. To enable this endpoint to serve content to requesting devices, select **Allow origination**. It is uncommon to disallow origination on an endpoint.

   Typically, the only reason that you won't allow an endpoint to serve content is if it's only being used to harvest VOD content from the live stream. For more information, see [Creating live-to-VOD assets with AWS Elemental MediaPackage](#).

2. To serve content to all requesting IP address, choose **Allow all incoming clients**. To limit the IP addresses that this endpoint serves, use these fields:

   a. Select **Restrict by IP address**.
   
   b. In **Whitelist**, type the IP addresses that this endpoint serves content to.
3. To require that content requests to this endpoint include a valid authorization code, select **Use authorization**. Complete the remaining fields:
   a. In **Secrets role ARN**, enter the ARN for the IAM role that grants MediaPackage access to AWS Secrets Manager.
   b. In **CDN identifier secret**, enter the ARN for the authorization code secret in Secrets Manager.

   For information about how this authorization works, see **CDN authorization in AWS Elemental MediaPackage** (p. 89).

### Streams to include fields

1. **(Optional) For Bitrate order**, choose from the following:
   - **Original** to sort the output streams in the same order that the incoming source uses.
   - **Ascending** to sort the output streams starting with the lowest bitrate and ending with the highest.
   - **Descending** to sort the output streams starting with the highest bitrate and ending with the lowest.

2. To make all incoming streams available for playback from this endpoint, select **Include all incoming streams**.

3. To limit which incoming streams are available for playback from this endpoint, select **Filter incoming streams** and enter filter criteria:
   - **(Optional) For Min video bitrate**, type the minimum bitrate threshold that video tracks must meet to be available for playback from this endpoint.
   - **(Optional) For Max video bitrate**, type the maximum bitrate that video tracks can have to be available for playback from this endpoint.

   The minimum and maximum values take into account only the video bitrates. If the video bitrate is *below the minimum* specified rate, it is *not* included in the output, regardless of the sum of the bitrates for other tracks. Likewise, if the video bitrate is *below the maximum* specified rate, it *is* included in the output, regardless of the sum of the bitrates for other tracks.

### Viewing all endpoints associated with a channel

View all endpoints that are associated with a specific channel to ensure that the content is available in all necessary stream formats.

You can use the AWS Elemental MediaPackage console, the AWS CLI, or the MediaPackage API to view the endpoints that are associated with a channel. For information about viewing endpoints through the AWS CLI or MediaPackage API, see the [AWS Elemental MediaPackage API Reference](#).

### To view a channel's endpoints (console)

1. Access the channel that the endpoint is associated to, as described in **Viewing channel details** (p. 41).

   AWS Elemental MediaPackage displays all existing endpoints as a table or as individual cards.

2. **(Optional) To adjust your viewing preferences (such as page size and properties that are displayed), choose Preferences.**

### Viewing a single endpoint

View the details about a specific endpoint to obtain its playback URL and to view the packaging settings that it is currently using.
You can use the AWS Elemental MediaPackage console, the AWS CLI, or the MediaPackage API to view the details of an endpoint. For information about viewing endpoint details through the AWS CLI or MediaPackage API, see the AWS Elemental MediaPackage API Reference.

To view a single endpoint's details (console)

1. Access the channel that the endpoint is associated with, as described in Viewing channel details (p. 41).
2. On the channel's details page, choose the endpoint name to view details such as package information and playback preview. For downstream device requests, you must provide the endpoint URL from the Endpoint URL field or the CloudFront CDN URL.

Editing an endpoint

Edit the packaging preferences on an endpoint to optimize the viewing experience. You can't change the packager type after you save an endpoint. To serve content with a different packager, create a different endpoint.

If you edited the channel to enable Amazon CloudFront distribution creation from the AWS Elemental MediaPackage console, you can also edit the endpoint to add an origin to the distribution (if you didn't already add one through alternate means). When you save the edited endpoint, MediaPackage automatically works with CloudFront to create the origin.

You can use the AWS Elemental MediaPackage console, the AWS CLI, or the MediaPackage API to change an endpoint's settings. For information about editing an endpoint through the AWS CLI or MediaPackage API, see the AWS Elemental MediaPackage API Reference.

When you're editing an endpoint, do not put sensitive identifying information like customer account numbers into free-form fields such as the Name field. This includes when you work with AWS Elemental MediaPackage using the console, REST API, AWS CLI, or AWS SDKs. Any data that you enter into MediaPackage might get picked up for inclusion in diagnostic logs or Amazon CloudWatch Events.

To edit an endpoint (console)

1. Access the channel that the endpoint is associated with, as described in Viewing channel details (p. 41).
2. On the channel's details page, do one of the following:
   - Choose Add and edit endpoints and on the Edit page, choose the endpoint to edit.
   - Choose the name of the endpoint to edit, and then choose Edit endpoint.
3. Edit the endpoint options that you want to change.
   - For information about endpoint attributes, see Creating an endpoint (p. 44).
4. Choose Save all.

Deleting an endpoint

Endpoints can serve content until they are deleted. Delete the endpoint if it should no longer respond to playback requests. You must delete all endpoints from a channel before you can delete the channel.

Warning
If you delete an endpoint, the playback URL stops working.

You can use the AWS Elemental MediaPackage console, the AWS CLI, or the MediaPackage API to delete an endpoint. For information about deleting an endpoint through the AWS CLI or MediaPackage API, see the AWS Elemental MediaPackage API Reference.
To delete an endpoint (console)

1. Access the channel that the endpoint is associated with, as described in Viewing channel details (p. 41).
2. On the channel details page, choose the endpoint name.
3. On the endpoint details page, choose Delete endpoint.
4. On the Delete Endpoints page, choose Save all.

Previewing an endpoint

Preview an endpoint's playback to ensure that AWS Elemental MediaPackage is receiving the content stream and can package it. The preview is helpful for avoiding playback failures after the endpoint is published, as well as for troubleshooting later if there are any playback issues.

You can use the AWS Elemental MediaPackage console to preview playback from the endpoint.

To preview an endpoint's playback (console)

1. Access the channel that the endpoint is associated with, as described in Viewing channel details (p. 41).
2. On the channel's details page, locate the endpoint name.
3. To preview playback, do one of the following:
   - Choose Play to play content with the embedded player.
   - Choose QR code to view and scan the QR code for playback on a compatible device.
Delivering VOD content from AWS Elemental MediaPackage

AWS Elemental MediaPackage uses the following resources for video on demand (VOD) content:

- **Packaging groups** hold one or more packaging configurations. The group enables you to apply multiple output configurations to an asset at the same time. You can associate a group to multiple assets so that they all have the same configurations for their outputs.
- **Packaging configurations** tell MediaPackage how to package the output from an asset. In the configuration, you define encryption, bitrate, and packaging settings.
- **Assets** ingest your source content and dynamically apply packaging configurations in response to playback requests.

For supported VOD inputs and codecs, see VOD supported codecs and input types (p. 6).

The following sections describe how to use these resources to manage VOD content in AWS Elemental MediaPackage.

**Topics**

- Working with packaging groups in AWS Elemental MediaPackage (p. 65)
- Working with packaging configurations in AWS Elemental MediaPackage (p. 67)
- Working with assets in AWS Elemental MediaPackage (p. 80)

Working with packaging groups in AWS Elemental MediaPackage

A packaging group holds one or more packaging configurations. When a packaging group is associated with an asset, the packaging configurations define the outputs that are available from the asset. You can associate multiple assets with one packaging group. This enables you to apply the same configurations to multiple assets.

**Topics**

- Creating a packaging group (p. 65)
- Viewing packaging group details (p. 66)
- Editing a packaging group (p. 66)
- Deleting a packaging group (p. 67)
- Adding a packaging configuration to a packaging group (p. 67)

Creating a packaging group

Create a packaging group to hold all of the packaging configurations for an asset. The packaging group, for example, tells AWS Elemental MediaPackage that an asset is available for output to devices that support Apple HLS and DASH-ISO.
When you create a packaging group, you have the option to enable CDN authorization. For more information about CDN authorization, see CDN authorization in AWS Elemental MediaPackage (p. 89).

To create a packaging group, you can use the AWS Elemental MediaPackage console, the AWS CLI, or the MediaPackage API. Information about creating a packaging group with the AWS CLI or MediaPackage API, see Packaging_groups in the AWS Elemental MediaPackage VOD API Reference.

When you're creating a packaging group, don't put sensitive identifying information like customer account numbers into free-form fields, such as the ID field. This applies when you're using the console, REST API, AWS CLI, or AWS SDKs. Any data that you enter into MediaPackage might get picked up for inclusion in diagnostic logs or Amazon CloudWatch Events.

**To create a packaging group (console)**

1. Open the MediaPackage console at https://console.aws.amazon.com/mediapackage/.
2. In the navigation pane, under Video on demand, choose Packaging groups.
3. On the Packaging groups page, choose Create.
4. In the Creating packaging group dialog box, do the following:
   1. For ID, enter a name that describes the packaging group. The ID is the primary identifier for the group, and must be unique for your account in this AWS Region.
   2. Choose Create.

MediaPackage displays the new packaging group's details page.

If you exceed the quotas for your account when you're creating a packaging group, you get an error. If you get an error similar to Too many requests, please try again. Resource limit exceeded, either you have exceeded the API request quotas, or you have already reached the maximum number of packaging groups allowed on your account. If this is your first group, or if you think you mistakenly received this error, use the Service Quotas console to request quota increases. For more information about quotas in MediaPackage, see Quotas in AWS Elemental MediaPackage (p. 168).

**Viewing packaging group details**

You can view all packaging groups that are configured in AWS Elemental MediaPackage or the details of a specific packaging group, including the packaging configurations that are associated with it.

To view packaging group details, you can use the AWS Elemental MediaPackage console, the AWS CLI, or the MediaPackage API. Information about viewing a packaging group with the AWS CLI or MediaPackage API, see Packaging_groups id in the AWS Elemental MediaPackage VOD API Reference.

**To view packaging groups (console)**

1. Open the MediaPackage console at https://console.aws.amazon.com/mediapackage/.
2. In the navigation pane, under Video on demand, choose Packaging groups.
   All groups are displayed on the console.
3. To view more information about a specific packaging group, choose the name of the group.
   AWS Elemental MediaPackage displays summary information, such as the assets associated with this packaging group.

**Editing a packaging group**

Edit the packaging group to configure access control settings.
Note
You can't edit the packager group ID after the packaging group is created. If you want to change the packaging group ID, you must create a new packaging group.

You can use the AWS Elemental MediaPackage console, the AWS CLI, or the MediaPackage API to edit a packaging group's access control settings. For information about editing an packaging group using the AWS CLI or MediaPackage API, see the MediaPackage VOD API reference.

Deleting a packaging group

To stop AWS Elemental MediaPackage from delivering more content from an asset, delete the packaging group. Before you can delete the packaging group, you must delete the group's packaging configurations and any assets that use the group.

• To delete a packaging configuration, see Deleting a packaging configuration (p. 79).
• To delete an asset, see Deleting an asset (p. 83).

To delete a packaging group, you can use the AWS Elemental MediaPackage console, the AWS CLI, or the MediaPackage API. Information about deleting a packaging group with the AWS CLI or MediaPackage API, see Packaging groups id in the AWS Elemental MediaPackage VOD API Reference.

To delete a packaging group (console)

1. Open the MediaPackage console at https://console.aws.amazon.com/mediapackage/.
2. In the navigation pane, under Video on demand, choose Packaging groups.
3. On the Packaging groups page, choose the group using one of the following methods:
   • Choose the group name
   • Select the check box next to the group ID
4. Choose Delete.
5. In the confirmation dialog box, choose Delete to finish deleting the packaging group.

Adding a packaging configuration to a packaging group

To define how AWS Elemental MediaPackage formats outputs from an asset, add a packaging configuration to a packaging group.

To add a packaging configuration to a packaging group, you can use the AWS Elemental MediaPackage console, the AWS CLI, or the MediaPackage API. Information about adding a packaging configuration with the AWS CLI or MediaPackage API, see Packaging configurations in the AWS Elemental MediaPackage VOD API Reference.

For instructions on adding packaging configurations to a packaging group from the MediaPackage console, see Creating a packaging configuration (p. 68).

Working with packaging configurations in AWS Elemental MediaPackage

A packaging configuration defines a single delivery point for an asset. The configuration holds all of the information that's needed for AWS Elemental MediaPackage to integrate with a player or content
Creating a packaging configuration

Create a packaging configuration to define how AWS Elemental Media Package prepares content for delivery from an asset.

To create a packaging configuration, you can use the AWS Elemental Media Package console, the AWS CLI, or the Media Package API. For information about creating a packaging configuration with the AWS CLI or Media Package API, see Packaging_configurations in the AWS Elemental Media Package VOD API Reference.

When you're creating a packaging configuration, don't put sensitive identifying information like customer account numbers into free-form fields, such as the ID field. This applies when you're using the console, REST API, AWS CLI, or AWS SDKs. Any data that you enter into Media Package might get picked up for inclusion in diagnostic logs or Amazon CloudWatch Events.

**Topics**
- Creating an HLS packaging configuration (p. 68)
- Creating a DASH packaging configuration (p. 71)
- Creating a Microsoft Smooth packaging configuration (p. 74)
- Creating a Common Media Application Format (CMAF) packaging configuration (p. 76)

Creating an HLS packaging configuration

Create a packaging configuration that formats content for devices that support Apple HLS.

**To create an Apple HLS packaging configuration (console)**

1. Open the Media Package console at https://console.aws.amazon.com/mediapackage/.
2. In the navigation pane, under Video on demand, choose Packaging groups.
3. On the Packaging groups page, choose the group that will contain the configuration that you're creating.
4. On the details page for the packaging group, in the Packaging configurations section, choose Add or remove configs.
5. On the Add or remove packaging configurations page, in the Packaging configurations section, choose Add and select New config.
6. Complete the fields as described in the following topics:
   - General settings fields (p. 69)
   - Manifest settings fields (p. 69)
   - Stream selection fields (p. 70)
   - Encryption fields (p. 70)
7. Choose Save.

If you exceed the quotas for your account when you're creating a packaging configuration, you get an error. If you get an error similar to Too many requests, please try again. Resource limit exceeded, either you have exceeded the API request quotas, or you have already reached the maximum number of packaging groups allowed on your account. If this is your first group, or if you think you mistakenly received this error, use the Service Quotas console to request quota increases. For more information about quotas in MediaPackage, see Quotas in AWS Elemental MediaPackage (p. 168).

General settings fields
Provide general settings that apply to the entire packaging configuration.

1. For ID, enter a name that describes the configuration. The ID is the primary identifier for the configuration, and must be unique for your account in the AWS Region.
2. For Package type, choose Apple HLS.
3. (Optional) For Segment duration, enter the duration (in seconds) of each segment. If the value that you enter is different from the input segment size, AWS Elemental MediaPackage rounds segments to the nearest multiple of the input segment duration.

Manifest settings fields
Specify the format of the manifest that AWS Elemental MediaPackage delivers from an asset that uses this packaging configuration.

1. (Optional) For Manifest name, enter a short string that will be appended to the endpoint URL. The manifest name creates a unique path to this endpoint. If you don't enter a value, MediaPackage uses the default manifest name, index.
2. (Optional) In stream sets with a single video track, to include an additional I-frame only stream along with the other tracks in the manifest, choose Include IFrame only stream. MediaPackage inserts EXT-I-FRAMES-ONLY tags in the manifest, and then compiles and includes an I-frames only playlist in the stream. This playlist enables player functionality like fast forward and rewind.
3. (Optional) To group all audio tracks into a single HLS rendition group, choose Use audio rendition group. For more information about rendition groups, see Rendition groups reference in AWS Elemental MediaPackage (p. 113).
4. (Optional) Select Repeat EXT-X-KEY if you want the service to repeat the key before every segment of the manifest. By default, the key is written just once, after the header and before the segments. If you select Repeat EXT-X-KEY, the manifest is written as header, key, segment, key, segment, key, and so on, with every segment preceded by the key. Set this according to the needs of the player. Selecting this option might result in an increase in client requests to the DRM server.
5. (Optional) To include EXT-X-PROGRAM-DATE-TIME tags in the output manifest, choose Program date/time interval, and then enter the interval at which MediaPackage should insert the tags in the manifest.
Creating a packaging configuration

The EXT-X-PROGRAM-DATE-TIME tag synchronizes the stream to the wall clock, enabling functionality like viewer seek in the playback timeline and time display on the player.

6. (Optional) In Ad markers, choose how ad markers are included in the packaged content.

Choose from the following:
- **None** – Omit all SCTE-35 ad markers from the output.
- **SCTE-35 enhanced** – Generate ad markers and blackout tags based on the SCTE-35 input messages from the input source.
- **Passthrough** – Copy the SCTE-35 ad markers directly from the input HLS input manifest to the output manifest.

**Stream selection fields**

Limit what incoming bitrates are available for playback and sort the streams in the output of an asset that uses this packaging configuration.

The minimum and maximum values take into account only the video bitrates. If the video bitrate is below the minimum specified rate, it is not included in the output, regardless of the sum of the bitrates for other tracks. Likewise, if the video bitrate is below the maximum specified rate, it is included in the output, regardless of the sum of the bitrates for other tracks.

To set minimum and maximum bitrates and sort the output, choose **Stream selection** and complete the additional fields as follows:

1. (Optional) For **Bitrate order**, choose from the following:
   - **Original** to sort the output streams in the same order that the incoming source uses.
   - **Ascending** to sort the output streams starting with the lowest bitrate and ending with the highest.
   - **Descending** to sort the output streams starting with the highest bitrate and ending with the lowest.

2. (Optional) To ensure tracks are at least a certain bitrate, choose **Min video bitrate** and enter the minimum bitrate threshold that video tracks must meet to be available for playback from this endpoint.

3. (Optional) To ensure tracks are no more than a certain bitrate, choose **Max video bitrate** and enter the maximum bitrate threshold that video tracks must meet to be available for playback from this endpoint.

**Encryption fields**

Protect your content from unauthorized use through content encryption and digital rights management (DRM). AWS Elemental MediaPackage uses the **AWS Secure Package and Encoder Key Exchange (SPEKE) API** to facilitate content encryption and decryption by a DRM provider. Using SPEKE, the DRM provider supplies encryption keys to AWS Elemental MediaPackage through the SPEKE API. The DRM provider also supplies licenses to supported media players for decryption. For more information about how SPEKE is used with services and features running in the cloud, see AWS cloud-based architecture in the **Secure Packager and Encoder Key Exchange API Specification guide**.

**Note**

To encrypt content, you must have a DRM solution provider, and be set up to use encryption. For information, see the section called “Content encryption and DRM” (p. 93).

To serve content with copyright protection, choose **Encryption** and complete the additional fields as follows:

1. For **Encryption method**, choose **Sample-AES** for Apple HLS FairPlay or choose **AES-128** for Apple HLS AES-128.
Creating a packaging configuration

2. For **URL**, enter the URL of the API Gateway proxy that you set up to talk to your key server. The API Gateway proxy must reside in the same AWS Region as MediaPackage.

   The following example shows a URL.

   ```plaintext
   https://1wm2dx1f33.execute-api.us-west-2.amazonaws.com/SpekeSample/copyProtection
   ```

3. For **Role ARN**, enter the Amazon Resource Name (ARN) of the IAM role that provides you access to send your requests through API Gateway. Get this from your DRM solution provider.

   The following example shows a role ARN.

   ```plaintext
   arn:aws:iam::444455556666:role/SpekeAccess
   ```

4. For **System IDs**, enter unique identifiers for your streaming protocol and DRM system. Provide up to three IDs for CMAF, two IDs for DASH, and exactly one for the other streaming protocols. If you provide more than one system ID, enter one per line and choose **Add**. For a list of common system IDs, see **DASH-IF System IDs**. If you do not know your IDs, ask your DRM solution provider.

Creating a DASH packaging configuration

Create a packaging configuration that formats content for devices that support DASH-ISO.

**To create a DASH-ISO packaging configuration (console)**

2. In the navigation pane, under **Video on demand**, choose **Packaging groups**.
3. On the **Packaging groups** page, choose the group that will contain the configuration that you're creating.
4. On the details page for the packaging group, in the **Packaging configurations** section, choose **Add or remove configs**.
5. On the **Add or remove packaging configurations** page, in the **Packaging configurations** section, choose **Add** and **New config**.
6. Complete the fields as described in the following topics:
   - **General settings fields** (p. 71)
   - **Manifest settings fields** (p. 72)
   - **Stream selection fields** (p. 73)
   - **Encryption fields** (p. 73)
7. Choose **Save**.

If you exceed the quotas for your account when you're creating a packaging configuration, you get an error. If you get an error similar to **Too many requests**, please try again. Resource limit exceeded, either you have exceeded the API request quotas, or you have already reached the maximum number of packaging groups allowed on your account. If this is your first group, or if you think you mistakenly received this error, use the Service Quotas console to request quota increases. For more information about quotas in MediaPackage, see **Quotas in AWS Elemental MediaPackage** (p. 168).

**General settings fields**

Provide general settings that apply to the entire packaging configuration.

1. For **ID**, enter a name that describes the configuration. The ID is the primary identifier for the configuration, and must be unique for your account in the Region.
2. For **Package type**, choose **DASH-ISO**.

3. (Optional) For **Segment duration**, enter the duration (in seconds) of each segment. If the value that you enter is different from the input segment size, AWS Elemental MediaPackage rounds segments to the nearest multiple of the input segment duration.

**Manifest settings fields**

Specify the format of the manifest that AWS Elemental MediaPackage delivers from an asset that uses this packaging configuration.

1. (Optional) For **Manifest name**, enter a short string that will be appended to the endpoint URL. The manifest name helps to create a unique path to this endpoint. If you don't enter a value, the default manifest name is **index**.

2. (Optional) For **Min buffer time**, enter the minimum amount of time (in seconds) that a player must keep in the buffer. If network conditions interrupt playback, the player will have additional buffered content before playback fails, allowing for recovery time before the viewer's experience is affected.

3. (Optional) In **Profile**, specify a DASH profile, like HbbTV.

Choose from the following:

- **None** – the output doesn't use a DASH profile
- **Hbbtv 1.5** – the output is HbbTV-compliant

4. (Optional) In **Manifest layout**, choose if you want AWS Elemental MediaPackage to serve a full or compact manifest in response to playback requests.

   - If you choose **Full**, MediaPackage presents the `SegmentTemplate` and `SegmentTimeline` tags for every `Representation` in the manifest.
   - If you choose **Compact**, MediaPackage combines duplicate `SegmentTemplate` tags and presents them at the start of the manifest. This shortens the manifest and makes it easier for some devices to process it.

   For more information about the manifest layout options, see [Compacted DASH manifests](p. 100).

5. (Optional) In **Segment template format**, choose how AWS Elemental MediaPackage and playback requests refer to each segment.

   - If you choose **Number with timeline**, MediaPackage uses the `$Number$` variable to refer to the segment in the `media` attribute of the `SegmentTemplate` tag. The value of the variable is the sequential number of the segment. `SegmentTimeline` is included in each segment template.
   - If you choose **Number with duration**, MediaPackage uses the `$Number$` variable and replaces the `SegmentTimeline` objects with a `duration` attribute in the segment template.

   **Note**

   This option isn't supported in combination with multi-period DASH.

   - If you choose **Time with timeline**, MediaPackage uses the `$Time$` variable to refer to the segment. The value of the variable is the timestamp of when on the manifest timeline the segment starts. `SegmentTimeline` is included in each segment template.

   For more information about the formatting options of the `SegmentTemplate` tag, see [DASH manifest segment template format](p. 102).

6. For **Period triggers**, choose how AWS Elemental MediaPackage creates media presentation description (MPD) periods in the DASH output manifest. Choose from the following:

   - **None** – MediaPackage doesn't create additional periods. It formats the manifest as a single period and doesn't include SCTE-35 markers in the segments.
   - **Trigger new periods on ads** – MediaPackage creates and inserts in the manifest multiple periods based on SCTE-35 ad markers from the input content. These periods separate portions of the content, such as setting boundaries between the main content and ad content. For more
information about how AWS Elemental MediaPackage configures periods in the manifest, see DASH manifest options in AWS Elemental MediaPackage (p. 97).

**Important**
Multiple periods are required if you use AWS Elemental MediaTailor for personalized ad insertion in DASH content. For more information about this service, see the AWS Elemental MediaTailor User Guide.

7. (Optional) For **Include encoder configuration in segments**, AWS Elemental MediaPackage places your encoder’s Sequence Parameter Set (SPS), Picture Parameter Set (PPS), and Video Parameter Set (VPS) metadata in every video segment instead of in the init fragment. This lets you use different SPS/PPS/VPS settings for your assets during content playback.

**Stream selection fields**

Limit which incoming bitrates are available for playback and sort the streams in the output of an asset that uses this packaging configuration.

The minimum and maximum values take into account only the video bitrates. If the video bitrate is below the minimum specified rate, it is not included in the output, regardless of the sum of the bitrates for other tracks. Likewise, if the video bitrate is below the maximum specified rate, it is included in the output, regardless of the sum of the bitrates for other tracks.

To set minimum and maximum bitrates and sort the output, choose **Stream selection** and complete the additional fields as follows:

1. (Optional) For **Bitrate order**, choose from the following:
   - **Original** to sort the output streams in the same order that the incoming source uses.
   - **Ascending** to sort the output streams starting with the lowest bitrate and ending with the highest.
   - **Descending** to sort the output streams starting with the highest bitrate and ending with the lowest.

2. (Optional) To ensure tracks are at least a certain bitrate, choose **Min video bitrate** and enter the minimum bitrate threshold that video tracks must meet to be available for playback from this endpoint.

3. (Optional) To ensure tracks are no more than a certain bitrate, choose **Max video bitrate** and enter the maximum bitrate threshold that video tracks must meet to be available for playback from this endpoint.

**Encryption fields**

Protect your content from unauthorized use through content encryption and digital rights management (DRM). AWS Elemental MediaPackage uses the AWS Secure Package and Encoder Key Exchange (SPEKE) API to facilitate content encryption and decryption by a DRM provider. Using SPEKE, the DRM provider supplies encryption keys to AWS Elemental MediaPackage through the SPEKE API. The DRM provider also supplies licenses to supported media players for decryption. For more information about how SPEKE is used with services and features running in the cloud, see AWS cloud-based architecture in the Secure Packager and Encoder Key Exchange API Specification guide.

**Note**
To encrypt content, you must have a DRM solution provider, and be set up to use encryption. For information, see the section called “Content encryption and DRM” (p. 93).

To serve content with copyright protection, choose **Encryption** and complete the additional fields as follows:

1. For **URL**, enter the URL of the API Gateway proxy that you set up to talk to your key server. The API Gateway proxy must reside in the same AWS Region as MediaPackage.
Creating a packaging configuration

The following example shows a URL.

```
https://1wm2dx1f33.execute-api.us-west-2.amazonaws.com/SpekeSample/copyProtection
```

2. For **Role ARN**, enter the Amazon Resource Name (ARN) of the IAM role that provides you access to send your requests through API Gateway. Get this from your DRM solution provider.

