# Table of Contents

What Is AWS Elemental MediaPackage? ................................................................. 1
Are You a First-Time User of AWS Elemental MediaPackage? ........................................ 1
Concepts and Terminology ...................................................................................... 1
  Live Components ............................................................................................... 1
  VOD Components ............................................................................................ 2
How AWS Elemental MediaPackage Works ......................................................... 3
  Live Content Processing .................................................................................... 3
  VOD Content Processing .................................................................................. 6
  Live and VOD Manifest Reference .................................................................. 6
Features of AWS Elemental MediaPackage ......................................................... 7
Related Services .................................................................................................... 8
Accessing AWS Elemental MediaPackage .......................................................... 9
Pricing for AWS Elemental MediaPackage .......................................................... 9
Regions for AWS Elemental MediaPackage ......................................................... 9
Setting Up ............................................................................................................ 10
  Signing Up for AWS .......................................................................................... 10
  Creating an Admin IAM User .......................................................................... 10
  Creating a Non-Admin IAM User ...................................................................... 11
    Creating a Policy for Amazon CloudFront ...................................................... 12
Allowing AWS Elemental MediaPackage to Access Amazon S3 ......................... 13
  Step 1: Create a Policy .................................................................................... 13
  Step 2: Create a Role ....................................................................................... 14
  Step 3: Modify the Trust Relationship ............................................................... 16
(Optional) Setting Up Encryption ........................................................................ 17
Getting Started .................................................................................................... 18
  Live Content Delivery ...................................................................................... 18
    Prerequisites ................................................................................................. 18
    Step 1: Access AWS Elemental MediaPackage ............................................. 18
    Step 2: Create a Channel .............................................................................. 18
    Step 3: Create Endpoints .............................................................................. 19
    (Optional) Step 4: Monitor AWS Elemental MediaPackage Activity ............ 19
    Step 5: Clean Up ............................................................................................ 20
  VOD Content Delivery ...................................................................................... 20
    Prerequisites ................................................................................................. 21
    Step 1: Access AWS Elemental MediaPackage ............................................. 21
    Step 2: Create a Packaging Group ................................................................. 21
    Step 3: Create a Packaging Configuration .................................................... 21
    Step 4: Create an Asset ................................................................................ 22
    Step 5: Provide Playback URLs ..................................................................... 22
    (Optional) Step 6: Monitor AWS Elemental MediaPackage Activity ............ 23
    Step 7: Clean Up ............................................................................................ 23
Delivering Live Content ....................................................................................... 24
  Working with Channels ..................................................................................... 24
    Creating a Channel ....................................................................................... 24
    Viewing Channel Details ............................................................................. 25
    Editing a Channel ......................................................................................... 26
    Rotating Credentials on an Input URL .......................................................... 26
    Deleting a Channel ....................................................................................... 27
    Adding an Endpoint to a Channel .................................................................. 27
  Working with Endpoints ................................................................................... 28
    Creating an Endpoint .................................................................................... 28
    Viewing All Endpoints Associated with a Channel ....................................... 43
    Viewing a Single Endpoint .......................................................................... 43
    Editing an Endpoint ...................................................................................... 43

AWS Elemental MediaPackage User Guide
What Is AWS Elemental MediaPackage?

AWS Elemental 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).

AWS Elemental 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 AWS Elemental MediaPackage? (p. 1)
• Concepts and Terminology (p. 1)
• How AWS Elemental MediaPackage Works (p. 3)
• Features of AWS Elemental MediaPackage (p. 7)
• Related Services (p. 8)
• Accessing AWS Elemental MediaPackage (p. 9)
• Pricing for AWS Elemental MediaPackage (p. 9)
• Regions for AWS Elemental MediaPackage (p. 9)

Are You a First-Time User of AWS Elemental MediaPackage?

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

• How AWS Elemental MediaPackage Works (p. 3)
• Concepts and Terminology (p. 1)
• Features of AWS Elemental MediaPackage (p. 7)
• Getting Started with AWS Elemental MediaPackage (p. 18)

Concepts and Terminology

AWS Elemental MediaPackage includes the following components:

**Just-in-time packaging**

AWS Elemental 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**

AWS Elemental 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 AWS Elemental 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.