The following example shows a role ARN.

```
arn:aws:iam::444455556666:role/SpekeAccess
```

3. For **System IDs**, enter unique identifiers for your streaming protocol and DRM system. Provide up to three IDs for CMAF, two IDs for DASH, and exactly one for the other streaming protocols. If you provide more than one system ID, enter one per line and choose **Add**. For a list of common system IDs, see [DASH-IF System IDs](#). If you do not know your IDs, ask your DRM solution provider.

**Creating a Microsoft Smooth packaging configuration**

Create a packaging configuration that formats content for devices that support Microsoft Smooth.

**To create a Microsoft Smooth packaging configuration (console)**

2. In the navigation pane, under **Video on demand**, choose **Packaging groups**.
3. On the **Packaging groups** page, choose the group that will contain the configuration that you’re creating.
4. On the details page for the packaging group, in the **Packaging configurations** section, choose **Add or remove configs**.
5. On the **Add or remove packaging configurations** page, in the **Packaging configurations** section, choose **Add** and select **New config**.
6. Complete the fields as described in the following topics:
   - **General settings fields** (p. 74)
   - **Manifest settings fields** (p. 75)
   - **Stream selection fields** (p. 75)
   - **Encryption fields** (p. 75)
7. Choose **Save**.

If you exceed the quotas for your account when you’re creating a packaging configuration, you get an error. If you get an error similar to Too many requests, please try again. Resource limit exceeded, either you have exceeded the API request quota, or you have already reached the maximum number of packaging groups allowed on your account. If this is your first group, or if you think you mistakenly received this error, use the Service Quotas console to request quota increases. For more information about quotas in MediaPackage, see [Quotas in AWS Elemental MediaPackage](#) (p. 168).

**General settings fields**

Provide general settings that apply to the entire packaging configuration.

1. For **ID**, enter a name that describes the configuration. The ID is the primary identifier for the configuration, and must be unique for your account in the Region.
2. For **Package type**, choose **Microsoft Smooth**.
3. (Optional) For **Segment duration**, enter the duration (in seconds) of each segment. If the value that you enter is different from the input segment size, AWS Elemental MediaPackage rounds segments to the nearest multiple of the input segment duration.

**Manifest settings fields**

Specify the format of the manifest that AWS Elemental MediaPackage delivers from an asset that uses this packaging configuration.

1. (Optional) For **Manifest name**, enter a short string that will be appended to the endpoint URL. The manifest name helps to create a unique path to this endpoint. If you don't enter a value, the default manifest name is *index*.

**Stream selection fields**

Limit which incoming bitrates are available for playback and sort the streams in the output of an asset that uses this packaging configuration.

The minimum and maximum values take into account only the video bitrates. If the video bitrate is below the minimum specified rate, it is not included in the output, regardless of the sum of the bitrates for other tracks. Likewise, if the video bitrate is below the maximum specified rate, it is included in the output, regardless of the sum of the bitrates for other tracks.

To set minimum and maximum bitrates and sort the output, choose **Stream selection** and complete the additional fields as follows:

1. (Optional) For **Bitrate order**, choose from the following:
   - **Original** to sort the output streams in the same order that the incoming source uses.
   - **Ascending** to sort the output streams starting with the lowest bitrate and ending with the highest.
   - **Descending** to sort the output streams starting with the highest bitrate and ending with the lowest.

2. (Optional) To ensure tracks are **at least** a certain bitrate, choose **Min video bitrate** and enter the minimum bitrate threshold that video tracks must meet to be available for playback from this endpoint.

3. (Optional) To ensure tracks are **no more than** a certain bitrate, choose **Max video bitrate** and enter the maximum bitrate threshold that video tracks must meet to be available for playback from this endpoint.

**Encryption fields**

Protect your content from unauthorized use through content encryption and digital rights management (DRM). AWS Elemental MediaPackage uses the AWS Secure Package and Encoder Key Exchange (SPEKE) API to facilitate content encryption and decryption by a DRM provider. Using SPEKE, the DRM provider supplies encryption keys to AWS Elemental MediaPackage through the SPEKE API. The DRM provider also supplies licenses to supported media players for decryption. For more information about how SPEKE is used with services and features running in the cloud, see AWS cloud-based architecture in the Secure Packager and Encoder Key Exchange API Specification guide.

**Note**

To encrypt content, you must have a DRM solution provider, and be set up to use encryption. For information, see the section called “Content encryption and DRM” (p. 93).

To serve content with copyright protection, choose **Encryption** and complete the additional fields as follows:

1. For **URL**, enter the URL of the API Gateway proxy that you set up to talk to your key server. The API Gateway proxy must reside in the same AWS Region as MediaPackage.
Creating a packaging configuration

The following example shows a URL.

```plaintext
https://1wm2dxlf33.execute-api.us-west-2.amazonaws.com/SpekeSample/copyProtection
```

2. For **Role ARN**, enter the Amazon Resource Name (ARN) of the IAM role that provides you access to send your requests through API Gateway. Get this from your DRM solution provider.

The following example shows a role ARN.

```plaintext
arn:aws:iam::444455556666:role/SpekeAccess
```

3. For **System IDs**, enter unique identifiers for your streaming protocol and DRM system. Provide up to three IDs for CMAF, two IDs for DASH, and exactly one for the other streaming protocols. If you provide more than one system ID, enter one per line and choose Add. For a list of common system IDs, see DASH-IF System IDs. If you do not know your IDs, ask your DRM solution provider.

### Creating a Common Media Application Format (CMAF) packaging configuration

Create a packaging configuration that formats content for devices that support Apple HLS fragmented MP4 (fMP4).

**To create a CMAF packaging configuration (console)**

1. Open the MediaPackage console at https://console.aws.amazon.com/mediapackage/.
2. In the navigation pane, under **Video on demand**, choose **Packaging groups**.
3. On the **Packaging groups** page, choose the group that will contain the configuration that you're creating.
4. On the details page for the packaging group, in the **Packaging configurations** section, choose **Add or remove configs**.
5. On the **Add or remove packaging configurations** page, in the **Packaging configurations** section, choose **Add** and select **New config**.
6. Complete the fields as described in the following topics:
   - General settings fields (p. 76)
   - Manifest settings fields (p. 77)
   - Stream selection fields (p. 77)
   - Encryption fields (p. 78)
7. Choose **Save**.

If you exceed the quotas for your account when you're creating a packaging configuration, you get an error. If you get an error similar to Too many requests, please try again. Resource limit exceeded, either you have exceeded the API request quotas, or you have already reached the maximum number of packaging groups allowed on your account. If this is your first group, or if you think you mistakenly received this error, use the Service Quotas console to request quota increases. For more information about quotas in MediaPackage, see Quotas in AWS Elemental MediaPackage (p. 168).

**General settings fields**

Provide general settings that apply to the entire packaging configuration.

1. For **ID**, enter a name that describes the configuration. The ID is the primary identifier for the configuration, and must be unique for your account in the Region.
2. For **Package type**, choose **Common Media Application Format (CMAF)**.

3. (Optional) For **Segment duration**, enter the duration (in seconds) of each segment. If the value that you enter is different from the input segment size, AWS Elemental MediaPackage rounds segments to the nearest multiple of the input segment duration.

**Manifest settings fields**

Specify the format of the manifest that AWS Elemental MediaPackage delivers from an asset that uses this packaging configuration.

1. (Optional) For **Manifest name**, enter a short string that will be appended to the endpoint URL. The manifest name creates a unique path to this endpoint. If you don’t enter a value, AWS Elemental MediaPackage uses the default manifest name, *index*.

2. (Optional) In stream sets with a single video track, to include an additional I-frame-only stream along with the other tracks in the manifest, choose **Include IFrame only stream**. MediaPackage inserts EXT-I-FRAMES-ONLY tags in the manifest, and then compiles and includes an I-frames only playlist in the stream. This playlist enables player functionality like fast forward and rewind.

3. (Optional) Select **Repeat EXT-X-KEY** if you want the service to repeat the key before every segment of the manifest. By default, the key is written just once, after the header and before the segments. If you select **Repeat EXT-X-KEY**, the manifest is written as header, key, segment, key, segment, key, and so on, with every segment preceded by the key. Set this according to the needs of the player. Selecting this option might result in an increase in client requests to the DRM server.

4. (Optional) To include EXT-X-PROGRAM-DATE-TIME tags in the output manifest, choose **Program date/time interval**, and then enter the interval at which AWS Elemental MediaPackage should insert the tags in the manifest.

The EXT-X-PROGRAM-DATE-TIME tag synchronizes the stream to the wall clock, enabling functionality like viewer seek in the playback timeline and time display on the player.

5. (Optional) In **Ad markers**, choose how ad markers are included in the packaged content.

Choose from the following:

- **None** – Omit all SCTE-35 ad markers from the output.
- **SCTE-35 enhanced** – Generate ad markers and blackout tags based on the SCTE-35 input messages from the input source.
- **Passthrough** – Copy the SCTE-35 ad markers directly from the input HLS input manifest to the output manifest.

6. (Optional) For **Include encoder configuration in segments**, AWS Elemental MediaPackage places your encoder’s Sequence Parameter Set (SPS), Picture Parameter Set (PPS), and Video Parameter Set (VPS) metadata in every video segment instead of in the init fragment. This lets you use different SPS/PPS/VPS settings for your assets during content playback.

**Stream selection fields**

Limit which incoming bitrates are available for playback and sort the streams in the output of an asset that uses this packaging configuration.

The minimum and maximum values take into account only the video bitrates. If the video bitrate is **below the minimum** specified rate, it is not included in the output, regardless of the sum of the bitrates for other tracks. Likewise, if the video bitrate is **below the maximum** specified rate, it is included in the output, regardless of the sum of the bitrates for other tracks.

To set minimum and maximum bitrates and sort the output, choose **Stream selection** and complete the additional fields as follows:

1. (Optional) For **Bitrate order**, choose from the following:
• **Original** to sort the output streams in the same order that the incoming source uses.

• **Ascending** to sort the output streams starting with the lowest bitrate and ending with the highest.

• **Descending** to sort the output streams starting with the highest bitrate and ending with the lowest.

2. (Optional) To ensure tracks are at least a certain bitrate, choose **Min video bitrate** and enter the minimum bitrate threshold that video tracks must meet to be available for playback from this endpoint.

3. (Optional) To ensure tracks are no more than a certain bitrate, choose **Max video bitrate** and enter the maximum bitrate threshold that video tracks must meet to be available for playback from this endpoint.

### Encryption fields

Protect your content from unauthorized use through content encryption and digital rights management (DRM). AWS Elemental MediaPackage uses the AWS Secure Package and Encoder Key Exchange (SPEKE) API to facilitate content encryption and decryption by a DRM provider. Using SPEKE, the DRM provider supplies encryption keys to AWS Elemental MediaPackage through the SPEKE API. The DRM provider also supplies licenses to supported media players for decryption. For more information about how SPEKE is used with services and features running in the cloud, see AWS cloud-based architecture in the Secure Packager and Encoder Key Exchange API Specification guide.

**Note**

To encrypt content, you must have a DRM solution provider, and be set up to use encryption. For information, see the section called “Content encryption and DRM” (p. 93).

To serve content with copyright protection, choose **Encryption** and complete the additional fields as follows:

1. (Optional) For **Constant initialization vector** enter a 128-bit, 16-byte hex value represented by a 32-character string, to be used with the key for encrypting content.

2. For **URL**, enter the URL of the API Gateway proxy that you set up to talk to your key server. The API Gateway proxy must reside in the same AWS Region as MediaPackage.

   The following example shows a URL.

   ```plaintext
   https://1wm2dx1f33.execute-api.us-west-2.amazonaws.com/SpekeSample/copyProtection
   ```

3. For **Role ARN**, enter the Amazon Resource Name (ARN) of the IAM role that provides you access to send your requests through API Gateway. Get this from your DRM solution provider.

   The following example shows a role ARN.

   ```plaintext
   arn:aws:iam::444455556666:role/SpekeAccess
   ```

4. For **System IDs**, enter unique identifiers for your streaming protocol and DRM system. Provide up to two system IDs. If you provide more than one system ID, enter one per line and choose **Add**. If you do not know your IDs, ask your DRM solution provider.

### Viewing packaging configuration details

To ensure that the content is available in all necessary stream formats, view all packaging configurations that are associated with a specific packaging group or with an asset.

To view packaging configurations, you can use the AWS Elemental MediaPackage console, the AWS CLI, or the MediaPackage API. For information about viewing a packaging configuration with the AWS CLI
or MediaPackage API, see Packaging_configurations id in the AWS Elemental MediaPackage VOD API Reference.

To view packaging configurations in a packaging group (console)
1. Open the MediaPackage console at https://console.aws.amazon.com/mediapackage/.
2. In the navigation pane, under Video on demand, choose Packaging groups.
3. On the Packaging groups page, choose the group that contains the configurations that you want to view.
   The Packaging configurations section displays all of the configurations that are in this group.
4. To view the details of a specific packaging configuration, choose the Id of that configuration.

AWS Elemental MediaPackage displays summary information, such as the assets associated with this packaging configuration.

To view all packaging configurations associated with an asset (console)
1. Open the MediaPackage console at https://console.aws.amazon.com/mediapackage/.
2. In the navigation pane, under Video on demand, choose Assets.
3. On the Assets page, choose the asset that you want to audit.
   The Playback details section displays all of the configurations that are associated with this asset.

Editing a packaging configuration
You can't edit a packaging configuration. If you need to make changes, create a new configuration and delete the original.
- To create a configuration, see Creating a packaging configuration (p. 65).
- To delete a configuration, see Deleting a packaging configuration (p. 67).

Deleting a packaging configuration
To remove a playback endpoint from an asset, delete the packaging configuration.

To delete a packaging configuration, you can use the AWS Elemental MediaPackage console, the AWS CLI, or the MediaPackage API. For information about deleting a packaging configuration with the AWS CLI or MediaPackage API, see Packaging_configurations id in the AWS Elemental MediaPackage VOD API Reference.

To delete a packaging configuration (console)
1. Open the MediaPackage console at https://console.aws.amazon.com/mediapackage/.
2. In the navigation pane, under Video on demand, choose Packaging groups.
3. On the Packaging groups page, choose the group that contains the configuration that you're deleting.
4. On the details page for the packaging group, in the Packaging configurations section, choose the Id of the configuration that you're deleting.
5. On the Packaging configuration details page, choose Delete.
Working with assets in AWS Elemental MediaPackage

An asset holds all of the information that AWS Elemental MediaPackage requires to ingest file-based video content from a source such as Amazon S3. Through the asset, MediaPackage ingests and dynamically packages content in response to playback requests. The configurations associated with the asset determine how it can be packaged for output.

After you ingest an asset, AWS Elemental MediaPackage provides a URL for each playback configuration associated with the asset. This URL is fixed for the lifetime of the asset, regardless of any failures that might happen over time. Downstream devices use the URL to send playback requests.

For supported VOD inputs and codecs, see VOD supported codecs and input types (p. 6).

Topics
- Ingesting an asset (p. 80)
- Viewing asset details (p. 82)
- Editing an asset (p. 83)
- Deleting an asset (p. 83)

Ingesting an asset

To ingest source content, create an asset in AWS Elemental MediaPackage. When MediaPackage ingests content, it creates a unique playback URL for every packaging configuration that’s associated with the asset.

**Important**
To ingest an asset, MediaPackage must have permissions to access the Amazon S3 bucket where the source content is stored. To create a role that gives MediaPackage the right permissions, see Allowing AWS Elemental MediaPackage to access other AWS services (p. 21).

To create an asset, you can use the AWS Elemental MediaPackage console, the AWS CLI, or the MediaPackage API. For information about creating a packaging configuration with the AWS CLI or MediaPackage API, see Assets in the AWS Elemental MediaPackage VOD API Reference.

When you're creating an asset, don't put sensitive identifying information like customer account numbers into free-form fields, such as the ID field. This applies when you're using the console, REST API, AWS CLI, or AWS SDKs. Any data that you enter into MediaPackage might get picked up for inclusion in diagnostic logs or Amazon CloudWatch Events.

To ingest an asset (console)

1. Open the MediaPackage console at https://console.aws.amazon.com/mediapackage/.
2. In the navigation pane, under Video on demand, choose Assets.
3. On the Assets page, choose Ingest asset.
4. On the Ingest asset page, complete the fields as described in the following topics:
   - Asset access fields (p. 81)
   - Asset details fields (p. 81)
   - Packaging settings field (p. 82)
5. Choose Ingest assets.
Ingesting an asset

Ingesting a VOD asset is an asynchronous action. The time it takes before an asset becomes available for playback can vary based on several factors, such as asset duration and asset complexity. You can track when a VOD asset is ready for playback by monitoring the CloudWatch `VodAssetPlayable` events that MediaPackage sends when the asset is ready for playback. For more information, see VOD Playback Events (p. 151).

If you exceed the quotas for your account when you're creating a packaging configuration, you get an error. If you get an error similar to Too many requests, please try again. Resource limit exceeded, either you have exceeded the API request quotas, or you have already reached the maximum number of packaging groups allowed on your account. If this is your first group, or if you think you mistakenly received this error, use the Service Quotas console to request quota increases. For more information about quotas in MediaPackage, see Quotas in AWS Elemental MediaPackage (p. 168).

**Asset access fields**

The following fields describe how AWS Elemental MediaPackage accesses the source content in your Amazon S3 bucket. MediaPackage must have permissions to access the bucket. To create an IAM role with the right permissions, see Allowing AWS Elemental MediaPackage to access other AWS services (p. 21).

**Amazon S3 bucket name**

The Amazon S3 bucket holds the source content that AWS Elemental MediaPackage ingests and packages for playback. Do one of the following:

- To choose from a list of buckets that MediaPackage has detected in your account, choose **Use existing bucket** and choose the bucket.
  
  **Note**
  
  If you don't have permissions to view Amazon S3 buckets, MediaPackage doesn't display any options. Contact your AWS administrator or enter the bucket name manually in the **Specify bucket name** field.

- To use a bucket that MediaPackage hasn't detected, choose **Specify bucket name** and enter the name of the bucket. MediaPackage doesn't have visibility into this bucket, so it can't tell if the bucket is compatible or not.

**IAM role**

The AWS Identity and Access Management (IAM) role provides AWS Elemental MediaPackage permissions to read from the Amazon S3 bucket. Do one of the following:

- To choose from a list of roles that MediaPackage has detected on your account, choose **Use existing role** and choose the role.
  
  **Note**
  
  If you don't have permissions to view IAM roles, AWS Elemental MediaPackage doesn't display any options. Contact your AWS administrator or enter the role ARN manually in the **Specify custom role name** field.

- To use a role that MediaPackage hasn't detected, choose **Specify custom role name** and enter the ARN of the role. Because MediaPackage doesn't have visibility into this role, it can't tell if the role provides the correct permissions or not.

**Asset details fields**

The following fields describe the source content that this asset uses.

If you have multiple sources for this asset, choose **Add asset** and complete the fields. Do this for all source contents.

**Important**

Source content must be in a .smil (MP4) or .m3u8 (HLS/TS) file format.
Filename

The filename identifies the source content.

Enter the path to the file within your Amazon S3 bucket, including the name of the source content.

Example

If your content is called `lion_movie.m3u8` and is in a subdirectory called `thursday_night` in a bucket called `movies`, you would enter the following in the Filename field:

```
thursday_night/lion_movie.m3u8
```

You don't need to enter the bucket name because you chose it in S3 bucket name field.

ID

The ID is the primary identifier for the asset, and must be unique for your account in this Region.

Enter a name that describes the asset.

Resource ID

When you're using SPEKE, the resource ID is the identifier that your key server uses to reference the content. AWS Elemental MediaPackage sends the ID to the key server to identify the current asset. How unique you make the ID depends on the level of access controls you need. The service doesn't allow you to use the same ID for two simultaneous encryption processes. The resource ID is also known as the content ID.

Enter an identifier for the content.

Example

```
MovieNight2017126093045
```

Packaging settings field

The following field determines how AWS Elemental MediaPackage packages outputs from this asset.

Packaging group

The packaging group determines which packaging configurations AWS Elemental MediaPackage uses when it packages content to fulfill playback requests.

Choose the group that holds the configurations that you want to use for this asset.

Viewing asset details

You can view all assets that are configured in AWS Elemental MediaPackage or the details of a specific asset, including the packaging configurations that are associated with it.

To view asset details, you can use the AWS Elemental MediaPackage console, the AWS CLI, or the MediaPackage API. For information about creating a packaging configuration with the AWS CLI or MediaPackage API, see Assets id in the AWS Elemental MediaPackage VOD API Reference.

To view assets (console)

1. Open the MediaPackage console at https://console.aws.amazon.com/mediapackage/.

82
2. In the navigation pane, under **Video on demand**, choose **Assets**.

All assets are displayed on the console.

3. To view more information about a specific asset, choose the name of the asset.

AWS Elemental MediaPackage displays summary information, such as the packaging configurations associated with this packaging asset and their playback URLs.

## Editing an asset

You can't edit an asset. To make changes, ingest the asset again and delete the original.

- To ingest an asset, see Creating a packaging configuration (p. 80).
- To delete an asset, see Deleting a packaging configuration (p. 83).

## Deleting an asset

To remove the packaging group URLs and to stop AWS Elemental MediaPackage from delivering further content, delete an asset.

To delete an asset, you can use the AWS Elemental MediaPackage console, the AWS CLI, or the MediaPackage API. For information about creating a packaging configuration with the AWS CLI or MediaPackage API, see **Assets id** in the *AWS Elemental MediaPackage VOD API Reference*.

### To delete an asset (console)

1. Open the MediaPackage console at https://console.aws.amazon.com/mediapackage/.
2. In the navigation pane, under **Video on demand**, choose **Assets**.
3. On the **Assets** page, choose the asset using one the following methods:
   - Choose the asset name
   - Select the check box next to the asset **ID**
4. Choose **Delete**.
Creating live-to-VOD assets with AWS Elemental MediaPackage

A live-to-VOD (video on demand) asset is a portion of a live stream that’s been extracted and saved for playback later. For example, you might save clips from a game for a highlight reel, or a clip of a broadcast show to use later in advertisements for the show.

To create a live-to-VOD asset in MediaPackage, create a harvest job resource. The harvest job is a request that you create for MediaPackage to extract a portion of a live stream and save the clip as a live-to-VOD asset in an Amazon S3 bucket. The job runs once, then MediaPackage keeps a record of it on your account for 90 days. This record is for reference purposes only. You can’t delete or modify it.

Important
To create live-to-VOD assets, you must allow MediaPackage to access and save to an Amazon S3 bucket. For instructions, see Allowing AWS Elemental MediaPackage to access other AWS services (p. 21).

The following topics provide more information about live-to-VOD assets in MediaPackage.

Topics
- Live-to-VOD requirements (p. 84)
- How live-to-VOD works (p. 85)
- Working with harvest jobs (p. 85)

Live-to-VOD requirements

Keep in mind these requirements when you're creating live-to-VOD assets in AWS Elemental MediaPackage.

Endpoint requirements

The endpoint that you’re harvesting the live-to-VOD asset from must meet these requirements:

- Startover must be enabled and have a **startover window** of 14 days or less. To check or change the size of the window, see the section called “Viewing a single endpoint” (p. 62).
- Your endpoint must serve DASH or HLS content.
- For DASH endpoints - Your DASH endpoint must use either the **Number with timeline** or **Time with timeline** segment template format. For information about creating DASH endpoints, see Creating a DASH endpoint (p. 57).
- MediaPackage VOD doesn't currently support the ingestion of encrypted assets. If you're using your harvested assets in an MediaPackage video on demand workflow and your endpoint is encrypted, create an unencrypted shadow endpoint on the same channel. To do this, deselect **allow origination** so that the new endpoint can't be used for playback. MediaPackage creates the URL for endpoints that don't have origination enabled, but MediaPackage responds with an error to playback requests sent to this endpoint.

Live-to-VOD asset requirements

The live-to-VOD asset must meet these requirements:
How live-to-VOD works

In the processing flow for live-to-VOD (video on demand) content, AWS Elemental MediaPackage extracts a clip of video from a live content stream. MediaPackage saves this clip as a live-to-VOD asset in Amazon S3. You can use the VOD content processing functionality in MediaPackage to deliver the asset to playback devices, or you can use a VOD encoding service that supports HLS or DASH inputs.

Here's an overview of the main steps:

1. You create a channel and endpoint to ingest a live stream and package it for HLS or DASH output. The endpoint must meet the requirements outlined in Live-to-VOD requirements (p. 84).
2. You create a harvest job, which defines the live-to-VOD asset that you're extracting from the live stream. The asset must also meet the requirements outlined in Live-to-VOD Requirements.
3. MediaPackage harvests the timeframe that you indicated in the harvest job. The asset is segment-accurate. This means that if you have a 6-second segment, and the harvest job has a start time of three seconds into the segment, the asset will start three seconds earlier, at the start of the segment.

   After MediaPackage harvests the asset, it saves the asset in the Amazon S3 bucket that you indicated in the harvest job. MediaPackage creates a directory within that bucket and names the parent manifest based on the information that you provided in the Manifest key on the harvest job. For example, if the manifest key is thursdaynight/highlights/index.m3u8, MediaPackage creates a thursdaynight/highlights directory in your Amazon S3 bucket and names the parent manifest index.m3u8.

   MediaPackage creates a CloudWatch event when the harvest job completes or fails. For information about events for harvest jobs, see Harvest job notification events (p. 153).

   MediaPackage keeps a read-only reference of the job on your account for 90 days. After 90 days, MediaPackage deletes the record of the job from your account. At this time, if your workflow requires it, you can reuse the identifier from harvest job.
4. At this point, the live-to-VOD functionality in MediaPackage is complete. The live-to-VOD asset is in your Amazon S3 bucket, and you can do with it what your workflow requires. For example, you can use the VOD functionality in MediaPackage or an encoding service to make the asset available for playback.

Working with harvest jobs

A harvest job represents a request to extract a live-to-VOD (video on demand) asset from an endpoint for a specific timeframe in the past. AWS Elemental MediaPackage uses information from the harvest job to determine the start and end times of the asset, and where to store it after the harvest job is complete.

A harvest job runs only once after it's been created. MediaPackage keeps a record of the job on your account for reference only. You can't modify or delete a record once you've created the harvest job.

Topics

- Creating a harvest job (p. 86)
- Viewing harvest job details (p. 88)
- Editing a harvest job (p. 88)
- Deleting a harvest job (p. 88)
Creating a harvest job

Create a harvest job to extract a live-to-VOD asset from an encrypted or clear (unencrypted) live HLS or DASH stream.

**Important**
To run a harvest job and save the live-to-VOD asset, MediaPackage must have permissions to access and write to the Amazon S3 bucket where the asset will be stored. To create a role that gives MediaPackage the right permissions, see Allowing AWS Elemental MediaPackage to access other AWS services (p. 21).

You can use the MediaPackage console, the AWS CLI, or the MediaPackage API to create a harvest job. For information about creating a job through the AWS CLI or MediaPackage API, see the AWS Elemental MediaPackage API Reference.

When you're creating a harvest job, don't put sensitive identifying information like customer account numbers into free-form fields, such as the ID field. This applies when you're using the console, REST API, AWS CLI, or AWS SDKs. Any data that you enter into MediaPackage might get picked up for inclusion in diagnostic logs or Amazon CloudWatch Events.