Stream

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

Track

Tracks make up the output content stream. AWS Elemental 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 AWS Elemental MediaPackage:

Channel

A channel represents the entry point for a content stream into AWS Elemental 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 AWS Elemental 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.

VOD Components

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

Asset

An asset represents the entry point for file-based content into AWS Elemental 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,
How AWS Elemental MediaPackage Works

AWS Elemental 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 AWS Elemental MediaPackage works.

Topics
- Live Content Processing (p. 3)
- VOD Content Processing (p. 6)
- Live and VOD Manifest Reference (p. 6)

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. 3)
- Live Input Redundancy AWS Elemental MediaPackage Processing Flow (p. 4)

General AWS Elemental MediaPackage Live Processing Flow

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

1. An upstream encoder (such as AWS Elemental MediaLive) sends an HLS live stream with digest authentication over WebDAV to the AWS Elemental 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. 4).

2. A downstream device requests content from AWS Elemental MediaPackage through the endpoint output URL. A downstream device is either a video player or a content distribution 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 AWS Elemental 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. AWS Elemental 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. AWS Elemental 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. 79).

Throughout the content input and output processes, AWS Elemental 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 AWS Elemental 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 AWS Elemental MediaPackage with AWS Elemental MediaLive (for example), here is the flow of input redundancy:

1. You create a channel in AWS Elemental MediaPackage, as described in Creating a Channel (p. 24). 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, AWS Elemental 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 AWS Elemental MediaPackage as described in Creating an Endpoint (p. 28).

   **Important**
   If you use short output segments, depending on your playback device, you might see buffering when AWS Elemental 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. 28).

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 AWS Elemental 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 AWS Elemental MediaPackage.

5. AWS Elemental 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.

The following section describes the VOD processing flow.

**Topics**
- General VOD Processing Flow (p. 6)

**General VOD Processing Flow**

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

1. From the AWS Elemental 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 AWS Elemental MediaPackage through the packaging configuration URL on the asset. A downstream device is either a video player or a content distribution 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 AWS Elemental 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. AWS Elemental 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. AWS Elemental 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. 79).

Throughout the content input and output processes, AWS Elemental 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

AWS Elemental 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. AWS Elemental 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. 64).

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

When you stop the output from your encoder, the manifest that it sends to AWS Elemental 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 AWS Elemental MediaPackage are stopped, MediaLive doesn’t include EXT-X-ENDLIST in the HLS manifest to MediaPackage. MediaPackage continues to produce a live manifest. If both pipelines are active when you stop the channel, MediaLive includes EXT-X-ENDLIST. MediaPackage delivers a VOD manifest to downstream devices.

If you restart the output from the encoder, the manifest from MediaPackage becomes live again. Playback devices might need to refresh to resume content playback.

If you're using input redundancy and the active stream ends, MediaPackage fails over to the other incoming stream for input. The manifest isn't marked as complete unless both incoming streams end.

Features of AWS Elemental MediaPackage

AWS Elemental MediaPackage supports the following features:

Audio

AWS Elemental 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+)

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

AWS Elemental MediaPackage supports input 608/708 captions and passes them through to the output stream.

DRM

AWS Elemental MediaPackage supports content protection through digital rights management (DRM).

Input Redundancy

Input redundancy is available with only live workflows in AWS Elemental MediaPackage.

AWS Elemental 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. 4).

Subtitles

AWS Elemental 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 AWS Elemental MediaPackage.

AWS Elemental 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. 64).

Video

AWS Elemental 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.

Whitelisting

Whitelisting is available with only live workflows in AWS Elemental MediaPackage.

AWS Elemental 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. 32).

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 AWS Elemental MediaPackage for packaging. For more information about how encoders (such as MediaLive) work with MediaPackage, see How AWS Elemental MediaPackage Works (p. 3).

- **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. 10).

- **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.