**To create a harvest job (console)**
1. Open the MediaPackage console at https://console.aws.amazon.com/mediapackage/.
2. In the navigation pane, under Live, choose Harvest jobs.
3. On the Harvest jobs page, choose Create job.
4. On the Create harvest job page, complete the fields as described in the following topics:
   - Basic details (p. 86)
   - Start and end date and time (p. 87)
   - Destination (p. 87)
5. Choose Create job.

**Basic details**

The basic details of a harvest job define its identifier and the source for the live-to-VOD asset.

1. For ID, enter a name that describes the harvest job. The ID is the primary identifier for the harvest job. You can reuse the ID when the harvest job expires from your account. Supported characters are letters, numbers, underscore (_), and dash (-).
2. For Origin endpoint, select the endpoint that serves the live stream that you're harvesting the live-to-VOD asset from.

Note the following considerations.

- Your harvest job must fall within your MediaPackage endpoint's startover window. The startover window determines the time frame that assets can be harvested from your endpoint. For example, if your endpoint has a startover window of three days, you can harvest your asset anytime within that time frame.

A MediaPackage endpoint can have a startover window between zero and 14 days. To adjust your endpoint's startover window, see Viewing a single endpoint (p. 62).

- Your harvested live-to-VOD asset can have a maximum duration of 24 hours. To set the live-to-VOD asset duration, see Start and end date and time (p. 87) in this chapter.

- Your endpoint must serve either clear (unencrypted) or encrypted DASH or HLS content.
MediaPackage VOD doesn’t currently support the ingestion of encrypted assets. If you’re using your harvested assets in an MediaPackage video on demand workflow and your endpoint is encrypted, create an unencrypted shadow endpoint on the same channel. To do this, deselect allow origination so that the new endpoint can’t be used for playback. MediaPackage creates the URL for endpoints that don’t have origination enabled, but MediaPackage responds with an error to playback requests sent to this endpoint. For more information, see Creating live-to-VOD assets with AWS Elemental MediaPackage (p. 84).

Start and end date and time

The start and end date and time information defines the time range for the harvest job. The maximum duration of the harvest job is 24 hours. Times are based on the program date time (PDT) from the encoder.

**Note**
The live-to-VOD asset timing is accurate up to the segment. This means that if you indicate a start or end time that falls within a segment, MediaPackage includes the entire segment in the asset. If you have a 3-second segment and that start time falls on the third second in the segment, the asset will begin two seconds earlier, at the start of the segment.

1. For **Date and time format**, choose the format that you’re using to indicate the start and end times of the live-to-VOD asset. These are the available options:
   - **Local time**: the date and time is formatted according to the settings of your current browser session. Local time uses a 24-hour clock.
   - **Epoch seconds**: the date and time is formatted in seconds since the epoch.
   - **ISO-8601**: the date and time is formatted according to the ISO-8601 standard.
2. For **Start date and time**, specify when the live-to-VOD asset begins. The asset’s begin time must be at the same time or after the live event started. The start time must also be within the startover window on the endpoint. If the endpoint has a window of 5 hours and the start time is 6 hours ago, the harvest job fails.
3. For **End date and time**, specify when the live-to-VOD asset ends. The length of the asset can’t exceed the startover window on the endpoint. If the endpoint has a window of 5 hours and your start time is 2019/07/29 07:15:00, the end time can’t be after 2019/07/29 12:15:00. The end time must also be in the past.

Destination

The destination information defines how MediaPackage saves the live-to-VOD asset after it has been harvested from the live stream.

1. For **IAM role**, enter the Amazon Resource Name (ARN) for the IAM role that provides MediaPackage access to read and write from your Amazon S3 bucket where the live-to-VOD asset will be stored. This is the role that you created in Allowing AWS Elemental MediaPackage to access other AWS services (p. 21).
2. For **S3 bucket name**, enter the bucket where you want MediaPackage to store the live-to-VOD asset.
3. For **Manifest key**, enter the path within the bucket to the live-to-VOD asset, including the file name for the parent manifest of the asset. If the directory structure doesn’t already exist in the bucket, MediaPackage creates it.

**Important**
The manifest key must be unique. When you use the same manifest key for multiple harvest jobs, the newest playlist for the asset overwrites existing playlists. The only time you should reuse a manifest key is when you are harvesting the same content, such as if there was a problem with a previous harvest of the content.
Viewing harvest job details

View all harvest jobs that you created within the last 90 days. After 90 days, a harvest job expires from your account.

You can use the MediaPackage console, the AWS CLI, or the MediaPackage API to view a harvest job. For information about viewing a job through the AWS CLI or MediaPackage API, see the AWS Elemental MediaPackage API Reference.

To view harvest job details (console)

1. Open the MediaPackage console at https://console.aws.amazon.com/mediapackage/.
2. In the navigation pane, under Live, choose Harvest jobs.
3. On the Harvest jobs page, choose the ID of a harvest job to view its details.

Editing a harvest job

You can't edit a harvest job. To create a harvest job with different settings, see Creating a harvest job (p. 86).

Deleting a harvest job

You can't delete a harvest job.

- To create a harvest job with different settings, see Creating a harvest job (p. 86).
- To delete a VOD asset that MediaPackage created with a harvest job, see Delete an Object and a Bucket in the Amazon Simple Storage Service Getting Started Guide.
AWS Elemental MediaPackage features

The following sections describe the features that are available in AWS Elemental MediaPackage and how they work.

Topics
- CDN authorization in AWS Elemental MediaPackage (p. 89)
- Content encryption and DRM in AWS Elemental MediaPackage (p. 93)
- DASH manifest options in AWS Elemental MediaPackage (p. 96)
- Manifest filtering (p. 106)
- Metadata passthrough (p. 111)
- Rendition groups reference in AWS Elemental MediaPackage (p. 113)
- SCTE-35 message options in AWS Elemental MediaPackage (p. 114)
- Time-shifted viewing reference in AWS Elemental MediaPackage (p. 118)

CDN authorization in AWS Elemental MediaPackage

Content Delivery Network (CDN) authorization helps you to protect your content from unauthorized use. When you configure CDN authentication, MediaPackage only fulfills playback requests that are authorized between MediaPackage and your CDN. This prevents users from bypassing the CDN in order to directly access your content on the origin.

How it works

You configure your CDN, such as Amazon CloudFront, to include a custom HTTP header in content requests to MediaPackage.

Custom HTTP header and example value.

```
X-MediaPackage-CDNIdentifier: 9ceebb7-9607-4552-8764-876e47032660
```

You store the header value as a secret in AWS Secrets Manager. When your CDN sends a playback request, MediaPackage verifies that the secret's value matches the custom HTTP header value. MediaPackage is given permission to read the secret via an AWS Identity and Access Management permissions policy and role.

Secret key and example value.

```
{"MediaPackageCDNIdentifier": "9ceebb7-9607-4552-8764-876e47032660"}
```

If the values match, MediaPackage serves the content along with an HTTP 200 OK status code. If it's not a match, or if the authorization request fails, then MediaPackage doesn't serve the content, and sends an HTTP 403 Unauthorized status code.

The following image shows successful CDN authorization using Amazon CloudFront.
For step-by-step instructions on how to set up CDN authorization, see Setting up CDN authorization (p. 90).

Setting up CDN authorization

Complete the following steps to set up CDN authorization.

Topics
- Step 1: Configure a CDN custom origin HTTP header (p. 90)
- Step 2: Store the value as a secret in AWS Secrets Manager (p. 91)
- Step 3: Create an IAM policy and role for MediaPackage access to Secrets Manager (p. 91)
- Step 4: Enable CDN authorization in MediaPackage (p. 92)

Step 1: Configure a CDN custom origin HTTP header

In your CDN, configure a custom origin HTTP header that contains the header X-MediaPackage-CDNIdentifier and a value. For the value, we recommend that you use the UUID version 4 format, which produces a 36 character string. If you aren't using the UUID version 4 format, the value must be 8-128 characters long.

Important
The value you choose should be a static value. There isn't native integration between your CDN and AWS Secrets Manager, so the value should be static both in your CDN and in AWS Secrets Manager. If you change this value after you configure your CDN and your secret, you have to manually rotate the value. For more information, see Rotating the CDN header value (p. 93).

Example header and value

X-MediaPackage-CDNIdentifier: 9ceebbe7-9607-4552-8764-876e47032660

To create a custom header in Amazon CloudFront

2. Create or edit a distribution.
3. In Origin Settings, complete the fields. You will use this same value for your secret in Secrets Manager.
   - For Header Name, enter X-MediaPackage-CDNIdentifier.
   - For Value, enter a value. We recommend that you use UUID version 4 format, which produces a 36 character string. If you aren’t using the UUID version 4 format, the value must be 8-128 characters long.
4. Complete the rest of the fields and save the distribution.

For more information about custom headers in CloudFront, see Forwarding customer headers to your origin in the Amazon CloudFront Developer Guide.

Step 2: Store the value as a secret in AWS Secrets Manager

Store the same value that you use in your custom origin HTTP header as a secret in AWS Secrets Manager. The secret must use the same AWS account and Region settings as your AWS Elemental MediaPackage resources. AWS Elemental MediaPackage doesn’t support sharing secrets across accounts or Regions. However, you can use the same secret across multiple endpoints in the same Region and on the same account.

To store a secret in Secrets Manager

2. Choose Store a new secret. For Select secret type, choose Other type of secrets.
3. For Specify the key/value pairs to be stored in this secret, choose Secret key/pair.
4. Enter the key and value information.
   - In the box on the left, enter MediaPackageCDNIdentifier.
   - In the box on the right, enter the value that you configured for your custom origin HTTP header. For example, 9ceebbe7-9607-4552-8764-876e47032660.
5. For Select the encryption key, you can keep the default value as DefaultEncryptionKey.
6. Choose Next.
7. For Secret name, we recommend that you prefix it with MediaPackage/ so that you know it’s a secret used for MediaPackage. For example, MediaPackage/cdn_auth_us-west-2.
8. Choose Next.
9. For Configure automatic rotation, keep the default Disable automatic rotation setting.
   - If you need to rotate the authorization code later, see Rotating the CDN header value (p. 93).
10. Choose Next, and then choose Store.

This takes you to the list of your secrets.

11. Select your secret name to view the Secret ARN. The ARN has a value similar to arn:aws:secretsmanager:us-west-2:123456789012:secret:MediaPackage/cdn_auth_test-xxxxxx. You use the Secret ARN when you configure CDN authorization for MediaPackage in Step 4: Enable CDN Authorization in MediaPackage.

Step 3: Create an IAM policy and role for MediaPackage access to Secrets Manager

Create an IAM policy and role to give MediaPackage read access to Secrets Manager. When MediaPackage receives a playback request from the CDN, it verifies that the stored secret value matches the value in the
Setting up CDN authorization

A custom HTTP header. Follow the steps in the section called "Allowing AWS Elemental MediaPackage to access other AWS services" (p. 21) to set up the policy and role.

**Step 4: Enable CDN authorization in MediaPackage**

You can enable CDN authorization for your endpoints or video on demand (VOD) packaging groups via the console, AWS CLI, or MediaPackage API. You use the Amazon Resource Numbers (ARN) for the IAM policy and role that you create in Step 3: Create an IAM policy and role for MediaPackage access to Secrets Manager above.

**Tip**
Use the same secret across multiple endpoints in the same Region and on the same account. Reduce costs by creating a new secret only when necessary for your workflow.

**To enable CDN authorization for live content via the console**

1. Open the MediaPackage console at https://console.aws.amazon.com/mediapackage/.
2. If you don't already have a channel, create one. For help, see Creating a channel (p. 40).
3. Create or edit an endpoint.
4. In **Access control settings**, select **Use authorization**. Complete the fields:
   - In **Secrets role ARN**, enter the Amazon Resource Name (ARN) for the IAM role that you created in Step 3: Create an IAM policy and role for MediaPackage access to Secrets Manager (p. 91).
   - In **CDN identifier secret**, enter the ARN for the secret that holds your authorization code in Secrets Manager.
5. Complete the remaining fields as needed and save the endpoint.

**To enable CDN authorization for VOD content via the console**

1. Open the MediaPackage console at https://console.aws.amazon.com/mediapackage/.
2. If you don't already have a VOD packaging group, create one. For help, see Creating a packaging group (p. 65).
3. Create or edit a packaging group.
4. In **Access control settings**, select **Use authorization**. Complete the fields:
   - In **Secrets role ARN**, enter the Amazon Resource Name (ARN) for the IAM role that you created in Step 3: Create an IAM policy and role for MediaPackage access to Secrets Manager (p. 91).
   - In **CDN identifier secret**, enter the ARN for the secret that holds your authorization code in Secrets Manager.
5. Complete the remaining fields as needed and save the packaging group.

You have now completed the setup for CDN authorization. Requests to this endpoint must contain the same authorization code that you saved in Secrets Manager.

**To enable CDN authorization via the MediaPackage API**

For information about enabling CDN authorization with the MediaPackage API, see the following API references:

- MediaPackage live API reference
- MediaPackage VOD API reference
Rotating the CDN header value

If you change the CDN custom origin HTTP header value, you need to rotate the stored secret value in Secrets Manager. The following procedure describes how to rotate your value in Secrets Manager to make sure that your CDN's HTTP header value and the Secrets Manager stored secret value are in sync.

To rotate the value

1. Update the stored secret value in Secrets Manager as described in Modifying a secret in the AWS Secrets Manager User Guide.

   To ensure continued playback for active streams, MediaPackage authorizes requests that use either the current value in Secrets Manager or one version back.

2. Wait 10 minutes for MediaPackage to recognize that the value has changed in Secrets Manager.

3. In your CDN, update the value in X-MediaPackage-CDNIdentifier to the new authorization code.

4. Wait for your CDN to update fully with the new value before you send any requests through it to MediaPackage.

   To disable the previous secret value, save the new secret value two times. This way, both the current and previous secret versions have the same value.

Content encryption and DRM in AWS Elemental MediaPackage

Protect your content from unauthorized use through content encryption and digital rights management (DRM). AWS Elemental MediaPackage uses the AWS Secure Package and Encoder Key Exchange (SPEKE) API to facilitate content encryption and decryption by a DRM provider. Using SPEKE, the DRM provider supplies encryption keys to AWS Elemental MediaPackage through the SPEKE API. The DRM provider also supplies licenses to supported media players for decryption. For more information about how SPEKE is used with services and features running in the cloud, see AWS cloud-based architecture in the Secure Packager and Encoder Key Exchange API Specification guide.

Limitations and requirements

Keep in mind the following limitations and requirements when implementing content encryption for AWS Elemental MediaPackage:

- You must use the AWS Secure Package and Encoder Key Exchange (SPEKE) API to facilitate integration with a digital rights management (DRM) provider. For information about SPEKE, see What is Secure Packager and Encoder Key Exchange?

- Your DRM provider must support SPEKE. For a list of DRM providers that support SPEKE, see the Get on board with a DRM platform provider topic in the MediaPackage User Guide. Your DRM solution provider can help you get set up to use DRM encryption in AWS Elemental MediaPackage.

- You can use MediaPackage to encrypt live and video on demand (VOD) content. Assets that must be delivered through the MediaPackage VOD service must be harvested from an unencrypted HLS live endpoint. You can harvest live-to-VOD assets from HLS and DASH endpoints that are protected by DRM or encryption, but the MediaPackage VOD service can't ingest these assets since they're not unencrypted (clear) content. For more information about this kind of workflow, see Creating live-to-VOD assets with AWS Elemental MediaPackage (p. 84).
Choosing the right SPEKE version

**SPEKE v1** supports the use of a single encryption key for all audio and video tracks, and uses **CPIX v2.0**. **SPEKE v2** supports the use of multiple, distinct encryption keys for audio and video tracks, and uses **CPIX v2.3**.

If multi-key encryption or **CPIX v2.3** are mandatory requirements for your content delivery, then SPEKE v2 is likely the right choice for you. However, SPEKE v2 support rollout is progressive across endpoint types in MediaPackage. This means that some live options like key rotation aren't available yet. Take these constraints in consideration when crafting your SPEKE integration strategy. Reach out to your AWS account team to learn more about the SPEKE v2 roadmap for MediaPackage.

**Supported protocols and DRM platforms**

The following tables list the different protocols and DRM platforms that SPEKE v1 and SPEKE v2 support:

<table>
<thead>
<tr>
<th>SPEKE v1 – Support matrix for protocol and DRM system</th>
<th>Microsoft PlayReady</th>
<th>Google Widevine</th>
<th>Apple FairPlay</th>
<th>AES-128</th>
</tr>
</thead>
<tbody>
<tr>
<td>Live</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Apple HLS</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>with key rotation</td>
<td>with key rotation</td>
<td></td>
</tr>
<tr>
<td>CMAF Apple HLS</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>with key rotation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DASH</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>with key rotation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Microsoft Smooth</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VOD</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Apple HLS</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>CMAF Apple HLS</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DASH</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Microsoft Smooth</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Deploying SPEKE

Your DRM solution provider can help you get set up to use DRM encryption in MediaPackage. Generally, the provider provides you a SPEKE Gateway to deploy in your AWS account, in the same AWS Region where MediaPackage is running. Along with configuring your origin endpoints with the right encryption settings, you must configure event notifications for the key provider events that MediaPackage is generating as CloudWatch events. For information about configuring encryption settings for your endpoint, see the applicable section for your protocol: HLS encryption fields, MSS encryption fields, CMAF encryption fields, and DASH encryption fields.

If you need to build your own API Gateway to connect MediaPackage to your key service, you can use the SPEKE Reference Server available on GitHub as a starting point.

Preparing and managing certificates for use with content keys

AWS Elemental MediaPackage uses a Content Protection Information Exchange (CPIX) document to communicate with SPEKE about content keys that are used to encrypt your content. For the most secure DRM encryption solution, use encrypted content keys in the CPIX document.

To use encrypted content keys, the following requirements must be met:

- The encrypted content must be live. Video on demand (VOD) and live-to-VOD workflows don’t support encrypted content keys in the CPIX document.
- Your DRM key provider must support encrypted content keys. If you enable this feature for a key provider that doesn’t handle content key encryption, playback fails.
- You must import a suitable certificate into AWS Certificate Manager (ACM) in the same Region that you run MediaPackage. For information about ACM, see the AWS Certificate Manager User Guide.

The following procedures describe how to prepare and manage the certificate.

To prepare a certificate for DRM content key encryption

1. Obtain a 2048 RSA, SHA-512-signed certificate.
2. Open the ACM console at https://console.aws.amazon.com/acm/.
3. Import the certificate into ACM according to the instructions at Importing certificates into AWS certificate manager. Note the resulting certificate ARN because you will need it later.

   For use in DRM encryption, your certificate must have a status of Issued in ACM.

To use a certificate in AWS Elemental MediaPackage
When you use DRM encryption in your endpoint configuration, provide your certificate ARN in the encryption parameters. This enables content key encryption. You can use the same certificate ARN for multiple events. For information, see the encryption settings information in the section called “Working with endpoints” (p. 44).

To renew a certificate

To renew a certificate that you are using in AWS Elemental MediaPackage, reimport it in AWS Certificate Manager. The certificate renews without any disruption of its use in MediaPackage.

To delete a certificate

To delete a certificate from AWS Certificate Manager, it must not be associated with any other service. Delete the certificate ARN from endpoint configurations where you have used it, then delete it from ACM.

Note

If you delete a certificate ARN from an active endpoint, the endpoint keeps running, but stops using content key encryption.

Understanding key rotation behavior

When you enable key rotation on live content from HLS, CMAF, and DASH endpoints, AWS Elemental MediaPackage retrieves content keys before the live content begins. As the content progresses, MediaPackage retrieves new keys at the interval that you set on the endpoint, as described in Encryption fields (p. 47).

If MediaPackage is unable to retrieve the content key, it takes the following actions:

- If MediaPackage successfully retrieved a content key for this endpoint before, it uses the last key that it fetched. This ensures that endpoints that worked previously continue to work.
- If MediaPackage has not successfully retrieved a content key for this endpoint before, MediaPackage responds to the playback request with error 404.

In all cases, when MediaPackage can’t fetch a content key, it generates a CloudWatch event, as described in Key provider notification events (p. 152).

DASH manifest options in AWS Elemental MediaPackage

This section describes the options that AWS Elemental MediaPackage offers for modifying live output DASH manifests.

These options don’t apply to video on demand (VOD) outputs or harvested live-to-VOD assets.

Default DASH manifest

The following is a truncated example of a DASH manifest with no treatments:

```
<MPD>
  <Period>
    <AdaptationSet>
      <Representation>
        <SegmentTemplate>
          96
```

The elements of the DASH manifest are nested within the MPD (media presentation description) object. These are the elements of the manifest:

- **Period**: The entire manifest is nested in one period.
- **AdaptationSet**: An AdaptationSet groups together representations of the same type (video, audio, or captions). There are one or more AdaptationSets in the Period.
- **Representation**: A Representation describes an audio, video, or captions track. There are one or more Representations in each AdaptationSet. Each representation is a track.
- **SegmentTemplate**: A SegmentTemplate defines properties of the representation, such as the timescale and access URLs for media and initialization segments. There is one SegmentTemplate for each Representation.
- **SegmentTimeline**: A SegmentTimeline describes when each segment is available for playback. There is one SegmentTimeline for each SegmentTemplate.
- **S**: An S describes when the segment is available (t value), the duration of the segment (d value), and a count of how many additional consecutive segments have this same duration (r value). There are one or more segments in the SegmentTimeline.

AWS Elemental MediaPackage can modify how some of these elements are presented in the output manifest. You can use the following treatment options on the output live manifest:

- Separate the manifest into multiple periods, to allow ad breaks. See DASH manifest options in AWS Elemental MediaPackage (p. 97).
- Reduce the length of the manifest to make processing and playback more efficient. See Compacted DASH manifests (p. 100).
- Control what segment information is used in the media URL in the SegmentTemplate properties. See DASH manifest segment template format (p. 102).

**Multi-period DASH in AWS Elemental MediaPackage**

Multi-period DASH manifests are available with only live workflows in AWS Elemental MediaPackage.

A period is a chunk of content in the DASH manifest, defined by a start time and duration. By default, the entire manifest is contained in one period but AWS Elemental MediaPackage can partition the DASH manifest into multiple periods to indicate boundaries between ads and the main content. For example, if you're using MediaPackage with a downstream ad service such as AWS Elemental MediaTailor, choose **Trigger new period on ads** on the MPEG-DASH endpoint in MediaPackage. This option tells MediaPackage that the DASH manifest is to be formatted with multiple periods.

- For information about AWS Elemental MediaTailor, see the AWS Elemental MediaTailor User Guide.
- For information about DASH-ISO endpoints in AWS Elemental MediaPackage, see Creating a DASH endpoint (p. 57).
- For more information about how multi-period DASH works in AWS Elemental MediaPackage, see the following How it Works section.
How multi-period DASH works

To use the multi-period DASH feature, the input to AWS Elemental MediaPackage must have SCTE-35 ad marker messages. These messages inform MediaPackage of where to create period boundaries. This is how MediaPackage processes those messages:

1. AWS Elemental MediaPackage detects the SCTE-35 messages from the input source.
2. Using the attributes of the SCTE-35 messages, AWS Elemental MediaPackage calculates where the boundaries are between the end of the main content and the ads. This calculation is \((\text{scte35 ptsAdjustment} + \text{scte35 ptsTime}) / \text{(EventStream timescale)})\).

**Example**

In the following example, the period starts at 44.075 seconds because \((183003 + 3783780) / 90000 = 44.075)\):

```xml
<Period start="PT44.075S" id="21">
    <EventStream timescale="90000" schemeIdUri="urn:scte:scte35:2013:xml">
        <Event>
            <scte35:SpliceInfoSection protocolVersion="0" ptsAdjustment="183003" tier="4095">
                <scte35:SpliceInsert spliceEventId="1000" spliceEventCancelIndicator="false" outOfNetworkIndicator="true" spliceImmediateFlag="false" uniqueProgramId="7" availNum="1" availsExpected="4">
                    <scte35:Program><scte35:SpliceTime ptsTime="3783780"/></scte35:Program>
                </scte35:SpliceInsert>
            </scte35:SpliceInfoSection>
        </Event>
    </EventStream>
</Period>
```

3. AWS Elemental MediaPackage inserts the EventStream, Event, and scte35 tags with additional information into the manifest and surrounds the ad period with a Period tag, as shown in the preceding example. MediaPackage groups all adaptation sets before the first ad period into a period, and any subsequent adaptation sets after the ad are grouped into a period, until the next SCTE-35 marker. Here is a complete manifest example with multiple periods. It uses SpliceInsert SCTE-35 ad markers:

**Example**

```xml
<?xml version="1.0" encoding="utf-8"?>
<MPD>
    <Period start="PT0.000S" id="0" duration="PT44.075S">
        <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=1528413503" initialization="index_video_1_0_init.mp4?m=1528413503" startNumber="6" presentationTimeOffset="0">
                    <SegmentTimeline>
                        <Segment timescale="90000" start="361301" duration="60060" r="15"/>
                    </SegmentTimeline>
                </SegmentTemplate>
            </Representation>
        </AdaptationSet>
        <AdaptationSet mimeType="audio/mp4" segmentAlignment="0" lang="eng">
            <Representation id="2" bandwidth="96964" audioSamplingRate="48000" codecs="mp4a.40.2"/>
        </AdaptationSet>
    </Period>
</MPD>
```
If your input has TimeSignal SCTE-35 ad markers instead of SpliceInsert, the EventStream within the ad period looks like this:

```xml
<EventStream timescale="90000" schemeIdUri="urn:scte:scte35:2013:xml">
  <Event>
    <scte35:SpliceInfoSection protocolVersion="0" ptsAdjustment="183003" tier="4095">
      <scte35:SpliceInsert spliceEventId="1000" spliceEventCancelIndicator="false" outOfNetworkIndicator="true" spliceImmediateFlag="false" uniqueProgramId="7" availNum="1" availItsExpected="4">
        <scte35:Program><scte35:SpliceTime ptsTime="3783780"/></scte35:Program>
      </scte35:SpliceInsert>
    </scte35:SpliceInfoSection>
  </Event>
</EventStream>
```

If your input has TimeSignal SCTE-35 ad markers instead of SpliceInsert, the EventStream within the ad period looks like this:

```xml
<EventStream timescale="90000" schemeIdUri="urn:scte:scte35:2013:xml">
  <Event>
    <scte35:SpliceInfoSection protocolVersion="0" ptsAdjustment="183003" tier="4095">
      <scte35:SpliceInsert spliceEventId="1000" spliceEventCancelIndicator="false" outOfNetworkIndicator="true" spliceImmediateFlag="false" uniqueProgramId="7" availNum="1" availItsExpected="4">
        <scte35:Program><scte35:SpliceTime ptsTime="3783780"/></scte35:Program>
      </scte35:SpliceInsert>
    </scte35:SpliceInfoSection>
  </Event>
</EventStream>
```
AWS Elemental MediaPackage also embeds `scte35:SpliceInsert` messages as metadata in the individual video segments.

If you're using a downstream ad service, that service looks for the SCTE-35 markers in the manifest that AWS Elemental MediaPackage provides and inserts ads based on those markers.

### Compacted DASH manifests

The ability to compact DASH manifests is available with only live workflows in AWS Elemental MediaPackage.