## Accessing AWS Elemental MediaPackage

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

- **Console access:**
  
  https://<region>.console.aws.amazon.com/mediapackage/home

- **AWS CLI endpoint:**
  
  aws mediapackage

- **AWS Elemental MediaPackage REST API endpoint:**
  
  https://config.mediapackage.<region>.amazonaws.com

## Pricing for AWS Elemental MediaPackage

As with other AWS products, there are no contracts or minimum commitments for using AWS Elemental 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 AWS Elemental 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 AWS Elemental MediaPackage Pricing.

## Regions for AWS Elemental MediaPackage

To reduce latency in your applications, AWS Elemental MediaPackage offers a regional endpoint for your requests. To view the list of AWS Regions where MediaPackage is available, see AWS Elemental 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. 10)
• Creating an Admin IAM User (p. 10)
• Creating a Non-Admin IAM User (p. 11)
• Allowing AWS Elemental MediaPackage to Access Amazon S3 (p. 13)
• (Optional) Setting Up Encryption (p. 17)

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. 11).

To create an administrator user for yourself and add the user to an administrators group (console)
1. Use your AWS account email address and password to sign in as the AWS account root user to the IAM console at https://console.aws.amazon.com/iam/.
Note
We strongly recommend that you adhere to the best practice of using the Administrator IAM user below 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. 11).

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.

To create users who can access AWS Elemental 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 AWS Elemental 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 AWS Elemental MediaPackage permissions level:
   - Use `AWSElementalMediaPackageFullAccess` to allow the user to perform all actions on all resources in AWS Elemental MediaPackage.
   - Use `AWSElementalMediaPackageReadOnly` to provide the user read-only rights for all resources in AWS Elemental MediaPackage.
7. Add policies to allow the AWS Elemental 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 AWS Elemental MediaPackage to communicate with CloudWatch, and also provide the user read-only access to all AWS services on your account.
   - Use `CloudWatchReadOnlyAccess`, `CloudWatchEventsReadOnlyAccess`, and `CloudWatchLogsReadOnlyAccess` to allow AWS Elemental 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 AWS Elemental MediaPackage console, create and attach a policy that provides required permissions for the user. The policy looks like this:

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

For help creating the policy, see [Creating a Policy for Amazon CloudFront](p. 12).
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**.

## Creating a Policy for Amazon CloudFront

If you or your users will create Amazon CloudFront distributions from the AWS Elemental MediaPackage console, you must attach 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. 96).

### 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/](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",
            "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.

### Allowing AWS Elemental MediaPackage to Access Amazon S3

To allow MediaPackage to access Amazon S3 and ingest source content from a bucket, create an IAM role and policy with the appropriate permissions.

**Topics**
- Step 1: Create a Policy (p. 13)
- Step 2: Create a Role (p. 14)
- Step 3: Modify the Trust Relationship (p. 16)

### Step 1: Create a Policy

The IAM policy defines the permissions that AWS Elemental MediaPackage requires to access Amazon S3. 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.
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"
      ],
      "Resource": [
        "arn:aws:s3:::{bucket_name}/*",
        "arn:aws:s3:::{bucket_name}"
      ],
      "Effect": "Allow"
    }
  ]
}
```


   **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. 16), you change the trusted entity to MediaPackage.
To create the service role for an EC2 (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.
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

   The IAM policy defines the permissions that AWS Elemental MediaPackage requires to access Amazon S3. 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.

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"],
   "Resource": [
   "arn:aws:s3:::{bucket_name}/*",
   "arn:aws:s3:::{bucket_name}"
   ],
   "Effect": "Allow"
   }
   ]
   }
   ```


   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: 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. 14). 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. 14).
   - 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 encryption. Digital rights management (DRM) systems provide keys to AWS Elemental MediaPackage for content encryption, and licenses to supported players for decryption.