The default DASH manifest from AWS Elemental MediaPackage includes duplicate data about each representation (track). For some players, processing a manifest with all this data is difficult and slow. To reduce some of the burden, MediaPackage can compact the manifest by moving some attributes from the `Representation` object to the `AdaptationSet` object. This way, rather than having the attributes defined for each representation in the manifest, they're defined once at a higher level. The representations then inherit these attributes from the adaptation set.

### Example Default DASH manifest

In the following example, the `SegmentTemplate` object and all of its elements are listed in every `Representation`. Each adaptation set in the manifest has this same layout:

```xml
<AdaptationSet mimeType="video/mp4" segmentAlignment="true" subsegmentAlignment="true" startWithSAP="1" subsegmentStartsWithSAP="1" bitstreamSwitching="true">
  <Representation id="1" width="640" height="360" frameRate="30/1" bandwidth="749952" codecs="avc1.640029">
    <SegmentTemplate timescale="30000" media="index_video_1_0_$Number$.mp4?m=1543947824" initialization="index_video_1_0_init.mp4?m=1543947824" startNumber="1">
      <SegmentTimeline>
        <S t="62000" d="60000" r="9"/>
      </SegmentTimeline>
    </SegmentTemplate>
  </Representation>
  <Representation id="2" width="854" height="480" frameRate="30/1" bandwidth="1000000" codecs="avc1.640029">
    <SegmentTemplate timescale="30000" media="index_video_3_0_$Number$.mp4?m=1543947824" initialization="index_video_3_0_init.mp4?m=1543947824" startNumber="1">
      <SegmentTimeline>
        <S t="62000" d="60000" r="9"/>
      </SegmentTimeline>
    </SegmentTemplate>
  </Representation>
  <Representation id="3" width="1280" height="720" frameRate="30/1" bandwidth="2499968" codecs="avc1.640029">
    <SegmentTemplate timescale="30000" media="index_video_2_0_$Number$.mp4?m=1543947824" initialization="index_video_2_0_init.mp4?m=1543947824" startNumber="1">
      <SegmentTimeline>
        <S t="62000" d="60000" r="9"/>
      </SegmentTimeline>
    </SegmentTemplate>
  </Representation>
</AdaptationSet>
```
Example Compacted DASH manifest

In this example, the SegmentTemplate objects and all of their elements are collapsed into one and moved to the AdaptationSet. The playback device understands that each representation in this adaptation set uses this same template:

```
<AdaptationSet mimeType="video/mp4" segmentAlignment="true" subsegmentAlignment="true"
startWithSAP="1" subsegmentStartsWithSAP="1" bitstreamSwitchings="true">
  <SegmentTemplate timescale="30000" media="index_video_$RepresentationID$_0_$Number$.mp4?m=1543947824"
initialization="index_video_$RepresentationID$_0_init.mp4?m=1543947824" startNumber="1">
    <SegmentTimeline>
      <S t="62000" d="60000" r="9"/>
    </SegmentTimeline>
  </SegmentTemplate>
  <Representation id="1" width="640" height="360" frameRate="30/1" bandwidth="749952"
codecs="avc1.640029"/>
  <Representation id="2" width="854" height="480" frameRate="30/1" bandwidth="1000000"
codecs="avc1.640029"/>
  <Representation id="3" width="1280" height="720" frameRate="30/1" bandwidth="2499968"
codecs="avc1.640029"/>
</AdaptationSet>
```

For information about compacting a DASH manifest, see How AWS Elemental MediaPackage Compacts Manifests (p. 101).

**How AWS Elemental MediaPackage compacts manifests**

To compact the DASH manifest from the AWS Elemental MediaPackage console, choose **Compact** for Manifest layout on the DASH endpoint. To ensure that tracks are available at the right time, AWS Elemental MediaPackage checks the frame rate and audio sampling rate in the source content to determine if the manifest can be compacted.

**Note**
Captions tracks always use the same rate, so AWS Elemental MediaPackage always compacts adaptation sets with captions.

AWS Elemental MediaPackage takes the following actions:

- If the rates are the same across all representations in an adaptation set, AWS Elemental MediaPackage collapses all of the SegmentTemplate objects into one and moves it to the AdaptationSet level. This way, the information in the template isn't repeated throughout the manifest. To allow the playback device to use the same template information across representations, MediaPackage adds a $RepresentationID$ variable to the media and initialization request URLs. The playback device replaces this variable with the ID of the representation that it's currently requesting. MediaPackage also moves the ContentProtection element, when it's present, to the adaptation set as well.

- If the rates are different across representations, AWS Elemental MediaPackage compacts and moves the SegmentTemplate with the most frequent rate to the AdaptationSet. Representations with a different rate keep their segment template. The rate for the representation overrides the one at the adaptation set.
• If there are exactly two frame rates in use in a video adaptation set, AWS Elemental MediaPackage compacts as follows:
  • When 24 and 48 are used, the compacted template uses 48 for the frame rate and 48000 for the timebase.
  • When 25 and 50 are used, the compacted template uses 50 for the frame rate and 50000 for the timebase.
  • When 29.97 and 59.94 are used, the compacted template uses 59.95 for the frame rate and 60000 for the timebase.
  • When 30 and 60 are used, the compacted template uses 60 for the frame rate and 60000 for the timebase.

If there are two video frame rates in use but they aren't in one of the doubled patterns above, then that set can't be compacted.

• If there are no duplicate rates across representations in an adaptation set, then that set can't be compacted.

DASH manifest segment template format

The ability to select the format of the DASH segment template is available with only live workflows in AWS Elemental MediaPackage.

The following sections describe how you can modify the SegmentTemplate object in DASH manifests to better fit your playback device's requirements.

Topics
  • media Attribute in SegmentTemplate (p. 102)
  • duration Attribute in the SegmentTemplate (p. 103)

media Attribute in SegmentTemplate

The media attribute in the SegmentTemplate properties defines the URL where playback devices send segment requests. By default, this URL uses a $Number$ variable to identify the specific segment that is requested. When a playback device requests the segment, it replaces the variable with the number identifier of the segment. For the first segment in the representation, replace this identifier with the value of the startNumber from the SegmentTemplate properties. Each additional segment increments by one.

Some players navigate the segments better when the segments are identified instead by the timestamp for when playback is available. To support this use case, MediaPackage uses the $Time$ variable instead of $Number$ in the URL of the media attribute. When a playback device requests the segment, it replaces the variable with the availability start time of the segment. This start time is identified in the t value of the segment (s) properties in the SegmentTimeline object. For an example, see How It Works (p. 102).

How the $Time$ variable works

Enable the $Time$ variable through the Segment template format setting on the DASH endpoint, as described in Creating a DASH endpoint (p. 57). AWS Elemental MediaPackage takes the following actions:

1. When AWS Elemental MediaPackage generates the DASH manifest, it uses the $Time$ variable in the media value of the SegmentTemplate object, as shown in the following example:
Example

```
<SegmentTemplate timescale="30" media="index_video_1_0_$Time$.mp4?m=1122792372"
  initialization="index_video_1_0_init.mp4?m=1122792372" startNumber="2937928">
</SegmentTemplate>
```

2. When a playback device requests segments, it uses the URL defined in the `media` attribute and replaces the variable with the availability start time of the segment that is requested.

**Important**

The value that replaces the variable must be an exact `t` value of a segment. If the request uses an arbitrary timestamp, AWS Elemental MediaPackage doesn't seek the closest segment.

Example

The following is an example of a segment template from a representation. It uses the `$Time$` variable:

```
<SegmentTemplate timescale="30000" media="155_video_1_2_$Time$.mp4?m=1545421124"
  initialization="155_video_1_2_init.mp4?m=1545421124" startNumber="710">
  <SegmentTimeline>
    <S t="255197799" d="360360" r="8"/>
    <S t="258441039" d="334334"/>
  </SegmentTimeline>
</SegmentTemplate>
```

The request URL for the first segment is `155_video_1_2_255197799.mp4`. With a 360360 duration, the next segment request is `155_video_1_2_255558159.mp4`, and so on through the ninth segment.

The final segment request is `155_video_1_2_258441039.mp4`.

**duration Attribute in the SegmentTemplate**

In a default DASH manifest, `SegmentTemplate` holds a `SegmentTimeline`. The timeline describes all the segments in a representation, including their duration and their start time. With live events, AWS Elemental MediaPackage adds segments to the timeline as it receives them from your encoder. To be aware of newly available segments, the playback device must regularly request an updated manifest from MediaPackage.

If all the segments in a representation have the same duration, you can help to reduce latency and shorten the manifest by enabling AWS Elemental MediaPackage to remove the `SegmentTimeline` objects. In their place, MediaPackage adds a `duration` attribute to the `SegmentTemplate` properties. The playback device calculates when segments are available by using `duration` and `startNumber`. Because the playback device doesn't have to rely on an updated manifest to know about segments, it doesn't have to constantly request updates to maintain playback. For information about how the `duration` attribute works, see the following sections.

**Topics**

- How the `duration` attribute works (p. 103)
- `duration` Attribute with compacted DASH manifests (p. 105)

**How the `duration` attribute works**

Enable the `duration` attribute through the `Segment template format` setting on the DASH endpoint, as described in Creating a DASH endpoint (p. 57). This is what happens with the manifest:
1. When AWS Elemental MediaPackage generates the DASH manifest, it adds the duration attribute to the SegmentTemplate object, as shown in the following example:

**Example**

```xml
<SegmentTemplate timescale="30000" media="index_video_1_0_$Number$.mp4?m=1535562908" initialization="index_video_1_0_init.mp4?m=1535562908" startNumber="175032" duration="90000" presentationTimeOffset="62061"/>
```

A segment timeline and individual segment descriptions are not included in the segment template.

**Important**

Except for the final segment, segments must be no more than 50% deviation from the value of the duration. With a 90000 duration, segments must be between 45000 and 135000 (1.5 to 4.5 seconds with a 30000 timescale).

**Example**

The following is an example of an adaptation set that uses the duration in the segment template:

```xml
<AdaptationSet mimeType="video/mp4" segmentAlignment="true" subsegmentAlignment="true" startWithSAP="1" subsegmentStartsWithSAP="1" bitstreamSwitching="true">
  <Representation id="1" width="852" height="480" frameRate="30/1" bandwidth="1200000" codecs="avc1.4D401F">
    <SegmentTemplate timescale="30000" media="index_video_1_0_$Number$.mp4?m=1535562908" initialization="index_video_1_0_init.mp4?m=1535562908" startNumber="175032" duration="90000" presentationTimeOffset="62061"/>
  </Representation>
  <Representation id="2" width="640" height="360" frameRate="30/1" bandwidth="800000" codecs="avc1.4D401E">
    <SegmentTemplate timescale="30000" media="index_video_3_0_$Number$.mp4?m=1535562908" initialization="index_video_3_0_init.mp4?m=1535562908" startNumber="175032" duration="90000" presentationTimeOffset="62061"/>
  </Representation>
  <Representation id="3" width="320" height="240" frameRate="30/1" bandwidth="499968" codecs="avc1.4D400D">
    <SegmentTemplate timescale="30000" media="index_video_5_0_$Number$.mp4?m=1535562908" initialization="index_video_5_0_init.mp4?m=1535562908" startNumber="175032" duration="90000" presentationTimeOffset="62061"/>
  </Representation>
</AdaptationSet>
```

2. The playback device requests segments using the URL that is defined in the media attribute. In the URL, it replaces the $Number$ variable with the number of the segment, starting with the value of the startNumber in the SegmentTemplate for the first segment.

3. If your playback device needs to determine the most recent segment, it uses this formula:

   \[
   \text{(wall clock time - availabilityStartTime) / (duration / timescale)} + \text{startNumber}
   \]

**Example**

A playback device is calculating the most recent segment with the following values:

- Wall clock from the playback device: 2018-11-16T19:18:30Z
- availabilityStartTime attribute from the MPD object of the manifest: 2018-11-16T19:08:30Z
- duration attribute from the SegmentTemplate object of the manifest: 90000
- timescale attribute from the SegmentTemplate: 30000
- startNumber attribute from the SegmentTemplate: 175032
The calculation it uses is \[(2018-11-16T19:18:30Z - 2018-11-16T19:08:30Z) / (90000/30000)) + 175032\]

This calculation then becomes \((600 \text{ seconds elapsed time}) / (3 \text{ second segment durations}) = 200\) elapsed segments. Adding those segments to the 175032 start segment makes the most recent segment 175232.

**duration Attribute limitations**

To ensure proper playback and help prevent issues with conflicting segment durations, AWS Elemental MediaPackage enforces the following limitations for the `duration` attribute:

- You can enable the feature only when you create the endpoint.
- You can't modify the endpoint to later add the `duration` attribute to your DASH manifests. This includes changing from one segment template format to one that uses duration. For example, you can't create an endpoint that uses the `$Time$` variable with `SegmentTimeline`, and then edit the endpoint to use the `$Number$` variable with `duration`.
- You must keep the **segment duration** value that you set when you create the endpoint.
- You can't edit the endpoint to modify the segment duration.
- You must produce single period DASH manifests from endpoints that use `duration`.
- You can't use multi-period DASH with the `duration` attribute.
- Your ingest stream must use a regular segmentation cadence.
- You can't use variable segment length in the ingest stream. For example, resulting of a SCTE-35-related segmentation.

**duration Attribute with compacted DASH manifests**

Combining compacted manifests with the `duration` attribute will further reduce the size of the manifest, but not by much. Compacted manifests have one `SegmentTemplate` and `SegmentTimeline` per adaptation set. When you use the `duration` attribute, AWS Elemental MediaPackage removes the segment timeline. With both treatments, the manifest has one `SegmentTemplate` per adaptation set, and no `SegmentTimeline`. See the following examples.

For more information about compacted manifests, see *Compacted DASH manifests (p. 100).*

**Important**

If the segments in a representation intentionally have varying sizes of segments, don't use the `duration` attribute. This treatment works only when the segments are a consistent size.

**Example**

**Compacted manifest**

```xml
<AdaptationSet mimeType="video/mp4" segmentAlignment="true" subsegmentAlignment="true" startWithSAP="1" subsegmentStartsWithSAP="1" bitstreamSwitching="true">
  <SegmentTemplate timescale="30000" media="index_video_$RepresentationID$_0_$Number$.mp4?m=1543947824" initialization="index_video_$RepresentationID$_0_init.mp4?m=1543947824" startNumber="1">
    <SegmentTimeline>
      <S t="62000" d="60000" r="9"/>
    </SegmentTimeline>
  </SegmentTemplate>
  <Representation id="1" width="640" height="360" frameRate="30/1" bandwidth="749952" codecs="avc1.640029"/>
</AdaptationSet>
```
Manifest filtering

With manifest filtering, MediaPackage dynamically produces client manifests based on parameters that you specify in a query appended to your playback request. This enables you to do things such as restrict viewer access to premium 4K HEVC content, or target specific device types and audio sample rate ranges, all from a single endpoint. Previously, you would have to configure multiple endpoints to accomplish this behavior. MediaPackage now provides a cost-effective way to dynamically produce different client manifests on the same endpoint.

Working with manifest filters

When you use a manifest filter, the resulting manifest includes only the audio and video streams that match the characteristics that you specify in your query. If no manifest filter is used, then all of the ingested streams are present in the endpoint output stream. The exception to this is if you have set stream filters for the endpoint, such as minimum video bitrate. In that case, the manifest filter is applied after the stream filter, which could skew your output, and is not recommended.

Manifest filtering can be used on all endpoint types supported by MediaPackage:

- Apple HLS
- DASH-ISO
- Microsoft Smooth
- CMAF

To use manifest filtering, append `aws.manifestfilter` query parameters to your playback request to MediaPackage. MediaPackage evaluates the query, and serves a client manifest based on those query parameters. Manifest queries are not case-sensitive and can be up to 1024 characters long. If the query is malformed, or if it there aren't streams that match the query parameters, MediaPackage returns an incomplete or empty manifest. For query syntax, see the following section.

**Note**
If you are using Apple HLS or CMAF endpoints, special conditions apply. For information about these conditions, see [Special conditions for HLS and CMAF manifests](p. 110).

Query syntax
The base query parameter is `aws.manifestfilter`, which is followed by optional parameter name and value pairs. To construct the query, append `?aws.manifestfilter=` to the end of the MediaPackage endpoint URL, followed by parameter names and values. For a list of all of the available parameters, see Manifest filter query parameters (p. 107).

An Apple HLS filter query might look like this:

```
https://example mediapackage-endpoint mediapackage.us-west-2.amazonaws.com/out/v1/example mediapackage/index.m3u8?
aws.manifestfilter=audio_sample_rate:0-44100;video_bitrate:0-2147483647;video_codec:h265;audio_language:fr,en-US,de
```

The query syntax is listed in the following table.

<table>
<thead>
<tr>
<th>Query string component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>A restricted character that marks the beginning of a query.</td>
</tr>
<tr>
<td>aws.manifestfilter=</td>
<td>The base query, which is followed by parameters constructed of name and value pairs. For a list of all of the available parameters, see Manifest filter query parameters (p. 107).</td>
</tr>
<tr>
<td>:</td>
<td>Used to associate the parameter name with a value. For example, <code>parameter_name:value</code>.</td>
</tr>
<tr>
<td>;</td>
<td>Separates parameters in a query that contains multiple parameters. For example, <code>parameter1_name:value;parameter2_name:minValue-maxValue</code>.</td>
</tr>
<tr>
<td>,</td>
<td>Separates a list of values. For example, <code>parameter_name:value1,value2,value3</code>. Comma-separated values in a list imply an OR relationship.</td>
</tr>
<tr>
<td>-</td>
<td>Used to define a parameter's minimum - maximum value range. For example, <code>audio_sample_rate:0-44100</code>. When a numerical value is used in a range, it is included in the range definition. This means that streams must be greater than or equal to the minimum value, and less than or equal to the maximum value. With ranges, the minimum and maximum values are mandatory. The supported range values are 0 - 2147483647.</td>
</tr>
</tbody>
</table>

**Note**

If you use Amazon CloudFront as your CDN, you might need to set additional configurations. For more information, see Configure cache behavior for all endpoints.

### Manifest filter query parameters

MediaPackage supports the following query parameters.

<table>
<thead>
<tr>
<th>Category</th>
<th>Name</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audio</td>
<td>audio_bitrate</td>
<td>• The audio bitrate in bits per second.</td>
<td>stream.mpd?aws.manifestfilter=audio_bitrate:0-2147483647</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>Accepted values:</strong> Two integers aggregated with a dash that define an inclusive range. The supported range values are 0 - 2147483647.</td>
<td></td>
</tr>
<tr>
<td>Category</td>
<td>Name</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Audio</td>
<td>audio_channels</td>
<td>• The number of audio channels.</td>
<td>stream.mpd? aws.manifestfilter=audio_channels:1-8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>Accepted values:</strong> Two integers aggregated with a dash that define an inclusive range. The supported range values are 1 - 32767.</td>
<td></td>
</tr>
<tr>
<td>Audio</td>
<td>audio_codec</td>
<td>• Audio codec type.</td>
<td>stream.mpd? aws.manifestfilter=audio_codec:AACL,AC-3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>Accepted values:</strong> AACL, AACH, AC-3, EC-3. You must include the – for AC-3 and EC-3. The values are not case-sensitive.</td>
<td></td>
</tr>
<tr>
<td>Audio</td>
<td>audio_language</td>
<td>• Audio languages or functional codes derived from encoder passthrough.</td>
<td>stream.mpd? aws.manifestfilter=audio_language:fr,en-US,de</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>Accepted values:</strong> Arbitrary strings, such as two or four character ISO-639-1 language codes. You must use the same language strings that are set for your encoder. The values are not case-sensitive.</td>
<td></td>
</tr>
<tr>
<td>Audio</td>
<td>audio_sample_rate</td>
<td>• Audio sample rate in Hz.</td>
<td>stream.mpd? aws.manifestfilter=audio_sample_rate:0-44100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>Accepted values:</strong> Two integers aggregated with a dash that define an inclusive range. The supported range values are 0 - 2147483647.</td>
<td></td>
</tr>
<tr>
<td>Subtitle</td>
<td>subtitle_language</td>
<td>• The subtitle language(s) or functional codes derived from encoder passthrough.</td>
<td>stream.mpd? aws.manifestfilter=subtitle_language:en-US,hi</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>Accepted values:</strong> Arbitrary strings, such as two or four character ISO-639-1 language codes. You must use the same language strings that are set for your encoder. The values are not case-sensitive.</td>
<td></td>
</tr>
<tr>
<td>Video</td>
<td>trickplay_height</td>
<td>• The height of the trickplay image in pixels.</td>
<td>stream.mpd? aws.manifestfilter=trickplay_height:200-1200</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>Note</strong> This filter is only available for live workflows.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>Accepted values:</strong> Two integers aggregated with a dash that define an inclusive range. The supported range values are 1 - 2147483647.</td>
<td></td>
</tr>
</tbody>
</table>
### Manifest filter query parameters

<table>
<thead>
<tr>
<th>Category</th>
<th>Name</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video</td>
<td>video_bitrate</td>
<td>Video bitrate in bits per second.</td>
<td>stream.mpd? aws.manifestfilter=video_bitrate:0-2147483647</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>If you're using this parameter, we recommend that you use only the video_bitrate filter parameter to set the video bitrate. Don't also set the minimum and maximum video bitrate via the MediaPackage console or AWS CLI. The video_bitrate filter applies to the video bitrate settings created at the endpoint. If you use the parameter and set the bitrate in the console or AWS CLI, your output might be skewed.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Accepted values</strong>: Two integers aggregated with a dash that define an inclusive range. The supported range values are 0 - 2147483647.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>You can't use this parameter with trick play streams.</td>
<td></td>
</tr>
<tr>
<td>Video</td>
<td>video_codec</td>
<td>Video codec type.</td>
<td>stream.mpd? aws.manifestfilter=video_codec:h264</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Accepted values</strong>: H264, H265.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The values are not case-sensitive.</td>
<td></td>
</tr>
<tr>
<td>Video</td>
<td>video_dynamic_range</td>
<td>Video dynamic range.</td>
<td>stream.mpd? aws.manifestfilter=video_dynamic_range:hdr10</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Accepted values</strong>: hdr10, hlg, sdr.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The values are not case-sensitive.</td>
<td></td>
</tr>
<tr>
<td>Video</td>
<td>video_framerate</td>
<td>Video frame rate range in the NTSC format.</td>
<td>stream.mpd? aws.manifestfilter=video_framerate:23.976-30</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Accepted values</strong>: Two floating-point numbers aggregated with a dash that define an inclusive range. Each number can have up to three optional fractional values. For example, 29.97 or 29.764. The supported range values are 1 - 999.999.</td>
<td></td>
</tr>
<tr>
<td>Video</td>
<td>video_height</td>
<td>The height of the video in pixels.</td>
<td>stream.mpd? aws.manifestfilter=video_height:720-1080</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Accepted values</strong>: Two integers aggregated with a dash that define an inclusive range. The supported range values are 1 - 32767.</td>
<td></td>
</tr>
</tbody>
</table>
**Manifest filtering examples**

These are manifest filtering examples.

**Example 1: Target a player that supports AVC and a 44.1k audio sample rate**

The viewer is playing content on a device that can only support AVC and a 44.1k audio sample rate. You set the `video_codec` and `audio_sample_rate` to filter out streams that don't fit these requirements.

```
/aws.manifestfilter=audio_sample_rate:0-44100;video_codec:h264
```

**Example 2: Restrict 4k HEVC content**

Your 4K HEVC stream is 15 Mbps, and all your other streams are less than 9 Mbps. To exclude the 4K stream from the stream set, you set a threshold of 9,000,000 bits per second to filter out the higher bitrate.

```
/aws.manifestfilter=video_bitrate:0-9000000
```

**Example 3: Include video between 23.976 and 30 frames per second**

To only include video within a certain frame rate range, use `video_framerate`. This parameter accepts floating-point numbers with up to three optional decimal values.

```
/aws.manifestfilter=video_framerate:23.976-30
```

**Special conditions for HLS and CMAF manifests**

If you are using HLS or CMAF manifests, these special conditions apply.

- For HLS manifests, we strongly recommend that you use audio rendition groups to avoid removing the video streams that are multiplexed with the audio streams that are filtered out. For more information about rendition groups, see Rendition groups reference in AWS Elemental MediaPackage (p. 113).
- In HLS and CMAF manifests, the audio sample rate is not signaled, so it's not easy to visually check the original or filtered manifests for this setting. To verify the audio sample rate, check the audio sample rate at the encoder level and output level.
- In HLS and CMAF manifests, the `BANDWIDTH` attribute for a variant associates the bandwidth of the audio track with the video track, whether it is multiplexed with the video track, or if it is an audio rendition track referenced by the video track. Therefore you can't visually inspect the original and filtered manifests to confirm the `video_bitrate` filter has worked. To verify the filter, check the video bitrate at the encoder level and output level.
- For HLS and CMAF manifests, request parameters appended to bitrate playlists or segments result in an HTTP 400 error.

**Error conditions**

Common error conditions are listed in the following table.

<table>
<thead>
<tr>
<th>Error condition</th>
<th>Example</th>
<th>HTTP status code</th>
</tr>
</thead>
<tbody>
<tr>
<td>A list parameter is not found and is not part of a constrained list</td>
<td><code>?aws.manifestfilter=audio_language:dahlia</code></td>
<td>200</td>
</tr>
<tr>
<td>Only subtitle streams are present in the stream</td>
<td><code>?aws.manifestfilter=audio_sample_rate:0-1;video_bitrate=0-1</code></td>
<td>200</td>
</tr>
</tbody>
</table>
Metadata passthrough

AWS Elemental MediaPackage automatically passes through ID3 and key-length-value (KLV) metadata from a channel's input to the channel's output stream. You don't need to adjust your endpoint's configuration to enable metadata passthrough.

For more information about how MediaPackage handles metadata, see the following sections.

Topics

- ID3 metadata considerations (p. 111)
- KLV metadata considerations (p. 112)

ID3 metadata considerations

Timed ID3 metadata is a general-purpose mechanism that adds synchronized metadata to streams. The metadata is used for a variety of purposes, ranging from interactive applications to audience measurement.

Supported MediaPackage endpoint types

MediaPackage supports ID3 metadata passthrough for the following endpoint types:
• Live and VOD HLS, DASH, and CMAF endpoints

Metadata carriage

Here is how ID3 is carried as metadata in the following specifications:

• HLS - Metadata is carried in the elementary stream. For more information, see section 2.0 of the Apple Timed Metadata for HTTP Live Streaming reference.

• CMAF and DASH - Metadata is carried in the Event Message box version 1. For more information, see Carriage of ID3 Timed Metadata in CMAF. Event Message boxes include a scheme_id_uri field set to https://aomedia.org/emsg/ID3 and a value field set to 0.

Metadata signaling

DASH manifests include a <InbandEventStream schemeIdUri="https://aomedia.org/emsg/ID3" value="0"/> element in AdaptationSets that include tracks with ID3 metadata.

HLS manifests don't have specific metadata signaling.

MediaLive configuration

You can produce ID3 metadata in AWS Elemental MediaLive MediaPackage output groups either by passing through ID3 metadata, or inserting ID3 metadata using the schedule.

KLV metadata considerations

KLV is a data encoding standard for including synchronized metadata in streams. The binary nature of KLV makes it efficient when the volume of metadata is significant. KLV can be used for various use cases ranging from aerial surveillance to transmitting sensors data in industry use cases, or for real-time athlete and object tracking in live sports use cases.

Supported MediaPackage endpoint types

MediaPackage supports ID3 metadata passthrough for the following endpoint types:

• Live DASH endpoints

Metadata carriage

Metadata is carried in the Event Message box version 1, as described in the MISB ST 1910.1 specification. For synchronous KLV tracks, Event Message boxes include a scheme_id_uri field set to urn:misb:KLV:bin:1910.1 and a value field set to KLVx:01FC. For asynchronous KLV tracks, the value field is set to KLVx:01BD. In both cases, x is the index of the track in the stream.

Metadata signaling

DASH manifests include a <InbandEventStream schemeIdUri="urn:misb:KLV:bin:1910.1" value="KLVx:01FC"/> or <InbandEventStream schemeIdUri="urn:misb:KLV:bin:1910.1" value="KLVx:01BD"/> element in AdaptationSets that include tracks with KLV metadata, depending on the synchronicity nature of the carried track.

MediaLive configuration

You can pass through KLV metadata from your MediaLive channel. For more information, see klv in the AWS Elemental MediaLive User Guide.
Rendition groups reference in AWS Elemental MediaPackage

Rendition groups are used in HLS and CMAF outputs. A rendition group collects all subtitle or audio tracks and makes them available for all video renditions in the stream. When you enable rendition groups, MediaPackage pulls together all audio variants (such as different languages or codecs) and groups them for use with any video rendition. MediaPackage automatically puts subtitles into a rendition group.

Audio and subtitles tracks are required to be in their own rendition groups for CMAF outputs.

The following sections further describe when you can use rendition groups.

Note
DASH and Microsoft Smooth do not use rendition groups. This is because all audio, video, and subtitle or caption tracks are presented to the player, and the player determines which are used during playback.

When to use rendition groups

Rendition groups are used only in HLS and CMAF outputs. Rendition groups are most beneficial when you have multiple languages or multiple audio codecs in your streams. Rendition groups should be used in the following use cases:

Note
If you harvest a live-to-VOD asset from a live HLS stream with rendition groups, the groups are passed through to the asset as well.

- With CMAF outputs, if there are any audio or subtitle tracks

  CMAF requires all audio tracks in one rendition group, and all subtitles in another. Audio or subtitles can't be muxed with video tracks.

- One or more video tracks with multiple audio languages or codecs

  When rendition groups are enabled, MediaPackage pulls all audio renditions together for shared use between the video tracks. In this way, you don't have to duplicate all the audio options across all the video tracks.

- Multiple audio-only tracks and multiple subtitle tracks

  When both the audio tracks and subtitle tracks are in rendition groups, all the audio options can be combined with any subtitle track.

- One audio-only track and multiple subtitle tracks

  MediaPackage automatically pulls subtitle tracks into a rendition group so that the audio track can be used with any subtitle. Because there is only one audio and the subtitles are already grouped, you don't need to tell MediaPackage to use rendition groups in this case.

When not to use rendition groups

Rendition groups can't or shouldn't be used in the following use cases:

- Multiple video tracks in the stream, but only one language or codec is used for the audio. If the same audio is used with multiple video tracks, and rendition groups are also used, then your rendition group will have duplicates of the same audio track (one for each video).

  Keep the audio and video muxed in the stream, and do not use a rendition group.
DASH or Microsoft Smooth Streaming outputs. These protocols do not support rendition groups. Instead, the output stream includes all tracks, and the player determines which to play based on rules from the player side or from the manifest (such as language or bitrate selection).

To limit the tracks available to a player, use the stream selection options from the MediaPackage console or the REST API.

SCTE-35 message options in AWS Elemental MediaPackage

This section describes the options that AWS Elemental MediaPackage offers for configuring how SCTE-35 messages are handled in live DASH, HLS, and CMAF outputs. For live-to-VOD assets, MediaPackage passes the SCTE-35 messages from the live stream through to the harvested asset. These options don't apply to Microsoft Smooth Streaming or video on demand (VOD) outputs.

SCTE-35 messages accompany video in your source content. These messages signal where MediaPackage should insert ad markers when it packages the content for output. By default, MediaPackage inserts markers for the following message types in the source content:

- splice_insert
- time_signal with the following segmentation types:
  - Provider advertisement
  - Distributor advertisement
  - Provider placement opportunity
  - Distributor placement opportunity

The `time_signal` must also include delivery restriction flags in the `segmentation_descriptor`.

When these commands are present, MediaPackage inserts corresponding ad markers in the output manifests:

- For HLS and CMAF outputs, MediaPackage inserts `EXT-X-CUE-OUT` and `EXT-X-CUE-IN` tags.
- For DASH outputs, MediaPackage inserts EventStream tags to create multiple periods, when you have multi-period manifests enabled.

The following sections describe how you can modify MediaPackage SCTE-35 message handling behavior.

SCTE-35 settings in MediaPackage

You can modify how MediaPackage interacts with SCTE-35 messages from your source content. Configure the following settings on your endpoints. For more information, see the following:

- For the MediaPackage console, see the section called “Creating an endpoint” (p. 44).
- For the MediaPackage REST API, see `Origin_endpoints` in the AWS Elemental MediaPackage API Reference.

**Important**

To modify how MediaPackage handles SCTE-35 messages, you should be familiar with the SCTE-35 standard. You can download a PDF of the most recent standards here: Download SCTE ISBE standards. You should also be familiar with how SCTE-35 is implemented in your source content.
Ad markers

This setting is available on HLS and CMAF endpoints.

Ad markers allows you to specify what MediaPackage does when it detects SCTE-35 messages. These are the options:

- **None** – MediaPackage ignores the SCTE-35 messages and doesn't include ad markers in the output manifest.
- **SCTE-35 enhanced** – MediaPackage includes ad markers and blackout tags in the output manifest for SCTE-35 messages that meet the requirements in Customize ad triggers and Ads on delivery restrictions.
- **Passthrough** – MediaPackage copies all SCTE-35 messages from the source content and inserts them in the output manifest.

Customize ad triggers

This setting is available on HLS, CMAF, and DASH endpoints.

Customize ad triggers identifies which SCTE-35 message types MediaPackage treats as ads in the output manifest.

If you don't change this setting, MediaPackage treats these message types as ads:

- Splice insert
- Provider advertisement
- Distributor advertisement
- Provider placement opportunity
- Distributor placement opportunity

Ads on delivery restrictions

This setting is available on HLS, CMAF, and DASH endpoints.

Ads on delivery restrictions sets conditions for what SCTE-35 messages become ads, based on the delivery restriction flags in the segmentation descriptor of the messages. MediaPackage inserts an ad marker that corresponds to the positioning of the messages of the right type that meet the delivery restriction conditions.

If you don't change this setting, MediaPackage converts messages that are classified as restricted (they have delivery restriction flags) to ad markers in the output manifest.

Note

Splice insert SCTE-35 messages don't have segmentation_descriptor. If you choose splice insert in Customize ad triggers, all splice inserts become ad markers in the output manifest.

How it works

The Ad markers, Customize ad triggers, and Ads on delivery restrictions settings work together to determine what MediaPackage does with SCTE-35 messages from the source content.

When there are SCTE-35 messages in the source content, MediaPackage takes the following action based on the value that you selected in Ad markers:

- For None, MediaPackage does nothing with the SCTE-35 messages. No ad markers are inserted in the output manifest.
- For Passthrough, MediaPackage copies all SCTE-35 messages from the source content and inserts them in the output manifest.
• For **SCTE-35 enhanced**, MediaPackage checks for messages that meet the requirements that you set. In the output manifest, MediaPackage inserts ad markers that correspond to the applicable messages. To check for your requirements, MediaPackage does the following:

1. Checks if any SCTE-35 messages match the message types that you indicated in **Customize ad triggers**
2. For messages of the right types, checks if the delivery restriction flags in **segmentation_descriptor** meet the conditions that you set in **Ads on delivery restrictions**
3. For messages of the right type that meet the delivery restriction conditions, inserts ad markers in the output manifest, as described earlier in this chapter
4. For **Daterange**, MediaPackage inserts EXT-X-DATERANGE tags to signal ads and program transition events in HLS and CMAF output manifests.

**EXT-x-DATERANGE ad markers**

Daterange ad markers are used to signal ads and program transitions in live HLS and CMAF manifests. When you enable daterange ad markers on your endpoint, MediaPackage inserts EXT-X-DATERANGE tags into the manifest where there are SCTE-35 time_signal or splice_insert tags present. EXT-X-DATERANGE is used in concert with EXT-X-PROGRAM-DATE-TIME tags.


**Enabling daterange via the console**

To enable daterange ad markers for your endpoint, do the following in the MediaPackage console:

Go to **Add/edit endpoints > Packager settings > Additional configuration > Ad marker** and select **Daterange**. If you select this option, you must also set a **Program date/time interval (sec)** value that is greater than 0. The program date/time interval is set in the same **Additional configuration** pane as the ad marker settings.

**Enabling daterange via the AWS CLI**

To enable daterange ad markers for your endpoint, run the following command in the AWS CLI:

```bash
aws --endpoint=https://mediapackage.region.amazonaws.com mediapackage --region region create-origin-endpoint --channel-id test_channel --id hlsmuxed --hls-package "{"ProgramDateTimeIntervalSeconds":60,"AdMarkers":"DATERANGE\"}"
```

**Important**

You must set a **ProgramDateTimeIntervalSeconds** value that is greater than 0.

**Enabling daterange via the MediaPackage live API or AWS SDK**

To learn how to enable daterange ad markers for HLS endpoints via the live API or AWS SDK, see the following:

- MediaPackage Live API reference
- AWS SDK
Example HLS manifest showing SCTE-35 EXT-x-DATERANGE signaling

This example HLS manifest generated by MediaPackage uses EXT-X-DATERANGE and EXT-X-PROGRAM-DATE-TIME tags to signal events in the live stream.

Note

The DURATION, PLANNED-DURATION, and END-DATE attributes of the EXT-X-DATERANGE tag are optional. If these attributes aren't present in the SCTE-35 input, or aren't set when you create your endpoint via the MediaPackage REST API, then they are omitted from the generated manifests.

```xml
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-TARGETDURATION:8
#EXT-X-MEDIA-SEQUENCE:11
#EXT-X-DATERANGE:ID="2415919105",START-DATE="2020-05-03T00:01:00.018Z",PLANNED-DURATION=29.988
#EXT-X-DATERANGE:ID="2047483649",START-DATE="2020-05-03T00:01:00.018Z",END-DATE="2020-05-03T00:01:30.006Z",DURATION=29.988
#EXT-X-PROGRAM-DATE-TIME:2020-05-03T00:01:08.040Z
#EXTINF:7.560,
../../../../index_1_11.ts?m=1588607409
#EXTINF:7.560,
../../../../index_1_12.ts?m=1588607409
#EXTINF:6.846,
../../../../index_1_13.ts?m=1588607409
#EXT-X-DATERANGE:ID="2415919105",START-DATE="2020-05-03T00:01:01.006Z",PLANNED-DURATION=29.988
#EXT-X-DATERANGE:ID="2147483649",START-DATE="2020-05-03T00:00:30.030Z",PLANNED-DURATION=90.006
#EXT-X-DATERANGE:ID="2147483650",START-DATE="2020-05-03T00:00:30.030Z",PLANNED-DURATION=90.006
#EXT-X-PROGRAM-DATE-TIME:2020-05-03T00:01:08.040Z
#EXTINF:7.560,
../../../../index_1_14.ts?m=1588607409
#EXTINF:7.560,
../../../../index_1_15.ts?m=1588607409
#EXTINF:6.636,
../../../../index_1_16.ts?m=1588607409
#EXTINF:6.636,
../../../../index_1_17.ts?m=1588607409
#EXTINF:0.924,
../../../../index_1_18.ts?m=1588607409
#EXT-X-DATERANGE:ID="2147483649",START-DATE="2020-05-03T00:00:30.030Z",END-DATE="2020-05-03T00:00:30.030Z",DURATION=90.006
#EXT-X-DATERANGE:ID="2147483650",START-DATE="2020-05-03T00:00:30.030Z",END-DATE="2020-05-03T00:00:30.030Z",DURATION=90.006
#EXT-X-PROGRAM-DATE-TIME:2020-05-03T00:02:08.520Z
#EXTINF:7.560,
../../../../index_1_19.ts?m=1588607409
#EXTINF:7.560,
../../../../index_1_20.ts?m=1588607409
#EXT-X-DATERANGE:ID="2147483649",START-DATE="2020-05-03T00:00:30.030Z",PLANNED-DURATION=90.006
#EXT-X-DATERANGE:ID="2147483650",START-DATE="2020-05-03T00:00:30.030Z",PLANNED-DURATION=90.006
#EXT-X-PROGRAM-DATE-TIME:2020-05-03T00:02:08.520Z
#EXTINF:7.560,
../../../../index_1_21.ts?m=1588607409
#EXT-X-ENDLIST
```
Time-shifted viewing reference in AWS Elemental MediaPackage

Time-shifted viewing is available with live workflows in AWS Elemental MediaPackage.

*Time-shifted viewing* means that viewers can start watching a live stream at a time earlier than "now," allowing them to join from the beginning a show that's already in progress or to watch a show that's already completed. MediaPackage supports time-shifted viewing for content that's up to 336 hours (14 days) old. You can enable time-shifted viewing for some or all of this content by defining the **startover window** on the endpoint. Content that falls within that window is available for playback when playback requests include valid start and end parameters. Requests for content outside the window configured on the endpoint result in an HTTP error 404.

Alternatively, you can harvest a clip of a live stream and make it available as a video on demand (VOD) asset. For information about harvesting VOD assets, see Creating live-to-VOD assets with AWS Elemental MediaPackage (p. 84).

In the following steps, "now" is the current time according to the program date time (PDT), when it's present in the source content from the encoder. If the source content doesn't include PDT information, "now" refers to the MediaPackage ingest time of the most recent segment.

**To enable time-shifted viewing**

1. Enable time-shifted viewing by typing a value for **Startover window** on the AWS Elemental MediaPackage endpoint object. You can do this through either the MediaPackage console or the REST API.

   When requests with start and end parameters that are within the startover window are sent to this endpoint, AWS Elemental MediaPackage generates a manifest for the requested timeframe. If the start or end parameters are outside of the startover window, the playback request fails. If no start and end parameters are used, the service generates a standard manifest.

   **Note**
   
   You might notice that the manifest lags behind real time when you initially create a startover window on an endpoint. This is because AWS Elemental MediaPackage starts filling the manifest from the start of the window, and works up to "now." So if you have a 24-hour startover window, MediaPackage fills the manifest starting 24 hours ago and working up to "now."

2. Ensure that content requests contain start and end parameters as needed. AWS Elemental MediaPackage accepts requests for up to 24 hours of content.

   For packager-specific rules about how you can notate the parameters, see Rules for start and end parameters (p. 119).

The start and end parameters determine the time boundaries of the manifest. These are the expected behaviors based on request start and end parameters:

- If both start and end parameters are used in the URL, the resulting manifest has a fixed start and end time that correspond to the specified start and end parameters.

  If the end time is in the future, the tags in the manifest are consistent with a live manifest. Otherwise, if the end time is in the past, the tags in the manifest are consistent with a video on demand (VOD) manifest. For information about the manifest differences, see Live and VOD manifest reference (p. 12).

- If a start parameter is specified but not an end, the resulting manifest has a fixed start time that corresponds to the specified start parameter, and the end of the manifest grows as the live content progresses.
Note
For HLS output, many playback devices start playback at the current time ("now"). To view the content from the actual start time of the playback window, viewers can seek back on the playback progress bar.

- If no parameters are specified, a standard manifest is generated starting "now" with no end time.
- If an end parameter is specified but no start, the manifest is generated in the same way as when no parameters are specified. The manifest starts "now" and has no end time.

Rules for start and end parameters

Start and end parameters denote the beginning and end of a time-shifted manifest. The playback device can append parameters to the end of a manifest request or include the parameters within the request.

In all cases, the date and time must be notated in one of the following formats:

- ISO 8601 dates, such as 2017-08-18T21:18:54+00:00
- POSIX (or Epoch) time, such as 1503091134

The following topics describe the location rules by packager type.

DASH parameter rules

Start and end parameters in the URL request for DASH content can use standard parameter notation, or can be included as path elements in the URL.

- Query parameter notation – start and end parameters are included at the end of the request URL

  Example

  https://cf98fa7b2ee4450e.mediapackage.us-east-1.amazonaws.com/out/v1/997cbb27697d4863bb65488133bfff26f/sports.mpd?start=1513717228&end=1513720828

- Path elements – start and end parameters are included in the path of the request URL

  Example

  https://cf98fa7b2ee4450e.mediapackage.us-east-1.amazonaws.com/out/v1/997cbb27697d4863bb65488133bfff26f/start/2017-12-19T13:00:28-08:00/end/2017-12-19T14:00:28-08:00/sports.mpd

HLS and CMAF parameter rules

Start and end parameters in the URL request for HLS content can use standard parameter notation, or can be included as path elements in the URL. The rules for HLS and CMAF are the same, except that when you're inserting path elements in the CMAF endpoint, the elements have to be after the manifest ID in the URL.

- Query parameter notation – start and end parameters are included at the end of the request URL
Rules for start and end parameters

Example HLS

https://cf98fa7b2ee4450e.mediapackage.us-east-1.amazonaws.com/out/v1/064134724fd74667ba294657a674ae72/comedy.m3u8?start=2017-12-19T13:00:28-08:00&end=2017-12-19T14:00:28-08:00

Example CMAF

https://cf98fa7b2ee4450e.mediapackage.us-east-1.amazonaws.com/out/v1/064134724fd74667ba294657a674ae72/manifest_id/news.m3u8?start=2018-04-04T01:14:00-08:00&end=2018-04-04T02:15:00-08:00

- Path elements – start and end parameters are included in the path of the request URL

Example HLS

https://cf98fa7b2ee4450e.mediapackage.us-east-1.amazonaws.com/out/v1/064134724fd74667ba294657a674ae72/start/1513717228/end/1513720828/comedy.m3u8

Example CMAF

https://cf98fa7b2ee4450e.mediapackage.us-east-1.amazonaws.com/out/v1/064134724fd74667ba294657a674ae72/manifest_id/start/1522807213/end/1522800013/news.m3u8

Microsoft smooth parameter rules

Start and end parameters in the URL request for Microsoft Smooth Streaming content can be included as path elements in the URL.

- Path elements – start and end parameters are included in the path of the request URL

Example

https://cf98fa7b2ee4450e.mediapackage.us-east-1.amazonaws.com/out/v1/1f76b3b4f94c44a485c0e4e560afe50e/start/1513717228/end/1513720828/drama.ism/Manifest
Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security of the cloud and security in the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS compliance programs. To learn about the compliance programs that apply to AWS Elemental MediaPackage, see AWS Services in Scope by Compliance Program.

- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company’s requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using MediaPackage. The following topics show you how to configure MediaPackage to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your MediaPackage resources.

**Topics**
- Data protection in AWS Elemental MediaPackage (p. 121)
- Identity and Access Management in AWS Elemental MediaPackage (p. 122)
- Logging and monitoring in AWS Elemental MediaPackage (p. 137)
- Compliance validation for AWS Elemental MediaPackage (p. 138)
- Resilience in AWS Elemental MediaPackage (p. 138)
- Infrastructure security in AWS Elemental MediaPackage (p. 139)

## Data protection in AWS Elemental MediaPackage

The AWS shared responsibility model applies to data protection in AWS Elemental MediaPackage. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. This content includes the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the AWS Security Blog.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual user accounts with AWS Identity and Access Management (IAM). That way each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We recommend TLS 1.2 or later.
• Set up API and user activity logging with AWS CloudTrail.
• Use AWS encryption solutions, along with all default security controls within AWS services.
• Use advanced managed security services such as Amazon Macie, which assists in discovering and securing personal data that is stored in Amazon S3.
• If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-2.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form fields such as a Name field. This includes when you work with MediaPackage or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

Topics
• Implementing DRM with AWS Elemental MediaPackage (p. 122)
• Implementing CDN authorization with AWS Elemental MediaPackage (p. 122)

Implementing DRM with AWS Elemental MediaPackage

Use encryption to protect your content from unauthorized access. MediaPackage supports digital rights management (DRM). With DRM, you can make sure that once you distribute your content, only authorized viewers can watch it.

For information about using DRM with MediaPackage, see Content encryption and DRM in AWS Elemental MediaPackage (p. 93).

Implementing CDN authorization with AWS Elemental MediaPackage

Use content delivery network (CDN) authorization to ensure only authorized devices can access your content. With CDN authorization, playback requests must include the appropriate header and authorization code that you create. MediaPackage refuses playback requests that don't include the correct code.

For more information about CDN authorization, see CDN authorization in AWS Elemental MediaPackage (p. 89).

Identity and Access Management in AWS Elemental MediaPackage

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be authenticated (signed in) and authorized (have permissions) to use MediaPackage resources. IAM is an AWS service that you can use with no additional charge.

Topics
Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work that you do in MediaPackage.

Service user – If you use the MediaPackage service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more MediaPackage features to do your work, you might need additional permissions. Understanding how access is managed can help you request the right permissions from your administrator. If you cannot access a feature in MediaPackage, see Troubleshooting AWS Elemental MediaPackage identity and access (p. 133).

Service administrator – If you're in charge of MediaPackage resources at your company, you probably have full access to MediaPackage. It's your job to determine which MediaPackage features and resources your employees should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with MediaPackage, see How AWS Elemental MediaPackage Works with IAM (p. 127).

IAM administrator – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to MediaPackage. To view example MediaPackage identity-based policies that you can use in IAM, see AWS Elemental MediaPackage Identity-based policy examples (p. 129).

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. For more information about signing in using the AWS Management Console, see Signing in to the AWS Management Console as an IAM user or root user in the IAM User Guide.

You must be authenticated (signed in to AWS) as the AWS account root user, an IAM user, or by assuming an IAM role. You can also use your company's single sign-on authentication or even sign in using Google or Facebook. In these cases, your administrator previously set up identity federation using IAM roles. When you access AWS using credentials from another company, you are assuming a role indirectly.

To sign in directly to the AWS Management Console, use your password with your root user email address or your IAM user name. You can access AWS programmatically using your root user or IAM users access keys. AWS provides SDK and command line tools to cryptographically sign your request using your credentials. If you don't use AWS tools, you must sign the request yourself. Do this using Signature Version 4, a protocol for authenticating inbound API requests. For more information about authenticating requests, see Signature Version 4 signing process in the AWS General Reference.

Regardless of the authentication method that you use, you might also be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see Using multi-factor authentication (MFA) in AWS in the IAM User Guide.
AWS account root user

When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account root user and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

IAM users and groups

An IAM user is an identity within your AWS account that has specific permissions for a single person or application. An IAM user can have long-term credentials such as a user name and password or a set of access keys. To learn how to generate access keys, see Managing access keys for IAM users in the IAM User Guide. When you generate access keys for an IAM user, make sure you view and securely save the key pair. You cannot recover the secret access key in the future. Instead, you must generate a new access key pair.

An IAM group is an identity that specifies a collection of IAM users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier to manage for large sets of users. For example, you could have a group named IAMAdmins and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. To learn more, see When to create an IAM user (instead of a role) in the IAM User Guide.

IAM roles

An IAM role is an identity within your AWS account that has specific permissions. It is similar to an IAM user, but is not associated with a specific person. You can temporarily assume an IAM role in the AWS Management Console by switching roles. You can assume a role by calling an AWS CLI or AWS API operation or by using a custom URL. For more information about methods for using roles, see Using IAM roles in the IAM User Guide.

IAM roles with temporary credentials are useful in the following situations:

- **Temporary IAM user permissions** – An IAM user can assume an IAM role to temporarily take on different permissions for a specific task.

- **Federated user access** – Instead of creating an IAM user, you can use existing identities from AWS Directory Service, your enterprise user directory, or a web identity provider. These are known as federated users. AWS assigns a role to a federated user when access is requested through an identity provider. For more information about federated users, see Federated users and roles in the IAM User Guide.

- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see How IAM roles differ from resource-based policies in the IAM User Guide.

- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.
Managing access using policies

You control access in AWS by creating policies and attaching them to IAM identities or AWS resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. You can sign in as the root user or an IAM user, or you can assume an IAM role. When you then make a request, AWS evaluates the related identity-based or resource-based policies. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see Overview of JSON policies in the IAM User Guide.

Administrators can use AWS JSON policies to specify who has access to what. That is, which principal can perform actions on what resources, and under what conditions.

Every IAM entity (user or role) starts with no permissions. In other words, by default, users can do nothing, not even change their own password. To give a user permission to do something, an administrator must attach a permissions policy to a user. Or the administrator can add the user to a group that has the intended permissions. When an administrator gives permissions to a group, all users in that group are granted those permissions.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, suppose that you have a policy that allows the iam:GetRole action. A user with that policy can get role information from the AWS Management Console, the AWS CLI, or the AWS API.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Creating IAM policies in the IAM User Guide.

Identity-based policies can be further categorized as inline policies or managed policies. Inline policies are embedded directly into a single user, group, or role. Managed policies are standalone policies that
you can attach to multiple users, groups, and roles in your AWS account. Managed policies include AWS managed policies and customer managed policies. To learn how to choose between a managed policy or an inline policy, see Choosing between managed policies and inline policies in the IAM User Guide.

**Resource-based policies**

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM role trust policies and Amazon S3 bucket policies. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

**Access control lists (ACLs)**

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

Amazon S3, AWS WAF, and Amazon VPC are examples of services that support ACLs. To learn more about ACLs, see Access control list (ACL) overview in the Amazon Simple Storage Service Developer Guide.

**Other policy types**

AWS supports additional, less-common policy types. These policy types can set the maximum permissions granted to you by the more common policy types.

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the Principal field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see Permissions boundaries for IAM entities in the IAM User Guide.

- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see How SCPs work in the AWS Organizations User Guide.

- **Session policies** – Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. For more information, see Session policies in the IAM User Guide.

**Multiple policy types**

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see Policy evaluation logic in the IAM User Guide.
How AWS Elemental MediaPackage Works with IAM

Before you use IAM to manage access to MediaPackage, you should understand what IAM features are available to use with MediaPackage. To get a high-level view of how MediaPackage and other AWS services work with IAM, see AWS Services That Work with IAM in the IAM User Guide.

Topics
- MediaPackage Identity-Based Policies (p. 127)
- MediaPackage Resource-based policies (p. 128)
- Authorization based on MediaPackage tags (p. 129)
- MediaPackage IAM Roles (p. 129)

MediaPackage Identity-Based Policies

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. MediaPackage supports specific actions, resources, and condition keys. To learn about all the elements that you use in a JSON policy, see IAM JSON policy elements reference in the IAM User Guide.

Actions

Administrators can use AWS JSON policies to specify who has access to what. That is, which principal can perform actions on what resources, and under what conditions.

The Action element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Policy actions usually have the same name as the associated AWS API operation. There are some exceptions, such as permission-only actions that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called dependent actions.

Include actions in a policy to grant permissions to perform the associated operation.

Policy actions in MediaPackage use the following prefix before the action: mediapackage:. For example, to grant someone permission to delete a MediaPackage endpoint with the MediaPackage DeleteOriginEndpoint API operation, you include the mediapackage:DeleteOriginEndpoint action in their policy. Policy statements must include either an Action or NotAction element. MediaPackage defines its own set of actions that describe tasks that you can perform with this service.

To specify multiple actions in a single statement, separate them with commas as follows:

```
"Action": [
    "mediapackage:action1",
    "mediapackage:action2"
]
```

You can specify multiple actions using wildcards (*). For example, to specify all actions that begin with the word Describe, include the following action:

```
"Action": "mediapackage:Describe*
```

For a list of MediaPackage actions, see Actions Defined by AWS Elemental MediaPackage in the IAM User Guide.

Resources

Administrators can use AWS JSON policies to specify who has access to what. That is, which principal can perform actions on what resources, and under what conditions.
The **Resource** JSON policy element specifies the object or objects to which the action applies. Statements must include either a **Resource** or a **NotResource** element. As a best practice, specify a resource using its [Amazon Resource Name (ARN)](https://docs.aws.amazon.com/general/latest/gr/aws-arns-and-namespaces.html). You can do this for actions that support a specific resource type, known as **resource-level permissions**.

For actions that don’t support resource-level permissions, such as listing operations, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*
```

MediaPackage has the following resource ARNs:

```
arn:${Partition}:mediapackage:${Region}:${Account}:channels/${channelID}
arn:${Partition}:mediapackage:${Region}:${Account}:origin_endpoints/${endpointID}
```

For more information about the format of ARNs, see [Amazon Resource Names (ARNs) and AWS Service Namespaces](https://docs.aws.amazon.com/insurance/latest/userguide/aws-service-namespaces.html).

For example, to specify the 9a6b3953e242400eb805f324d95788e3 channel in your statement, use the following ARN:

```
"Resource": "arn:aws:mediapackage:us-east-1:111122223333:channels/9a6b3953e242400eb805f324d95788e3"
```

To specify all instances that belong to a specific account, use the wildcard (*):

```
```

Some MediaPackage actions, such as those for creating resources, can’t be performed on a specific resource. In those cases, you must use the wildcard (*).

```
"Resource": "*
```

To see a list of MediaPackage resource types and their ARNs, see [Resources Defined by AWS Elemental MediaPackage](https://docs.aws.amazon.com/insurance/latest/userguide/media-package-resource-arns.html) in the **IAM User Guide**. To learn with which actions you can specify the ARN of each resource, see [Actions Defined by AWS Elemental MediaPackage](https://docs.aws.amazon.com/insurance/latest/userguide/media-package-actions.html).

### Condition keys

MediaPackage doesn’t provide any service-specific condition keys, but it does support using some global condition keys. To see all AWS global condition keys, see [AWS global condition context keys](https://docs.aws.amazon.com/insurance/latest/userguide/aws-service-namespaces.html) in the **IAM User Guide**.

### Examples

For examples of MediaPackage identity-based policies, see [AWS Elemental MediaPackage Identity-based policy examples](https://docs.aws.amazon.com/insurance/latest/userguide/media-package-identity-based-policy-examples.html) (p. 129).

### MediaPackage Resource-based policies

MediaPackage doesn’t support resource-based policies.
Authorization based on MediaPackage tags

You can attach tags to MediaPackage resources or pass tags in a request to MediaPackage. To control access based on tags, you provide tag information in the condition element of a policy using the mediapackage:ResourceTag/key-name, aws:RequestTag/key-name, or aws:TagKeys condition keys. For more information about tagging MediaPackage resources, see Tagging AWS Elemental MediaPackage resources (p. 163).

To view an example identity-based policy for limiting access to a resource based on the tags on that resource, see Viewing MediaPackage Channels based on tags (p. 131).

MediaPackage IAM Roles

An IAM role is an entity within your AWS account that has specific permissions.

Using temporary credentials with MediaPackage

You can use temporary credentials to sign in with federation, assume an IAM role, or assume a cross-account role. You obtain temporary security credentials by calling AWS STS API operations such as AssumeRole or GetFederationToken.

MediaPackage supports using temporary credentials.

Service-linked roles

Service-linked roles allow AWS services to access resources in other services to complete an action on your behalf. Service-linked roles appear in your IAM account and are owned by the service. An IAM administrator can view but not edit the permissions for service-linked roles.

MediaPackage does not support service-linked roles.

Service roles

This feature allows a service to assume a service role on your behalf. This role allows the service to access resources in other services to complete an action on your behalf. Service roles appear in your IAM account and are owned by the account. This means that an IAM administrator can change the permissions for this role. However, doing so might break the functionality of the service.

MediaPackage supports service roles.

Choosing an IAM role in MediaPackage

When you create an asset resource in MediaPackage, you must choose a role to allow MediaPackage to access Amazon S3 on your behalf. If you previously created a service role or service-linked role, MediaPackage provides you with a list of roles to choose from. It's important to choose a role that allows access to read from the S3 bucket and retrieve content. For more information, see Allowing AWS Elemental MediaPackage to access other AWS services (p. 21).

AWS Elemental MediaPackage Identity-based policy examples

By default, IAM users and roles don't have permission to create or modify MediaPackage resources. They also can't perform tasks using the AWS Management Console, AWS CLI, or an AWS API. An IAM
The administrator must create IAM policies that grant users and roles permission to perform specific API operations on the specified resources they need. The administrator must then attach those policies to the IAM users or groups that require those permissions.

To learn how to create an IAM identity-based policy using these example JSON policy documents, see Creating policies on the JSON tab in the IAM User Guide.

**Topics**

- Policy best practices (p. 130)
- Using the MediaPackage Console (p. 130)
- Allow users to view their own permissions (p. 131)
- Viewing MediaPackage Channels based on tags (p. 131)

**Policy best practices**

Identity-based policies are very powerful. They determine whether someone can create, access, or delete MediaPackage resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started using AWS managed policies** – To start using MediaPackage quickly, use AWS managed policies to give your employees the permissions they need. These policies are already available in your account and are maintained and updated by AWS. For more information, see Get started using permissions with AWS managed policies in the IAM User Guide.

- **Grant least privilege** – When you create custom policies, grant only the permissions required to perform a task. Start with a minimum set of permissions and grant additional permissions as necessary. Doing so is more secure than starting with permissions that are too lenient and then trying to tighten them later. For more information, see Grant least privilege in the IAM User Guide.

- **Enable MFA for sensitive operations** – For extra security, require IAM users to use multi-factor authentication (MFA) to access sensitive resources or API operations. For more information, see Using multi-factor authentication (MFA) in AWS in the IAM User Guide.

- **Use policy conditions for extra security** – To the extent that it’s practical, define the conditions under which your identity-based policies allow access to a resource. For example, you can write conditions to specify a range of allowable IP addresses that a request must come from. You can also write conditions to allow requests only within a specified date or time range, or to require the use of SSL or MFA. For more information, see IAM JSON policy elements: Condition in the IAM User Guide.

**Using the MediaPackage Console**

To access the AWS Elemental MediaPackage console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the MediaPackage resources in your AWS account. If you create an identity-based policy that is more restrictive than the minimum required permissions, the console won’t function as intended for entities (IAM users or roles) with that policy.

To ensure that those entities can still use the MediaPackage console, also attach the following AWS managed policy to the entities. For more information, see Adding permissions to a user in the IAM User Guide:

```
AWSElementalMediaPackageReadOnly
```

You don’t need to allow minimum console permissions for users that are making calls only to the AWS CLI or an AWS API. Instead, allow access to only the actions that match the API operation that you’re trying to perform.
Allow users to view their own permissions

This example shows how you might create a policy that allows IAM users to view the inline and managed policies that are attached to their user identity. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsForUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"],
      "Resource": ["*"
    }
  ],
  "Sid": "NavigateInConsole",
  "Effect": "Allow",
  "Action": [
    "iam:GetGroupPolicy",
    "iam:GetPolicyVersion",
    "iam:GetPolicy",
    "iam:ListAttachedGroupPolicies",
    "iam:ListGroupPolicies",
    "iam:ListPolicyVersions",
    "iam:ListPolicies",
    "iam:ListUsers"
  ],
  "Resource": ["*"
  }
}
```

Viewing MediaPackage Channels based on tags

You can use conditions in your identity-based policy to control access to MediaPackage resources based on tags. This example shows how you might create a policy that allows viewing a channel. However, permission is granted only if the channel tag `Owner` has the value of that user’s user name. This policy also grants the permissions necessary to complete this action on the console.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": ["mediapackage:DescribeChannel"],
      "Resource": ["arn:aws:mediapackage::*:channels/*"],
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Owner": "${aws:username}"
        }
      }
    },
    {
      "Sid": "VisualEditor1",
      "Effect": "Allow",
      "Action": ["mediapackage:DescribeChannel"],
      "Resource": ["arn:aws:mediapackage::*:channels/*"],
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Owner": "${aws:username}"
        }
      }
    }
  ]
}
```
"Effect": "Allow",
"Action": "mediapackage:ListChannels",
"Resource": "*",
"Condition": {
  "StringEquals": {
    "aws:ResourceTag/Owner": "${aws:username}"  
  }
}
]
}

You can attach this policy to the IAM users in your account. If a user named richard-roehattempts to view a MediaPackage channel, the channel must be tagged Owner=richard-roere or owner=richard-roе. Otherwise he is denied access. The condition tag key Owner matches both Owner and owner because condition key names are not case-sensitive. For more information, see IAM JSON policy elements: Condition in the IAM User Guide.

IAM policy examples for secrets in AWS Secrets Manager

During setup, you create an IAM policy (p. 21) that you assign to AWS Elemental MediaPackage. This policy allows AWS Elemental MediaPackage to read secrets that you have stored in AWS Secrets Manager. The settings for this policy are entirely up to you. The policy can range from most restrictive (allowing access to only specific secrets) to least restrictive (allowing access to any secret that you create using this AWS account). We recommend using the most restrictive policy as a best practice. However, the examples in this section show you how to set up policies with different levels of restriction. Because AWS Elemental MediaPackage needs only read access to secrets, all the examples in this section show only the actions necessary to read the values that you store.

Topics

- Allow read access to specific secrets in AWS Secrets Manager (p. 132)
- Allow read access to all secrets created in a specific Region in AWS Secrets Manager (p. 133)
- Allow read access to all resources in AWS Secrets Manager (p. 133)

Allow read access to specific secrets in AWS Secrets Manager

The following IAM policy allows read access to specific resources (secrets) that you create in AWS Secrets Manager.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    
    "Effect": "Allow",
    "Action": [
      "secretsmanager:GetResourcePolicy",
      "secretsmanager:GetSecretValue",
      "secretsmanager:DescribeSecret",
      "secretsmanager:ListSecretVersionIds"
    ],
    "Resource": [
      "arn:aws:secretsmanager:us-west-2:111122223333:secret:aes256-7g8H9l"
    ]
  ]
}
```
Allow read access to all secrets created in a specific Region in AWS Secrets Manager

The following IAM policy allows read access to all secrets that you create in a specific AWS Region in AWS Secrets Manager. This policy applies to resources that you have created already and all resources that you create in the future in the specified Region.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetResourcePolicy",
                "secretsmanager:GetSecretValue",
                "secretsmanager:DescribeSecret",
                "secretsmanager:ListSecretVersionIds"
            ],
            "Resource": [
            ]
        }
    ]
}
```

Allow read access to all resources in AWS Secrets Manager

The following IAM policy allows read access to all resources that you create in AWS Secrets Manager. This policy applies to resources that you have created already and all resources that you create in the future.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetResourcePolicy",
                "secretsmanager:GetSecretValue",
                "secretsmanager:DescribeSecret",
                "secretsmanager:ListSecretVersionIds"
            ],
            "Resource": ["*"]
        }
    ]
}
```

Troubleshooting AWS Elemental MediaPackage identity and access

Use the following information to help you diagnose and fix common issues that you might encounter when working with MediaPackage and IAM.

Topics

- I am not authorized to perform an action in MediaPackage (p. 134)
I am not authorized to perform an action in MediaPackage

If the AWS Management Console tells you that you're not authorized to perform an action, then you must contact your administrator for assistance. Your administrator is the person that provided you with your user name and password.

The following example error occurs when the mateojackson IAM user tries to use the console to view the channels on the account but does not have mediapackage:ListChannels permissions.

```
```

In this case, Mateo asks his administrator to update his policies to allow him to access the channels resource using the mediapackage:ListChannels action.

I Am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the iam:PassRole action, then you must contact your administrator for assistance. Your administrator is the person that provided you with your user name and password. Ask that person to update your policies to allow you to pass a role to MediaPackage.

Some AWS services allow you to pass an existing role to that service, instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named marymajor tries to use the console to perform an action in MediaPackage. However, the action requires the service to have permissions granted by a service role. Mary does not have permissions to pass the role to the service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform: iam:PassRole
```

In this case, Mary asks her administrator to update her policies to allow her to perform the iam:PassRole action.

I want to view my access keys

After you create your IAM user access keys, you can view your access key ID at any time. However, you can't view your secret access key again. If you lose your secret key, you must create a new access key pair.

Access keys consist of two parts: an access key ID (for example, AKIAIOSFODNN7EXAMPLE) and a secret access key (for example, wJalrXUtUEM5/E8jEAp50ym0XQyKcCZnEzbyWJ3Q). Like a user name and password, you must use both the access key ID and secret access key together to authenticate your requests. Manage your access keys as securely as you do your user name and password.

**Important**

Do not provide your access keys to a third party, even to help find your canonical user ID. By doing this, you might give someone permanent access to your account.

When you create an access key pair, you are prompted to save the access key ID and secret access key in a secure location. The secret access key is available only at the time you create it. If you lose your secret
access key, you must add new access keys to your IAM user. You can have a maximum of two access keys. If you already have two, you must delete one key pair before creating a new one. To view instructions, see Managing access keys in the IAM User Guide.

I'm an administrator and want to allow others to access MediaPackage

To allow others to access MediaPackage, you must create an IAM entity (user or role) for the person or application that needs access. They will use the credentials for that entity to access AWS. You must then attach a policy to the entity that grants them the correct permissions in MediaPackage.

To get started right away, see Creating your first IAM delegated user and group in the IAM User Guide.

For instruction specific to MediaPackage, see Creating a non-admin IAM user (p. 17).

I want to allow people outside of my AWS account to access my MediaPackage resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether MediaPackage supports these features, see How AWS Elemental MediaPackage Works with IAM (p. 127).
- To learn how to provide access to your resources across AWS accounts that you own, see Providing access to an IAM user in another AWS account that you own in the IAM User Guide.
- To learn how to provide access to your resources to third-party AWS accounts, see Providing access to AWS accounts owned by third parties in the IAM User Guide.
- To learn how to provide access through identity federation, see Providing access to externally authenticated users (identity federation) in the IAM User Guide.
- To learn the difference between using roles and resource-based policies for cross-account access, see How IAM roles differ from resource-based policies in the IAM User Guide.

Learn More

For more information about identity and access management for MediaPackage, continue to the following pages:

- How AWS Elemental MediaPackage Works with IAM (p. 127)
- AWS Elemental MediaPackage Identity-based policy examples (p. 129)
- Troubleshooting AWS Elemental MediaPackage identity and access (p. 133)

Using Service-Linked Roles for MediaPackage

AWS Elemental MediaPackage uses AWS Identity and Access Management (IAM) service-linked roles. A service-linked role is a unique type of IAM role that is linked directly to MediaPackage. Service-linked roles are predefined by MediaPackage and include all the permissions that the service requires to call other AWS services on your behalf.
A service-linked role makes setting up MediaPackage easier because you don't have to manually add the necessary permissions. MediaPackage defines the permissions of its service-linked roles, and unless defined otherwise, only MediaPackage can assume its roles. The defined permissions include the trust policy and the permissions policy, and that permissions policy cannot be attached to any other IAM entity.

You can delete a service-linked role only after first deleting their related resources. This protects your MediaPackage resources because you can't inadvertently remove permission to access the resources.

For information about other services that support service-linked roles, see AWS Services That Work with IAM and look for the services that have Yes in the Service-Linked Role column. Choose a Yes with a link to view the service-linked role documentation for that service.

**Service-Linked Role Permissions for MediaPackage**

MediaPackage uses the service-linked role named `AWSServiceRoleForMediaPackage` – MediaPackage uses this service-linked role to invoke CloudWatch to create and manage log groups, log streams, and log events.

The AWSServiceRoleForMediaPackage service-linked role trusts the following services to assume the role:

- mediapackage.amazonaws.com

The role permissions policy allows MediaPackage to complete the following actions on the specified resources:

- Action: `logs:PutLogEvents` on `arn:aws:logs:*:*:log-group:/aws/MediaPackage/*:log-stream:*`
- Action: `logs:CreateLogStream, logs:CreateLogGroup, logs:DescribeLogGroups, logs:DescribeLogStreams` on `arn:aws:logs:*:*:log-group:/aws/MediaPackage/*`

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. For more information, see Service-Linked Role Permissions in the IAM User Guide.

**Creating a Service-Linked Role for MediaPackage**

You don't need to manually create a service-linked role. When you enable access logging in the AWS Management Console, the AWS CLI, or the AWS API, MediaPackage creates the service-linked role for you.

If you delete this service-linked role, and then need to create it again, you can use the same process to recreate the role in your account. When you enable access logging, MediaPackage creates the service-linked role for you again.

You can also use the IAM console to create a service-linked role with the MediaPackage use case. In the AWS CLI or the AWS API, create a service-linked role with the mediapackage.amazonaws.com service name. For more information, see Creating a Service-Linked Role in the IAM User Guide. If you delete this service-linked role, you can use this same process to create the role again.

**Editing a Service-Linked Role for MediaPackage**

MediaPackage does not allow you to edit the AWSServiceRoleForMediaPackage service-linked role. After you create a service-linked role, you cannot change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM. For more information, see Editing a Service-Linked Role in the IAM User Guide.
Deleting a Service-Linked Role for MediaPackage

If you no longer need to use a feature or service that requires a service-linked role, we recommend that you delete that role. That way you don’t have an unused entity that is not actively monitored or maintained. However, you must clean up the resources for your service-linked role before you can manually delete it.

**Note**

If the MediaPackage service is using the role when you try to delete the resources, then the deletion might fail. If that happens, wait for a few minutes and try the operation again.

To delete MediaPackage resources used by the AWSServiceRoleForMediaPackage

- Disable access logging in the AWS Management Console, the AWS CLI, or the AWS API.

To manually delete the service-linked role using IAM

Use the IAM console, the AWS CLI, or the AWS API to delete the AWSServiceRoleForMediaPackage service-linked role. For more information, see Deleting a Service-Linked Role in the IAM User Guide.

Supported Regions for MediaPackage Service-Linked Roles

MediaPackage supports using service-linked roles in all of the regions where the service is available. For more information, see AWS Regions and Endpoints.

Logging and monitoring in AWS Elemental MediaPackage

This section provides an overview of the options for logging and monitoring in AWS Elemental MediaPackage for security purposes. For more information about logging and monitoring in MediaPackage see Monitoring AWS Elemental MediaPackage (p. 140).

Monitoring is an important part of maintaining the reliability, availability, and performance of AWS Elemental MediaPackage and your AWS solutions. You should collect monitoring data from all parts of your AWS solution so that you can more easily debug a multi-point failure if one occurs. AWS provides several tools for monitoring your MediaPackage resources and responding to potential incidents.

Amazon CloudWatch alarms

Using CloudWatch alarms, you watch a single metric over a time period that you specify. If the metric exceeds a given threshold, a notification is sent to an Amazon SNS topic or AWS Auto Scaling policy. CloudWatch alarms do not invoke actions because they are in a particular state. Rather, the state must have changed and been maintained for a specified number of periods. For more information, see Monitoring AWS Elemental MediaPackage with Amazon CloudWatch metrics (p. 140).

AWS CloudTrail logs

CloudTrail provides a record of actions taken by a user, role, or an AWS service in AWS Elemental MediaPackage. Using the information collected by CloudTrail, you can determine the request that was made to MediaPackage, the IP address from which the request was made, who made the request, when it was made, and additional details. For more information, see Logging AWS Elemental MediaPackage API calls with AWS CloudTrail (p. 155).
AWS Elemental MediaPackage access logs

Server access logs provide detailed records about requests that are made to a channel. Server access logs are useful for many applications. For example, access log information can be useful in security and access audits. For more information, see Access logging (p. 157).

AWS Trusted Advisor

Trusted Advisor draws upon best practices learned from serving hundreds of thousands of AWS customers. Trusted Advisor inspects your AWS environment and then makes recommendations when opportunities exist to save money, improve system availability and performance, or help close security gaps. All AWS customers have access to five Trusted Advisor checks. Customers with a Business or Enterprise support plan can view all Trusted Advisor checks.

For more information, see AWS Trusted Advisor.

Compliance validation for AWS Elemental MediaPackage

MediaPackage is not in scope of any AWS compliance programs.

For a list of AWS services in scope of specific compliance programs, see AWS services in scope by compliance program. For general information, see AWS compliance programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using MediaPackage is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- Security and compliance quick start guides – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- Architecting for HIPAA security and compliance whitepaper – This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.
- AWS compliance resources – This collection of workbooks and guides might apply to your industry and location.
- AWS Config – This AWS service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.

Resilience in AWS Elemental MediaPackage

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between Availability Zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.
For more information about AWS Regions and Availability Zones, see AWS global infrastructure.

Infrastructure security in AWS Elemental MediaPackage

As a managed service, AWS Elemental MediaPackage is protected by the AWS global network security procedures that are described in the Amazon Web Services: Overview of security processes whitepaper.

You use AWS published API calls to access MediaPackage through the network. Clients must support Transport Layer Security (TLS) 1.0 or later. We recommend TLS 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.
Monitoring AWS Elemental MediaPackage

Monitoring is an important part of maintaining the reliability, availability, and performance of AWS Elemental MediaPackage and your other AWS solutions. AWS provides the following monitoring tools to watch MediaPackage, report when something is wrong, and take automatic actions when appropriate:

- **Amazon CloudWatch** monitors your AWS resources and the applications that you run on AWS in real-time. You can collect and track metrics, create customized dashboards, and set alarms that notify you or take actions when a specified metric reaches a threshold that you specify. For example, you can have CloudWatch track CPU usage or other metrics of your Amazon EC2 instances and automatically launch new instances when needed. For more information, see the Amazon CloudWatch User Guide.

- **Amazon CloudWatch Events** delivers a near real-time stream of system events that describe changes in AWS resources. CloudWatch Events enables automated event-driven computing, as you can write rules that watch for certain events and trigger automated actions in other AWS services when these events happen. For more information, see the Amazon CloudWatch Events User Guide.

- **AWS CloudTrail** captures API calls and related events made by or on behalf of your AWS account and delivers the log files to an Amazon S3 bucket that you specify. You can identify which users and accounts called AWS, the source IP address from which the calls were made, and when the calls occurred. For more information, see the AWS CloudTrail User Guide.

Topics
- Monitoring AWS Elemental MediaPackage with Amazon CloudWatch metrics (p. 140)
- Monitoring AWS Elemental MediaPackage with CloudWatch Events (p. 149)
- Logging AWS Elemental MediaPackage API calls with AWS CloudTrail (p. 155)
- Access logging (p. 157)

Monitoring AWS Elemental MediaPackage with Amazon CloudWatch metrics

You can monitor AWS Elemental MediaPackage using CloudWatch, which collects raw data and processes it into readable, near real-time metrics. These statistics are kept for 15 months, so that you can access historical information and gain a better perspective on how your web application or service is performing. You can also set alarms that watch for certain thresholds, and send notifications or take actions when those thresholds are met. For more information, see the Amazon CloudWatch User Guide.

To view metrics using the AWS Elemental MediaPackage console

AWS Elemental MediaPackage displays metrics throughout the console.

1. Open the MediaPackage console at https://console.aws.amazon.com/mediapackage/.
2. Navigate to the appropriate page to view metrics:
   - For metrics on all channels and endpoints in the AWS Region, go to the Channels page.
   - For metrics on a specific channel and all of its endpoints, go to the channel's details page.
   - For metrics on a specific endpoint and its channel, go to the endpoint's details page.
3. (Optional) To refine the metrics view, choose Open in CloudWatch.
**To view metrics using the CloudWatch console**

Metrics are grouped first by the service namespace, and then by the various dimension combinations within each namespace.

1. Sign in to the AWS Management Console and open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose Metrics.
3. Under All metrics, choose the AWS/MediaPackage namespace.
4. Choose the metric dimension to view the metrics (for example, choose channel to view metrics per channel).

**To view metrics using the AWS CLI**

At a command prompt, use the following command:

```bash
aws cloudwatch list-metrics --namespace "AWS/MediaPackage"
```

**Topics**

- AWS Elemental MediaPackage live content metrics (p. 141)
- AWS Elemental MediaPackage VOD content metrics (p. 146)

**AWS Elemental MediaPackage live content metrics**

The AWS/MediaPackage namespace includes the following metrics for live content. AWS Elemental MediaPackage publishes metrics to CloudWatch every minute, if not sooner.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActiveInput</td>
<td>Indicates if an input has been used as the source for an endpoint in AWS Elemental MediaPackage (it has been active). A value of 1 indicates that the input was active, and a 0 (zero) indicates that it wasn't. Units: None</td>
</tr>
<tr>
<td>EgressBytes</td>
<td>Number of bytes that AWS Elemental MediaPackage successfully sends for each request. If MediaPackage doesn't receive any requests for output in the specified interval, then no data is given. Units: Bytes</td>
</tr>
</tbody>
</table>

Valid statistics:

- Average – average bytes (Sum/SampleCount) that AWS Elemental MediaPackage outputs over the configured interval.
### Metric

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum</td>
<td>• Maximum – largest individual output request (in bytes) made to AWS Elemental MediaPackage.</td>
</tr>
<tr>
<td>Minimum</td>
<td>• Minimum – smallest individual output request (in bytes) made to AWS Elemental MediaPackage.</td>
</tr>
<tr>
<td>SampleCount</td>
<td>• SampleCount – number of requests that is used in the statistical calculation.</td>
</tr>
<tr>
<td>Sum</td>
<td>• Sum – total number of bytes that AWS Elemental MediaPackage outputs over the configured interval.</td>
</tr>
</tbody>
</table>

Valid dimensions:
- Channel
- Combination of Channel and OriginEndpoint
- PackagingConfiguration
- No dimension

<table>
<thead>
<tr>
<th>EgressRequestCount</th>
<th>Number of content requests that AWS Elemental MediaPackage receives. If MediaPackage doesn't receive any requests for output in the specified interval, then no data is given. Units: Count</th>
</tr>
</thead>
</table>

Valid statistics:
- Sum – total number of output requests that AWS Elemental MediaPackage receives.

Valid dimensions:
- Channel
- Combination of Channel and OriginEndpoint
- StatusCodeRange
- Combination of Channel and StatusCodeRange
- Combination of Channel, OriginEndpoint, and StatusCodeRange
- PackagingConfiguration
- Combination of PackagingConfiguration and StatusCodeRange
- No dimension
## Live content metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EgressResponseTime</td>
<td>The time that it takes AWS Elemental MediaPackage to process each output request. If MediaPackage doesn't receive any requests for output in the specified interval, then no data is given. Units: Milliseconds</td>
</tr>
</tbody>
</table>

**Valid statistics:**

- **Average** – average amount of time (Sum/SampleCount) that it takes AWS Elemental MediaPackage to process output requests over the configured interval.
- **Maximum** – longest amount of time (in milliseconds) that it takes AWS Elemental MediaPackage to process an output request and provide a response.
- **Minimum** – shortest amount of time (in milliseconds) that it takes AWS Elemental MediaPackage to process an output request and provide a response.
- **SampleCount** – number of requests that is used in the statistical calculation.
- **Sum** – total amount of time that it takes AWS Elemental MediaPackage to process output requests over the configured interval.

**Valid dimensions:**

- Channel
- Combination of Channel and OriginEndpoint
- PackagingConfiguration
<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IngressBytes</td>
<td>Number of bytes of content that AWS Elemental MediaPackage receives for each input request. If MediaPackage doesn't receive any requests for input in the specified interval, then no data is given. Units: Bytes</td>
</tr>
</tbody>
</table>

Valid statistics:
- **Average** – average bytes \(\frac{\text{Sum}}{\text{SampleCount}}\) that AWS Elemental MediaPackage receives over the configured interval.
- **Maximum** – largest individual input request (in bytes) made to AWS Elemental MediaPackage.
- **Minimum** – smallest individual input request (in bytes) made to AWS Elemental MediaPackage.
- **SampleCount** – number of requests that is used in the statistical calculation.
- **Sum** – total number of bytes that AWS Elemental MediaPackage receives over the configured interval.

Valid dimensions:
- **Channel**
- **Combination of Channel and IngestEndpoint**
- **No dimension**
## IngressResponseTime

**Description:**

The time that it takes AWS Elemental MediaPackage to process each input request. If MediaPackage doesn't receive any requests for input in the specified interval, then no data is given.

**Units:** Milliseconds

**Valid statistics:**

- **Average** – average amount of time (Sum/SampleCount) that it takes AWS Elemental MediaPackage to process input requests over the configured interval.
- **Maximum** – longest amount of time (in milliseconds) that it takes AWS Elemental MediaPackage to process an input request and provide a response.
- **Minimum** – shortest amount of time (in milliseconds) that it takes AWS Elemental MediaPackage to process an input request and provide a response.
- **SampleCount** – number of requests that is used in the statistical calculation.
- **Sum** – total amount of time that it takes AWS Elemental MediaPackage to process input requests over the configured interval.

### Valid dimensions:

- **Channel**
- **Combination of Channel and IngestEndpoint**
- **No dimension**

## AWS Elemental MediaPackage live dimensions

You can filter the AWS/MediaPackage data using the following dimensions.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Dimension</td>
<td>Metrics are aggregated and shown for all channels, endpoints, or status codes.</td>
</tr>
<tr>
<td>Channel</td>
<td>Metrics are shown only for the specified channel.</td>
</tr>
<tr>
<td></td>
<td>Value: The autogenerated GUID of the channel.</td>
</tr>
<tr>
<td></td>
<td>Can be used alone or with other dimensions:</td>
</tr>
<tr>
<td></td>
<td>• Alone to show metrics for only the specified channel.</td>
</tr>
</tbody>
</table>
### AWS Elemental MediaPackage VOD content metrics

The **AWS/MediaPackage** namespace includes the following metrics for video on demand (VOD) content. AWS Elemental MediaPackage publishes metrics to CloudWatch every minute, if not sooner.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EgressBytes</td>
<td>Number of bytes that AWS Elemental MediaPackage successfully sends for each request.</td>
</tr>
</tbody>
</table>
## VOD content metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>If MediaPackage doesn't receive any requests for output in the specified interval, then no data is given.</td>
</tr>
<tr>
<td></td>
<td>Units: Bytes</td>
</tr>
<tr>
<td></td>
<td>Valid statistics:</td>
</tr>
<tr>
<td></td>
<td>• Average – average bytes (Sum/SampleCount) that AWS Elemental MediaPackage outputs over the configured interval.</td>
</tr>
<tr>
<td></td>
<td>• Maximum – largest individual output request (in bytes) made to AWS Elemental MediaPackage.</td>
</tr>
<tr>
<td></td>
<td>• Minimum – smallest individual output request (in bytes) made to AWS Elemental MediaPackage.</td>
</tr>
<tr>
<td></td>
<td>• SampleCount – number of requests that is used in the statistical calculation.</td>
</tr>
<tr>
<td></td>
<td>• Sum – total number of bytes that AWS Elemental MediaPackage outputs over the configured interval.</td>
</tr>
<tr>
<td></td>
<td>Valid dimensions:</td>
</tr>
<tr>
<td></td>
<td>• PackagingConfiguration</td>
</tr>
<tr>
<td>EgressRequestCount</td>
<td>Number of content requests that AWS Elemental MediaPackage receives. If MediaPackage doesn't receive any requests for output in the specified interval, then no data is given.</td>
</tr>
<tr>
<td></td>
<td>Units: Count</td>
</tr>
<tr>
<td></td>
<td>Valid statistics:</td>
</tr>
<tr>
<td></td>
<td>• Sum – total number of output requests that AWS Elemental MediaPackage receives.</td>
</tr>
<tr>
<td></td>
<td>Valid dimensions:</td>
</tr>
<tr>
<td></td>
<td>• PackagingConfiguration</td>
</tr>
<tr>
<td></td>
<td>• Combination of PackagingConfiguration and StatusCodeRange</td>
</tr>
</tbody>
</table>
### VOD content metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EgressResponseTime</td>
<td>The time that it takes AWS Elemental MediaPackage to process each output request. If MediaPackage doesn't receive any requests for output in the specified interval, then no data is given. Units: Milliseconds</td>
</tr>
</tbody>
</table>

Valid statistics:
- **Average** – average amount of time (Sum/SampleCount) that it takes AWS Elemental MediaPackage to process output requests over the configured interval.
- **Maximum** – longest amount of time (in milliseconds) that it takes AWS Elemental MediaPackage to process an output request and provide a response.
- **Minimum** – shortest amount of time (in milliseconds) that it takes AWS Elemental MediaPackage to process an output request and provide a response.
- **SampleCount** – number of requests that is used in the statistical calculation.
- **Sum** – total amount of time that it takes AWS Elemental MediaPackage to process output requests over the configured interval.

Valid dimensions:
- **PackagingConfiguration**

### AWS Elemental MediaPackage VOD dimensions

You can filter the AWS/MediaPackage data using the following dimensions.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Dimension</td>
<td>Metrics are aggregated and shown for all packaging configurations and status codes.</td>
</tr>
<tr>
<td>PackagingConfiguration</td>
<td>Metrics are shown only for the specified packaging configuration.</td>
</tr>
</tbody>
</table>

Value: The autogenerated GUID of the configuration.

Can be used alone or with other dimensions:
- Alone to show metrics for only the specified configuration.
### Dimension

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>StatusCodeRange</td>
<td>Metrics are shown for the specified status code range.</td>
</tr>
<tr>
<td></td>
<td>Value: 2xx, 3xx, 4xx, or 5xx.</td>
</tr>
<tr>
<td></td>
<td>Can be used alone or with other dimensions:</td>
</tr>
<tr>
<td></td>
<td>• Alone to show all output requests for the specified status range.</td>
</tr>
<tr>
<td></td>
<td>• With the channel dimension to show output requests for all endpoints that are associated with the specified channel, with the specified status code range.</td>
</tr>
<tr>
<td></td>
<td>• With the channel and originEndpoint dimensions to show output requests with a specific status code range on the specified endpoint that is associated with the specified channel.</td>
</tr>
</tbody>
</table>

---

## Monitoring AWS Elemental MediaPackage with CloudWatch Events

Amazon CloudWatch Events enables you to automate your AWS services and respond automatically to system events such as application availability issues or error conditions. AWS services deliver events to CloudWatch Events in near real-time. You can write simple rules to indicate which events are of interest to you, and what automated actions to take when an event matches a rule. The actions that can be automatically triggered include the following:

- Invoking an AWS Lambda function
- Invoking AWS Systems Manager Run Command
- Relaying the event to Amazon Kinesis Data Streams
- Activating an AWS Step Functions state machine

An example of using CloudWatch Events with AWS Elemental MediaPackage is notifying an Amazon SNS topic if you reach the maximum stream input. MediaPackage emits events on a best effort basis.

For more information about creating rules in CloudWatch Events, see the Amazon CloudWatch Events User Guide.

For a list of events that AWS Elemental MediaPackage emits, see AWS Elemental MediaPackage events (p. 149).

## AWS Elemental MediaPackage events

AWS Elemental MediaPackage integrates with Amazon CloudWatch Events to notify you of certain events that affect your channels and endpoints. Each event is represented in JSON (JavaScript Object Notation) and contains the event name, the date and time when the event occurred, the channel or
endpoint affected, and more. MediaPackage emits events on a best effort basis. You can use CloudWatch
Events to collect these events and set up rules that route them to one or more targets such as AWS
Lambda functions, Amazon SNS topics, Amazon SQS queues, streams in Amazon Kinesis Data Streams,
or built-in targets.

For more information about using CloudWatch Events with other kinds of events, see the Amazon
CloudWatch Events User Guide.

The following topics describe the CloudWatch Events that AWS Elemental MediaPackage creates.

Event types

- Input notification events (p. 150)
- Key provider notification events (p. 152)
- Harvest job notification events (p. 153)

Input notification events

You get input notification events for live and video on demand (VOD) content. These events notify you
when something happens with MediaPackage ingest. These are the input notification events you might
receive:

- Maximum input streams exceeded
- Input switch
- VOD ingest status change
- VOD playback readiness

The following sections describe each of these events.

Maximum Input Streams Exceeded Event

For live content, a channel in MediaPackage exceeds the quota for the number of input streams. For
information about quotas, see Quotas in AWS Elemental MediaPackage (p. 168).

Example

```
{
    "id": "7bf73129-1428-4cd3-a780-95db273d1602",
    "detail-type": "MediaPackage Input Notification",
    "source": "aws.mediapackage",
    "account": "aws_account_id",
    "time": "2015-11-11T21:29:54Z",
    "region": "us-west-2",
    "resources":
    ["arn:aws:mediapackage:us-west-2:aws_account_id:channels/262ff182d46d4b399fca#ea1364df682"
    ],
    "detail":
    {"event": "MaxIngestStreamsError",
    "message": "Parent Manifest [%s] has [23] streams, more than [20] allowed:
    (index_1.m3u8,index_2.m3u8,index_3.m3u8,index_4.m3u8,index_5.m3u8,index_6.m3u8,index_7.m3u8,index_8.m3u8,..."
    }
}
```

Input Switch Event

For live content, MediaPackage switches inputs for one of your endpoints.
One event is sent in a five-minute period. If the input switches multiple times in five minutes (for example, if MediaPackage switches to one input, then back to the other), you receive only one event.

For information about input redundancy and what causes inputs to switch, see Live input redundancy AWS Elemental MediaPackage processing flow (p. 10).

Example

```json
{
  "id": "8f9b8e72-0b31-e883-f19c-aec84742f3ce",
  "detail-type": "MediaPackage Input Notification",
  "source": "aws.mediapackage",
  "account": "aws_account_id",
  "time": "2018-07-16T17:29:36Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:mediapackage:us-east-1:aws_account_id:origin_endpoints/82d6b9bc04cb4612b487963d6c8d0f1a"
  ],
  "detail": {
    "event": "InputSwitchEvent",
    "message": "Origin endpoint experienced an Input Switch Event"
  }
}
```

VOD Ingest Status Event

For video on demand (VOD) content, an asset in MediaPackage changes ingest status. You get notifications for the following events:

- IngestStart
- IngestError
- IngestComplete

Example

```json
{
  "id": "8f9b8e72-0b31-e883-f19c-aec84742f3ce",
  "detail-type": "MediaPackage Input Notification",
  "source": "aws.mediapackage",
  "account": "aws_account_id",
  "time": "2019-05-03T17:29:36Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:mediapackage-vod:us-west-2:aws_account_id:assets/asset_name"
  ],
  "detail": {
    "event": "IngestComplete",
    "message": "message text"
  }
}
```

VOD Playback Event

For VOD content, an asset in MediaPackage is available for playback. There is a period of time between when asset ingest is complete, and when the asset can be played back. The event VodAssetPlayable means that MediaPackage can now fulfill playback requests for the asset.

You get individual VodAssetPlayable events for each packaging configuration in your packaging group. For example, if your packaging group contains one DASH and one HLS packaging
configuration, you receive two VodAssetPlayable events—one for your DASH packaging configuration, and one for your HLS packaging configuration.

Example

```json
{
    "id": "81e896e4-d9e5-ec79-f82a-b4cf3246c567",
    "detail-type": "MediaPackage Input Notification",
    "source": "aws.mediapackage",
    "account": "aws_account_id",
    "time": "2019-11-03T21:46:00Z",
    "region": "us-west-2",
    "resources": [
    ],
    "detail": {
        "event": "VodAssetPlayable",
        "message": "Asset 'asset_id' is now playable for PackagingConfiguration 'packaging_configuration_id'",
        "manifest_urls": ["https://555555555555.egress.mediapackage-vod.us-west-2.amazonaws.com/out/v1/b9cc115bf7f1a/b848dfb116920772aa69ba/a3c74b1ca6a451c/index.m3u8"
    }
}

{
    "id": "91e896e4-d9e5-ab80-f82a-b4cf3246c568",
    "detail-type": "MediaPackage Input Notification",
    "source": "aws.mediapackage",
    "account": "aws_account_id",
    "time": "2019-11-03T21:47:00Z",
    "region": "us-west-2",
    "resources": [
    ],
    "detail": {
        "event": "VodAssetPlayable",
        "message": "Asset 'asset_id' is now playable for PackagingConfiguration 'packaging_configuration_id'",
    }
}
```

Key provider notification events

You get key provider notification events when you’re using content encryption on an endpoint and MediaPackage can’t reach the key provider. For information about DRM and encryption, see https://docs.aws.amazon.com/speke/latest/documentation/.

Example

```json
{
...
}
Harvest job notification events

You get harvest job status events when you export a clip from a live stream to create a live-to-VOD asset. MediaPackage creates notifications when the harvest job succeeds or fails. For information about harvest jobs and live-to-VOD assets, see Creating live-to-VOD assets with AWS Elemental MediaPackage (p. 84).

Example Successful harvest job event

```json
{
  "id": "8f9b8e72-0b31-e883-f19c-aec84742f3ce",
  "detail-type": "MediaPackage HarvestJob Notification",
  "source": "aws.mediapackage",
  "account": "aws_account_id",
  "time": "2019-06-26T20:30:00-08:00",
  "region": "us-east-1",
  "resources": [
    "arn:aws:mediapackage:us-east-1:aws_account_id:harvest_jobs/harvest_job_id"
  ],
  "detail": {
    "harvest_job": {
      "id": "harvest_job_id",
      "status": "SUCCEEDED",
      "origin_endpoint_arn": "origin_endpoint_id",
      "start_time": "2019-06-26T20:30:00-08:00",
      "end_time": "2019-06-26T21:00:00-08:00",
      "s3_destination": {
        "bucket_name": "s3_bucket_name",
        "manifest_key": "path/and/manifest_name/index.m3u8",
        "role_arn": "arn:aws:iam::aws_account_id:role/S3Access_role",
      },
      "created_at": "2019-06-26T21:03:12-08:00"
    }
  }
}
```

Example Failed harvest job event

```json
{
  "id": "8f9b8e72-0b31-e883-f19c-aec84742f3ce",
  "detail-type": "MediaPackage HarvestJob Notification",
  "source": "aws.mediapackage",
  "account": "aws_account_id",
  "time": "2019-07-16T17:29:36Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:mediapackage:us-east-1:aws_account_id:origin_endpoints/70b44e2e666c4bdc9e5f4488e1f1aa99"
  ],
  "detail": {
    "event": "KeyProviderError",
    "message": "message-text"
  }
}
```
Creating event notifications

You can use Amazon CloudWatch Events and Amazon Simple Notification Service (Amazon SNS) to notify you of new events. In CloudWatch Events, the rule describes which events you’re notified about. In Amazon SNS, the topic describes what kind of notification you receive. This section provides high-level steps for creating a topic and rule for events from AWS Elemental MediaPackage. For detailed information about topics and rules, see the following:

- Create a topic and Subscribe to a topic in the Amazon Simple Notification Service Developer Guide
- Getting started with Amazon CloudWatch Events in the Amazon CloudWatch Events User Guide

To create notifications of CloudWatch events

1. Access Amazon SNS and create a topic. Give the topic a descriptive name that you will later recognize.
2. Subscribe to the topic that you just created. Choose what kind of notification you want to receive, and where that notification is sent. For example, for email notifications, choose the Email protocol and enter the email address to receive notifications for the endpoint.
3. Access CloudWatch Events and create a rule that uses a Custom event pattern. In the pattern preview space, enter the following:

```json
{
    "source": [
        "aws.mediapackage"
    ],
    "detail-type": [
        "detail-type from event"
    ]
}
```

For detail-type, enter the value for the detail-type field from the event. You can use the following values for detail-type:

- MediaPackage Input Notification
- MediaPackage Key Provider Notification
For information about the event types, see AWS Elemental MediaPackage events (p. 149).

**Example**

The following example rule creates notifications for all events on all detail-types.

```json
{
    "source": [
        "aws.mediapackage"
    ],
    "detail-type": [
        "MediaPackage Input Notification",
        "MediaPackage Key Provider Notification",
        "MediaPackage HarvestJob Notification"
    ]
}
```

4. Add a target to the rule that you just created. Choose **SNS topic**, and then choose the topic that you created in step 1.

5. Configure the details of the rule, and give it a descriptive name. To start using the rule, make sure it's enabled, and then save it.

---

**Logging AWS Elemental MediaPackage API calls with AWS CloudTrail**

Logging is available with only live workflows in AWS Elemental MediaPackage.

AWS Elemental MediaPackage is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in MediaPackage. CloudTrail captures all API calls for MediaPackage as events. These include calls from the MediaPackage console and code calls to the MediaPackage API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for MediaPackage. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to MediaPackage, the IP address from which the request was made, who made the request, when it was made, and additional details.

To learn more about CloudTrail, see the AWS CloudTrail User Guide.

**AWS Elemental MediaPackage information in CloudTrail**

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in AWS Elemental MediaPackage, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing events with CloudTrail event history.

For an ongoing record of events in your AWS account, including events for MediaPackage, create a trail. A **trail** enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:
Understanding AWS Elemental MediaPackage log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following example shows a CloudTrail log entry that demonstrates the UpdateChannel operation:

```json
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "ABCDEFGHIJKL123456789",
    "arn": "arn:aws:sts::444455556666:assumed-role/Admin/testUser",
    "accountId": "444455556666",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2018-12-18T00:50:58Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "ABCDEFGHIJKL123456789",
        "arn": "arn:aws:iam::444455556666:role/Admin",
        "accountId": "444455556666",
        "userName": "Admin"
      }
    }
  },
  "eventTime": "2018-12-18T00:50:59Z",
  "eventSource": "mediapackage.amazonaws.com",
  "eventName": "UpdateChannel",
  ...
}```
Access logging

MediaPackage provides access logs that capture detailed information about requests sent to your MediaPackage channel or packaging group. MediaPackage generates **ingress access logs** for requests sent to the channel's input endpoints, and **egress access logs** for requests sent to your channel's endpoints or packaging group's assets. Each log contains information such as the time the request was received, the client's IP address, latencies, request paths, and server responses. You can use these access logs to analyze service performance and troubleshoot issues. They can also help you learn about your customer base and understand your MediaPackage bill.

Access logging is an optional feature of MediaPackage that is disabled by default. After you enable access logging, MediaPackage captures the logs and saves them to the CloudWatch log group that you specify when you create or manage access logging. Typical CloudWatch Logs charges apply.

**Topics**
- Permissions to publish access logs to CloudWatch (p. 158)
- Enable access logging (p. 158)
- Disable access logging (p. 159)
- Access log format (p. 159)
- Read the access logs (p. 161)
Permissions to publish access logs to CloudWatch

When you enable access logging, MediaPackage creates an IAM service-linked role, AWSServiceRoleForMediaPackage, in your AWS account. This role allows MediaPackage to publish access logs to CloudWatch. For information about how MediaPackage uses service-linked roles, see Using Service-Linked Roles for MediaPackage (p. 135).

Enable access logging

You can enable access logs using the AWS Management Console or the AWS CLI.

To enable access logs for an existing channel using the console

1. Open the MediaPackage console at https://console.aws.amazon.com/mediapackage/.
2. Select your channel.
3. In the Configure Access Logs section, do the following:
   a. Choose Enable ingress access logs or Enable egress access logs, or both.
   b. You can specify a custom CloudWatch Log group name. If left blank, the default group is used.

To enable access logs for an existing packaging group using the console

1. Open the MediaPackage console at https://console.aws.amazon.com/mediapackage/.
2. Select Packaging groups from the navigation section.
3. Choose your packaging group.
   a. Select Edit in the navigation bar.
   b. In the Access logging section, select Enable egress access logs.
   c. You can specify a custom CloudWatch Log group name. If left blank, the default group is used.
4. Choose Save changes.

To enable access logs for a channel using the AWS CLI

Use the configure-logs command with the --ingress-access-logs parameter, --egress-access-logs parameter, or both, to enable access logging. You can include a CloudWatch log group name for the --ingress-access-logs and --egress-access-logs parameters. If you don't specify a log group name, then the MediaPackage default log group is used. For ingress logs, the default log group is /aws/MediaPackage/IngressAccessLogs, and for egress logs the default log group is /aws/MediaPackage/EgressAccessLogs.

Use the following command to enable both ingress and access logs using the default log groups:

```
aws mediapackage configure-logs --id channel-name --ingress-access-logs {} --egress-access-logs {}  
```

This command has no return value.

To enable access logs for a packaging group using the AWS CLI

Use the configure-logs command with the --egress-access-logs parameter to enable access logging. You can include a CloudWatch log group name for the --egress-access-logs parameter. If you don't specify a log group name, then the MediaPackage default log group is used. For ingress logs, the default log group is /aws/MediaPackage/IngressAccessLogs, and for egress logs the default log group is /aws/MediaPackage/EgressAccessLogs.
Use the following command to enable egress access logs using the default log groups:

```
aws mediapackage configure-logs --id package-name --egress-access-logs {}
```

This command has no return value.

**Disable access logging**

You can disable access logs for your MediaPackage channel or packaging group at any time.

**To disable access logging using the console**

1. Open the MediaPackage console at https://console.aws.amazon.com/mediapackage/.
   Select your channel or package group.
2. Choose **Edit**.
3. In the **Access logging** section, deselect **Ingress access logging**, **Egress access logging**, or both.
4. Choose **Save changes**.

**To disable access logging for a channel using the AWS CLI**

Use the `configure-logs` command to disable access logging. If one or more of the access log parameters aren't declared with the `configure-logs` command, then the corresponding access logs are disabled. For example, in the following command egress access logs are enabled for a channel, and ingress access logs are disabled:

```
aws mediapackage configure-logs --id channel-name --egress-access-logs {}
```

This command has no return value.

**To disable access logging for a packaging group using the AWS CLI**

Use the `configure-logs` command to disable access logging. If one or more of the access log parameters aren't declared with the `configure-logs` command, then the corresponding access logs are disabled. For example, in the following command `configure-logs` doesn't include `--egress-access-logs` so egress logs are disabled:

```
aws mediapackage configure-logs --id package-group-name
```

This command has no return value.

**Access log format**

The access log files consist of a sequence of JSON-formatted log records, where each log record represents one request. The order of the fields within the log can vary. The following is an example channel egress access log:

```json
{
  "timestamp": "2020-07-13T18:59:56.293656Z",
  "clientIp": "192.0.2.0/24",
  "processingTime": 0.445,
  "statusCode": "200",
  "receivedBytes": 468,
  "sentBytes": 2587370,
}
```
"method": "GET",
"request": "https://aaabbbcccdddee.mediapackage.us-east-1.amazonaws.com:443/out/v1/75ee4f20e6df43e5821e5cb17ea19238/hls_7_145095.ts?m=1538005779",
"protocol": "HTTP/1.1",
"userAgent": "sabr/3.0 Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US) AppleWebKit/528.18 (KHTML, like Gecko) Version/4.0 Safari/528.17",
"account": "111122223333",
"channelId": "my_channel",
"channelArn": "arn:aws:mediapackage:us-west-2:111122223333:channels/ExampleChannelID",
"domainName": "aaabbbcccdddee.mediapackage.us-east-1.amazonaws.com",
"requestId": "aaaAAA111bbbBBB222cccCCC333dddDDD",
"endpointId": "my_endpoint",
"endpointArn": "arn:aws:mediapackage:us-west-2:111122223333:origin_endpoints/ExampleEndpointID"
}

The following list describes the log record fields, in order:

**timestamp**

The time of day when the request was received. The value is ISO-8601 date time and is based on the system clock of the host that served the request.

**clientIp**

The IP address of the requesting client.

**processingTime**

The number of seconds that MediaPackage spent processing your request. This value is measured from the time the last byte of your request was received until the time the first byte of the response was sent.

**statusCode**

The numeric HTTP status code of the response.

**receivedBytes**

The number of bytes in the request body that the MediaPackage server receives.

**sentBytes**

The number of bytes in the response body that the MediaPackage server sends. This value often is the same as the value of the Content-Length header that's included with server responses.

**method**

The HTTP request method that was used for the request: DELETE, GET, HEAD, OPTIONS, PATCH, POST, or PUT.

**request**

The request URL.

**protocol**

The type of protocol used for the request, such as HTTP.

**userAgent**

A user-agent string that identifies the client that originated the request, enclosed in double quotes. The string consists of one or more product identifiers, product/version. If the string is longer than 8 KB, it is truncated.

**account**

The AWS account ID of the account that was used to make the request.
channelId

The ID of the channel that received the request.

.channelArn

The Amazon Resource Name (ARN) of the channel that received the request.

domainName

The server name indication domain provided by the client during the TLS handshake, enclosed in double quotes. This value is set to - if the client doesn't support SNI or the domain doesn't match a certificate and the default certificate is presented to the client.

requestId

A string that is generated by MediaPackage to uniquely identify each request.

.endpointId

The ID of the endpoint that received the request.

.endpointArn

The Amazon Resource Name (ARN) of the endpoint that received the request.

The order of the fields in the log can vary.

**Read the access logs**

MediaPackage writes the access logs to Amazon CloudWatch Logs. Typical CloudWatch Logs charges apply. Use CloudWatch Logs Insights to read the access logs. For information on how to use CloudWatch Logs Insights, see Analyzing Log Data with CloudWatch Logs Insights in the AWS CloudWatch Logs User Guide.

**Note**

The access logs can take a few minutes to appear in CloudWatch. If you don't see the logs, wait a few minutes and try again.

**Examples**

This section includes example queries that you can use to read MediaPackage debug log data.

**Example View the HTTP status code responses for a channel.**

Use this query to view the responses by HTTP status code for a channel. You can use this to view HTTP error code responses to help you to troubleshoot issues.

```
fields @timestamp, @message
| filter channelId like 'my-channel'
| stats count() by statusCode
```

**Example Get the number of requests per endpoint on a channel.**

```
fields @timestamp, @message
| filter channelId like 'my-channel'
| stats count() by endpointId
```

**Example View status codes per asset.**

```
fields @timestamp, @message
```
| filter assetArn like 'my-asset-id'  
| stats count() by statusCode

**Example Get the P99 response times for a packaging configuration over time**

```
fields @timestamp, @message  
| filter packagingConfigArn like 'my-dash-config'  
| stats pct(processingTime, 99) by bin(5m)
```
Tagging AWS Elemental MediaPackage resources

A tag is a label that you assign to an AWS resource. Each tag consists of a key and a value, both of which you define. For example, the key might be "stage" and the value might be "test". You can use tags for a variety of purposes. One common use is to control access to AWS resources using tags. For information, see the Controlling access to AWS resources using tags topic in the IAM User Guide.

Another common use of tags is to categorize and track your MediaPackage costs. When you apply cost allocation tags to MediaPackage channels, endpoints, and packaging configurations, AWS generates a cost allocation report as a comma-separated value (CSV) file with your usage and costs aggregated by your tags. You can apply tags that represent business categories (such as cost centers, application names, or owners) to organize your costs across multiple services. For more information about using tags for cost allocation, see Using cost allocation tags in the AWS Billing and Cost Management User Guide.

Tag restrictions

The following restrictions apply to tagging AWS Elemental MediaPackage resources:

- Cost allocation tagging is only available for channel, endpoint, and packaging configuration resources. You can't use cost allocation tags for asset or packaging group resources.
- Maximum number of tags that you can assign to a resource – 50.
- Maximum key length – 128 Unicode characters.
- Maximum value length – 256 Unicode characters.
- Valid characters for key and value – a-z, A-Z, 0-9, space, and the following characters: _ . : / = + - and @.
- Keys and values are case sensitive.
- Don't use aws: as a prefix for keys; it's reserved for AWS use.
- Can't be used for harvested live-to-VOD assets.

Managing tags

You can use the AWS Elemental MediaPackage API or the AWS CLI to add, edit, or delete the values for these properties.

For more information, see the actions related to tags in the following reference documentation:

- Tags resource-arn in the AWS Elemental MediaPackage live API reference.
- Tags resource-arn in the AWS Elemental MediaPackage VOD API reference.
- tag-resource in the AWS CLI MediaPackage reference.
Working with Content Delivery Networks (CDNs)

You can use a content delivery network (CDN) such as Amazon CloudFront to serve the content that you store in AWS Elemental MediaPackage. A CDN is a globally distributed set of servers that caches content such as videos. When a user requests your content, the CDN routes the request to the edge location that provides the lowest latency. If your content is already cached in that edge location, the CDN delivers it immediately. If your content is not currently in that edge location, the CDN retrieves it from your origin (in this case, the MediaPackage endpoint) and distributes it to the user. The following illustration shows this process.

The following sections provide procedures for working with distributions from Amazon CloudFront.

**Topics**

- Creating a Distribution (p. 165)
- Viewing a Distribution (p. 166)
- Editing a Distribution (p. 166)
Creating a Distribution

A distribution in Amazon CloudFront holds all information about content delivery, including where content is coming from and how it's tracked and managed. The distribution holds origins (where content is originating from) and behaviors (where content requests are routed based on specified patterns in the request).

You can create a distribution automatically from the AWS Elemental MediaPackage live console, or manually from the CloudFront console. The following sections describe each approach.

Topics

- Creating a Distribution from AWS Elemental MediaPackage (p. 165)
- Creating a Distribution from Amazon CloudFront (p. 166)

Creating a Distribution from AWS Elemental MediaPackage

You can create a CloudFront distribution from the AWS Elemental MediaPackage console when you're working with live content only.

AWS Elemental MediaPackage communicates with Amazon CloudFront on your behalf to create a distribution for a channel and its endpoints. When you enable the feature, MediaPackage creates a distribution in CloudFront when you save the channel, and then adds an origin and updates cache behaviors when you save an endpoint. Because the creation process is automated and initiated from your actions in MediaPackage, there is no additional action required from you.

Important

You need additional permissions to create distributions in CloudFront. Have an admin user add the correct level of permissions through AWS Identity and Access Management (IAM) using the steps described in Step 1: (Optional) create a policy for Amazon CloudFront (p. 18).

To create a distribution from the AWS Elemental MediaPackage console

1. Start a new channel as described in the section called “Creating a channel” (p. 40), and then choose Create a CloudFront distribution.

You can also edit an existing channel to add a distribution. For instructions on editing a channel, see Editing a channel (p. 42).

2. When you've completed the channel, choose Save. AWS Elemental MediaPackage communicates with CloudFront to create a distribution. CloudFront uses placeholder values for settings that require information from the endpoint in MediaPackage, such as the origin domain name and caching behaviors.

If you receive an error message that CloudFront couldn't create the distribution, choose Edit on the channel and save it again to restart the creation process.

Note that when the distribution is first created, it isn't fully functional until it has an origin, which AWS Elemental MediaPackage creates in the next step (when you create an endpoint in MediaPackage).

3. Create an endpoint on the channel, as described in Creating an endpoint (p. 44). AWS Elemental MediaPackage updates the origin and cache behaviors with information from the endpoint, and
configures the distribution with settings that optimize live video streaming, as described in Serving Live Video Formatted with AWS Elemental MediaPackage of the Amazon CloudFront Developer Guide.

If you're adding a distribution to an existing channel with endpoints, AWS Elemental MediaPackage automatically adds the origin for you.

AWS Elemental MediaPackage communicates with CloudFront to add an origin to the distribution, and to update the settings on the distribution.

When the distribution status on the channel's details page says Deployed, you can start using the distribution. From the details page, note the CloudFront CDN URL and provide it for downstream devices to send playback requests.

Note AWS Elemental MediaPackage adds only one origin to the distribution. All endpoints on the channel are served by the same origin on the distribution.

Creating a Distribution from Amazon CloudFront

After you create a channel and its endpoints in AWS Elemental MediaPackage, note the URLs for each of the endpoints. These URLs are what you use for the origin domain names for your CloudFront distribution. You need one origin for each endpoint on the channel in MediaPackage.

For detailed steps about creating a distribution in Amazon CloudFront with AWS Elemental MediaPackage endpoints as the origins, see Delivering Live Streaming Video in the Amazon CloudFront Developer Guide.

Viewing a Distribution

As described in Viewing channel details (p. 41), you can view basic information about a distribution that was created in MediaPackage, such as the distribution ID and description. Note that the ID links to the CloudFront management console.

Access more detailed information about the distribution from the Amazon CloudFront console. For help accessing this information, see Viewing and Updating Distribution in the Amazon CloudFront Developer Guide.

Editing a Distribution

Edit an Amazon CloudFront distribution from the CloudFront console.

The only edit that AWS Elemental MediaPackage can make to an origin is to create an origin when you add an endpoint to a channel in MediaPackage. You can't edit a distribution from the MediaPackage console.

To access the distribution in CloudFront, choose the distribution's ID on the channel's details page. For more information about editing a distribution in CloudFront, see Viewing and Updating Distribution in the Amazon CloudFront Developer Guide.

Important When you're editing a distribution, do not change the default on the Tagging page. CloudFront uses the AWS Elemental MediaPackage channel ID in this tag to link the distribution and the channel together. If the tag is modified, then you will no longer be able to view or manage the distribution from MediaPackage.
Deleting a Distribution

Delete an Amazon CloudFront distribution from the CloudFront console. You can't delete a distribution from the AWS Elemental MediaPackage console.

To access the distribution in CloudFront, choose the distribution's ID on the channel's details page. For more information about deleting a distribution in CloudFront, see Deleting a Distribution in the Amazon CloudFront Developer Guide.
Quotas in AWS Elemental MediaPackage

The following sections provide information about the quotas in AWS Elemental MediaPackage.

**Topics**
- Live content quotas (p. 168)
- VOD content quotas (p. 169)

**Live content quotas**

This section describes the quotas for live content in AWS Elemental MediaPackage. For information about requesting an increase to soft quotas, see AWS service quotas. Hard quotas can't be changed.

**Live soft quotas**

The following table describes quotas in AWS Elemental MediaPackage for live content that can be increased. For information about changing quotas, see AWS Service Quotas.

For some customers, your account quota might be below these published quotas. If you believe that you encountered a Resource limit exceeded error wrongfully, use the Service Quotas console to request quota increases.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Default quota</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Channels</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td>Increasing your channel quota doesn't always mean that you also need to increase your endpoints. For example, if you need 34 channels and want to serve HLS, HLS encrypted, and DASH content from each channel, you need only 3 endpoints for each channel (one for each output type). The default endpoint quota is 10 so, although you do need a channel quota increase, you don't need to increase your endpoint quota. You won't exceed the quota of 10 endpoints <em>per channel</em>.</td>
</tr>
<tr>
<td>Maximum Endpoints per Channel</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td>This is a <em>per channel</em> quota. Each endpoint represents the output package that you use. If one channel serves HLS, HLS encrypted, DASH, DASH encrypted, Microsoft Smooth, and Microsoft</td>
</tr>
</tbody>
</table>
Live hard quotas

The following table describes quotas in AWS Elemental MediaPackage for live content that can't be increased.

<table>
<thead>
<tr>
<th>Resource or operation</th>
<th>Quota</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Stream Quotas</td>
<td>30 streams per channel, and 10 tracks per stream</td>
</tr>
<tr>
<td>Maximum Content Age for Time-shifted Viewing</td>
<td>336 hours (14 days)</td>
</tr>
<tr>
<td>Maximum Time-shifted Manifest Length</td>
<td>24 hours for all supported output formats: Apple HTTP Live Streaming (HLS), DASH-ISO, Common Media Application Format (CMAF), and Microsoft Smooth Streaming (MSS).</td>
</tr>
<tr>
<td>Maximum Live-to-VOD Manifest Length</td>
<td>24 hours for all supported output formats: Apple HTTP Live Streaming (HLS), DASH-ISO, Common Media Application Format (CMAF), and Microsoft Smooth Streaming (MSS).</td>
</tr>
<tr>
<td>Request Rates per Channel</td>
<td>• Input: 50 requests per second</td>
</tr>
<tr>
<td></td>
<td>• Output: 200 requests per second</td>
</tr>
<tr>
<td>REST API Requests</td>
<td>• Steady state: 5 requests per second</td>
</tr>
<tr>
<td></td>
<td>• Bursting: 50 requests per second</td>
</tr>
</tbody>
</table>

VOD content quotas

This section describes the quotas for video on demand (VOD) content in AWS Elemental MediaPackage. For information about requesting an increase to soft quotas, see AWS Service Quotas. Hard quotas can't be changed.

VOD soft quotas

The following table describes quotas in AWS Elemental MediaPackage for VOD content that can be increased. For information about changing quotas, see AWS Service Quotas.

For some customers, your account quota might be below these published quotas. If you believe that you encountered a Resource limit exceeded error wrongfully, use the Service Quotas console to request quota increases.
VOD hard quotas

The following table describes quotas within AWS Elemental MediaPackage for VOD content that can't be increased.

<table>
<thead>
<tr>
<th>Resource or operation</th>
<th>Quota</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Stream Quotas</td>
<td>30 streams per asset, and 10 tracks per stream</td>
</tr>
<tr>
<td>Request Rates per Asset</td>
<td>200 output requests per second</td>
</tr>
</tbody>
</table>
| REST API Requests           | • Steady state: 5 requests per second  
                                • Bursting: 50 requests per second        |

**Resource**

<table>
<thead>
<tr>
<th>Default quota</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Packaging Groups</td>
</tr>
<tr>
<td>Maximum Packaging Configurations per Packaging Group</td>
</tr>
<tr>
<td>Maximum Assets per Packaging Group</td>
</tr>
</tbody>
</table>

**Note**

Increasing your packaging group quota doesn't always mean that you also need to increase your assets or packaging configurations. For example, if you need 14 groups and want to serve HLS, HLS encrypted, and DASH content from each asset, you need only 3 packaging configurations for each asset (one for each output type). You do need to increase your packaging group quota, but not the packaging configuration quota because you have fewer than 10 configurations per packaging group.

This is a per packaging group quota. Each packaging configuration represents the output package that you use. If one packaging group has configurations for HLS, HLS encrypted, DASH, DASH encrypted, Microsoft Smooth, and Microsoft Smooth encrypted content, then that group has 6 packaging configurations and falls within the 10 configurations quota. If you have 10 packaging groups set up this same way, then you still haven't exceeded the quota because each group uses only 6 configurations.

This is a per packaging group quota. For example, if you have 10,500 assets spread out over multiple packaging groups, then you still haven't exceeded the quota if each group has no more than 10,000 assets.
### AWS Elemental MediaPackage related information

The following table lists related resources that you'll find useful as you work with AWS Elemental MediaPackage.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classes and Workshops</td>
<td>Links to role-based and specialty courses as well as self-paced labs to help sharpen your AWS skills and gain practical experience.</td>
</tr>
<tr>
<td>AWS Developer Tools</td>
<td>Links to developer tools, SDKs, IDE tool kits, and command line tools for developing and managing AWS applications.</td>
</tr>
<tr>
<td>AWS Whitepapers</td>
<td>Links to a comprehensive list of technical AWS whitepapers, covering topics such as architecture, security, and economics and authored by AWS Solutions Architects or other technical experts.</td>
</tr>
<tr>
<td>AWS Support Center</td>
<td>The hub for creating and managing your AWS Support cases. Also includes links to other helpful resources, such as forums, technical FAQs, service health status, and AWS Trusted Advisor.</td>
</tr>
<tr>
<td>AWS Support</td>
<td>The primary web page for information about AWS Support, a one-on-one, fast-response support channel to help you build and run applications in the cloud.</td>
</tr>
<tr>
<td>Contact Us</td>
<td>A central contact point for inquiries concerning AWS billing, account, events, abuse, and other issues.</td>
</tr>
<tr>
<td>AWS Site Terms</td>
<td>Detailed information about our copyright and trademark; your account, license, and site access; and other topics.</td>
</tr>
</tbody>
</table>
# Document history for User Guide

The following table describes important changes in each release of the *AWS Elemental MediaPackage User Guide* after May 2018. For notification about updates to this documentation, you can subscribe to an RSS feed.

- **API version:** 1.0

<table>
<thead>
<tr>
<th>update-history-change</th>
<th>update-history-description</th>
<th>update-history-date</th>
</tr>
</thead>
<tbody>
<tr>
<td>MediaPackage now supports SPEKE v2 (p. 93)</td>
<td>Added information about SPEKE v2.</td>
<td>September 7, 2021</td>
</tr>
<tr>
<td>New guidance for integrating with SPEKE v2 (p. 94)</td>
<td>MediaPackage now supports SPEKE v2.</td>
<td>September 6, 2021</td>
</tr>
<tr>
<td>New metadata passthrough topic (p. 111)</td>
<td>MediaPackage now supports ID3 and KLV metadata passthrough. When ID3 or KLV metadata is present in a channel's input stream, MediaPackage automatically passes through the metadata to the output stream.</td>
<td>June 30, 2021</td>
</tr>
<tr>
<td>New CMAF encryption options (p. 78)</td>
<td>MediaPackage now offers constant IV and multiple system IDs for CMAF packaging configurations.</td>
<td>June 30, 2021</td>
</tr>
<tr>
<td>Updated cost allocation tagging content. (p. 163)</td>
<td>Clarified that cost allocation tagging is currently only available for.</td>
<td>May 27, 2021</td>
</tr>
<tr>
<td>MediaPackage now supports DASH endpoints for live-to-VOD (p. 84)</td>
<td>Added DASH endpoint requirements for live-to-VOD.</td>
<td>May 14, 2021</td>
</tr>
<tr>
<td>New DASH and CMAF manifest setting (p. 72)</td>
<td>Added the <em>Include encoder configuration in segments</em> manifest settings option. If you turn on this option, MediaPackage places Sequence Parameter Set (SPS), Picture Parameter Set (PPS), and Video Parameter Set (VPS) metadata in each video segment instead of in the init fragment.</td>
<td>April 28, 2021</td>
</tr>
<tr>
<td>Added audio_bitrate query parameter. (p. 106)</td>
<td>audio_bitrate can now be used as a manifest filter query parameter.</td>
<td>March 22, 2021</td>
</tr>
<tr>
<td>Access logging (p. 157)</td>
<td>MediaPackage now supports access logging for VOD.</td>
<td>February 24, 2021</td>
</tr>
<tr>
<td>Added information about MediaPackage event handling behavior for CloudWatch events (p. 149)</td>
<td>MediaPackage emits events to CloudWatch on a best effort basis.</td>
<td>January 7, 2021</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Access logging (p. 157)</td>
<td>MediaPackage now supports access logging, which provides detailed records for the requests that are made to a channel. This feature is available for live workflows.</td>
<td>October 21, 2020</td>
</tr>
<tr>
<td>Support for SCTE-35 EXT-x-DATERANGE tag (p. 116)</td>
<td>Added a new EXT-X-DATERANGE section to the SCTE-35 Ad Marker topic.</td>
<td>August 7, 2020</td>
</tr>
<tr>
<td>Moved the maximum live manifest length from hard quota to soft quota (p. 169)</td>
<td>The maximum live manifest length is a soft quota. Moved this entry from hard quotas to soft quotas.</td>
<td>June 24, 2020</td>
</tr>
<tr>
<td>Added information about CDN authorization for VOD. (p. 89)</td>
<td>Added CDN Authorization in AWS Elemental MediaPackage topic to describe how to add authorization to requests from your CDN.</td>
<td>May 29, 2020</td>
</tr>
<tr>
<td>Updated manifest filtering topic. (p. 106)</td>
<td>Added 6 new parameters and updated the character limit to 1024.</td>
<td>May 15, 2020</td>
</tr>
<tr>
<td>Remove VOD tagging restriction. (p. 163)</td>
<td>MediaPackage now supports tagging for VOD.</td>
<td>April 23, 2020</td>
</tr>
<tr>
<td>New manifest filtering topic. (p. 106)</td>
<td>Added a new Manifest Filtering topic.</td>
<td>April 8, 2020</td>
</tr>
<tr>
<td>Updated maximum time-shifted and live-to-VOD manifest length (p. 169)</td>
<td>The maximum manifest length is now 24 hours for all supported output formats.</td>
<td>March 9, 2020</td>
</tr>
<tr>
<td>New VOD DASH-ISO manifest console settings. (p. 72)</td>
<td>Added new DASH-ISO manifest console settings for VOD packaging configurations. Compact DASH, new segment template formats, and period trigger options are now available.</td>
<td>February 25, 2020</td>
</tr>
<tr>
<td>Multi-period DASH is now available for both live and VOD. (p. 97)</td>
<td>Removed references to &quot;live only&quot; support for multi-period DASH.</td>
<td>February 25, 2020</td>
</tr>
<tr>
<td>Compact DASH manifests are now available for both VOD and live. (p. 100)</td>
<td>Removed reference to &quot;live only&quot; support for compact DASH.</td>
<td>February 25, 2020</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>Fixed bug (p. 153)</td>
<td>Successful harvest jobs CloudWatch events return &quot;status&quot;: &quot;SUCCEEDED&quot;.</td>
<td>February 10, 2020</td>
</tr>
<tr>
<td>Added information content delivery network (CDN) authorization. (p. 89)</td>
<td>Added CDN Authorization in AWS Elemental MediaPackage topic to describe how to add authorization to requests from your CDN.</td>
<td>December 23, 2019</td>
</tr>
<tr>
<td>Added information about VOD playback events. (p. 150)</td>
<td>Added example playback ready notification events for ingested VOD content.</td>
<td>November 8, 2019</td>
</tr>
<tr>
<td>Added information about live-to-VOD CloudWatch Events. (p. 153)</td>
<td>Added example harvest job notification events for live-to-VOD content harvesting.</td>
<td>October 15, 2019</td>
</tr>
<tr>
<td>Added SML manifest information. (p. 7)</td>
<td>Added Creating a SMIL File topic to describe the supported .smil manifest format for VOD ingest.</td>
<td>October 10, 2019</td>
</tr>
<tr>
<td>Added live-to-VOD (video on demand) topics. (p. 1)</td>
<td>Throughout guide, added and updated topics about creating live-to-VOD assets, including Creating Live-to-VOD Assets and Live-to-VOD Content Delivery.</td>
<td>October 1, 2019</td>
</tr>
<tr>
<td>Updated time-shifted manifest length limit. (p. 169)</td>
<td>AWS Elemental MediaPackage can now produce time-shifted manifests up to 18 hours for DASH with compact manifest, HLS, and CMAF.</td>
<td>August 21, 2019</td>
</tr>
<tr>
<td>Added supported inputs and outputs info. (p. 3)</td>
<td>Added Supported Inputs and Outputs topic that describes what input types, containers, and codecs MediaPackage supports.</td>
<td>June 21, 2019</td>
</tr>
<tr>
<td>Added configurable SCTE-35 options. (p. 114)</td>
<td>Added SCTE-35 Message Options in AWS Elemental MediaPackage topic that describes how you can configure MediaPackage behavior when there are SCTE-35 markers in your input content.</td>
<td>June 21, 2019</td>
</tr>
<tr>
<td>Added security chapter. (p. 121)</td>
<td>Added the Security chapter to enhance and standardize security topics for MediaPackage.</td>
<td>June 5, 2019</td>
</tr>
<tr>
<td>Added video on demand (VOD) topics. (p. 1)</td>
<td>Throughout guide, added topics about working with VOD content: VOD Content Processing, Allowing MediaPackage to Access Amazon Simple Storage Service, VOD Content Delivery, Delivering VOD Content, VOD Content Metrics, and VOD Content Limits.</td>
<td>May 17, 2019</td>
</tr>
<tr>
<td>Added further information about DASH manifest SegmentTemplate format options. (p. 102)</td>
<td>Added the Duration Attribute topic to discuss how to include duration information in SegmentTemplate instead of using SegmentTimeline.</td>
<td>May 10, 2019</td>
</tr>
<tr>
<td>Updated time-shifted manifest length limit. (p. 169)</td>
<td>AWS Elemental MediaPackage can now produce time-shifted manifests up to 9 hours.</td>
<td>May 1, 2019</td>
</tr>
<tr>
<td>Added information about live and VOD manifests. (p. 12)</td>
<td>Added the Live and VOD Manifest Reference topic that explains when MediaPackage serves a live or VOD manifest.</td>
<td>April 16, 2019</td>
</tr>
<tr>
<td>Added tagging information. (p. 163)</td>
<td>Added Tagging Resources topic to discuss how tagging channels and endpoints works in AWS Elemental MediaPackage.</td>
<td>March 4, 2019</td>
</tr>
<tr>
<td>Added information about DASH manifest SegmentTemplate format options. (p. 102)</td>
<td>Added the DASH Manifest Segment Template Format topic to discuss how to change variables in the media URL in the SegmentTemplate object of the DASH manifest.</td>
<td>February 6, 2019</td>
</tr>
<tr>
<td>Added DASH manifest treatment information. (p. 96)</td>
<td>Added DASH Manifest Options topic to discuss the ways that you can modify output DASH manifests.</td>
<td>February 6, 2019</td>
</tr>
<tr>
<td>Added AWS CloudTrail logging information. (p. 155)</td>
<td>Added Logging AWS Elemental MediaPackage API Calls with AWS CloudTrail topic to discuss using CloudTrail to log actions in the AWS Elemental MediaPackage API.</td>
<td>December 21, 2018</td>
</tr>
<tr>
<td>Added information about compact DASH manifests. (p. 100)</td>
<td>Added a Compacted DASH Manifests topic to discuss how compacting DASH output manifests works in AWS Elemental MediaPackage.</td>
<td>December 18, 2018</td>
</tr>
<tr>
<td>Updated content retention window limit. (p. 169)</td>
<td>AWS Elemental MediaPackage now retains content for 336 hours (14 days).</td>
<td>November 13, 2018</td>
</tr>
</tbody>
</table>
### Earlier updates

The following table describes important changes in each release of the *AWS Elemental MediaPackage User Guide* before May 2018.

<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Added content key encryption to DRM encryption</td>
<td>Added the option to encrypt content keys. Prior to this, AWS Elemental MediaPackage supported clear key delivery only. To use content key encryption, your DRM key provider must support encrypted content keys. If you enable this feature for a key provider that doesn't handle content key encryption, the operation fails.</td>
<td>November 8, 2018</td>
</tr>
<tr>
<td>Added input redundancy information.</td>
<td>Added <em>How Input Redundancy Works</em> topic to discuss how MediaPackage can receive two identical streams for back-up purposes.</td>
<td>August 28, 2018</td>
</tr>
<tr>
<td>Added Amazon CloudFront console integration</td>
<td>Added sections about working with distributions in CloudFront, including how to create a distribution from the AWS Elemental MediaPackage console.</td>
<td>August 3, 2018</td>
</tr>
<tr>
<td>Added information about multi-period DASH</td>
<td>Added <em>Multi-period DASH in AWS Elemental MediaPackage</em> topic to discuss the purpose and functionality of multiple periods in DASH manifests.</td>
<td>July 18, 2018</td>
</tr>
<tr>
<td>Added content delivery network (CDN) information.</td>
<td>Added <em>Working with CDNs</em> topic to discuss how AWS Elemental MediaPackage works with CDNs such as Amazon CloudFront.</td>
<td>May 31, 2018</td>
</tr>
<tr>
<td>Added information about creating event notifications</td>
<td>Added the <em>Creating Event Notifications</em> topic that describes how to use Amazon CloudWatch Events and Amazon Simple Notification Service to notify you of new events.</td>
<td>January 22, 2018</td>
</tr>
</tbody>
</table>
### Earlier updates

<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>In Working with Endpoints, added reference to access control fields.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Added IAM policy information specific to AWS Elemental MediaPackage.</td>
<td>In Setting up AWS Elemental MediaPackage (p. 16), added instructions for creating non-admin roles with limited permissions.</td>
<td>December 13, 2017</td>
</tr>
<tr>
<td>Added hard limit information.</td>
<td>In Quotas in AWS Elemental MediaPackage (p. 168), added information about limits that can't be changed (hard limits).</td>
<td>December 20, 2017</td>
</tr>
<tr>
<td>Updated IAM policy information.</td>
<td>In Setting up AWS Elemental MediaPackage (p. 16), added information about policies specific to AWS Elemental MediaPackage.</td>
<td>January 5, 2018</td>
</tr>
<tr>
<td>Added CMAF endpoint information.</td>
<td>Added Creating a common media application format (CMAF) endpoint (p. 52) section for new output type.</td>
<td>April 6, 2018</td>
</tr>
<tr>
<td>Updated feature functionality.</td>
<td>In Features of AWS Elemental MediaPackage (p. 13), added feature support for HDR-10.</td>
<td>April 30, 2018</td>
</tr>
<tr>
<td>Added content delivery network (CDN) information.</td>
<td>Added topic Working with Content Delivery Networks (CDNs) (p. 164) to discuss how AWS Elemental MediaPackage works with CDNs such as Amazon CloudFront.</td>
<td>May 31, 2018</td>
</tr>
</tbody>
</table>

**Note**

- The AWS Media Services are not designed or intended for use with applications or in situations requiring fail-safe performance, such as life safety operations, navigation or communication systems, air traffic control, or life support machines in which the unavailability, interruption or failure of the services could lead to death, personal injury, property damage or environmental damage.
AWS glossary

For the latest AWS terminology, see the AWS glossary in the AWS General Reference.