To encrypt content, you must have a DRM solution provider and be set up to use encryption. For more information, see the section called "Using Encryption" (p. 66).
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. 18)
- Getting Started with VOD Content Delivery in AWS Elemental MediaPackage (p. 20)

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. 18)
- Step 1: Access AWS Elemental MediaPackage (p. 18)
- Step 2: Create a Channel (p. 18)
- Step 3: Create Endpoints (p. 19)
- (Optional) Step 4: Monitor AWS Elemental MediaPackage Activity (p. 19)
- Step 5: Clean Up (p. 20)

Prerequisites

Before you can use AWS Elemental MediaPackage, you need an AWS account and the appropriate permissions to access, view, and edit MediaPackage components. Complete the steps in Setting Up (p. 10), and then return to this tutorial.

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 Channel

The channel is the first component in AWS Elemental MediaPackage. It represents the input to MediaPackage for incoming content from an encoder such as AWS Elemental MediaLive.

AWS Elemental 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 AWS Elemental MediaPackage Channels page, choose Create channel.
Step 3: Create Endpoints

The endpoint is attached to a channel, and represents the output of the 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.

AWS Elemental 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. 80).

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 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. 21)
- Step 1: Access AWS Elemental MediaPackage (p. 21)
- Step 2: Create a Packaging Group (p. 21)
- Step 3: Create a Packaging Configuration (p. 21)
- Step 4: Create an Asset (p. 22)
- Step 5: Provide Playback URLs (p. 22)
- (Optional) Step 6: Monitor AWS Elemental MediaPackage Activity (p. 23)
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. 10), and then return to this tutorial.
- You have file-based source content in one or more Amazon S3 buckets.

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 group enables you to define what kind of outputs you want, and then to apply these output definitions to multiple assets. For example, you have 15 pieces of source content. You want them all to serve DASH, HLS, and encrypted HLS outputs, so you define one packaging group with these types of packaging configurations. You then select that group on all the assets. You don't have to create a new configuration for each asset.

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 kidsmovies. 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.

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, select Add and choose New configuration.
4. For ID, enter a name that describes the configuration, such as hls_movies. 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 represents the source content in AWS Elemental MediaPackage. You use the MediaPackage asset to ingest file-based videos. Playback requests point to specific packaging configurations on the asset.

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 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. 21).

7. Choose Ingest assets.

Step 5: Provide Playback URLs

After creating the asset, AWS Elemental MediaPackage begins preparing to serve the package 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 completes, MediaPackage sends an Amazon CloudWatch event to your account.

On the asset, AWS Elemental 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 AWS Elemental MediaPackage console, go to the Assets page and choose the ID of the asset that you created in Step 4: Create an Asset (p. 22).

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 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 packagingConfiguration to view metrics per configuration). For a list of AWS Elemental MediaPackage metrics, see AWS Elemental MediaPackage VOD Content Metrics (p. 85).

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 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.
- **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. 24)
- Working with Endpoints in AWS Elemental MediaPackage (p. 28)

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.

**Topics**
- Creating a Channel (p. 24)
- Viewing Channel Details (p. 25)
- Editing a Channel (p. 26)
- Rotating Credentials on an Input URL (p. 26)
- Deleting a Channel (p. 27)
- Adding an Endpoint to a Channel (p. 27)

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. 4).

When you're creating a channel, you will receive an error if you exceed the limits 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 limits, 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, contact AWS Support. For more information about limits in AWS Elemental MediaPackage, see Limits in AWS Elemental MediaPackage (p. 100).

**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. 4).

**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. 97).

**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)**

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. 44)) 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. 28).
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. 28)
- Viewing All Endpoints Associated with a Channel (p. 43)
- Viewing a Single Endpoint (p. 43)
- Editing an Endpoint (p. 43)
- Deleting an Endpoint (p. 44)
- Previewing an Endpoint (p. 44)

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. 4).

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. 29)
- Creating a Microsoft Smooth Streaming Endpoint (p. 32)
- Creating a Common Media Application Format (CMAF) Endpoint (p. 35)
- Creating a DASH Endpoint (p. 39)
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. 25).
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. 29)
   - Packager Settings Fields (p. 30)
   - Encryption Fields (p. 30)
   - Access Control Fields (p. 32)
   - Streams to Include Fields (p. 32)
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 limits 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 limits, or you have already reached the maximum number of endpoints allowed on this channel. If you think you received this error wrongfully, contact AWS Support. For more information about limits in AWS Elemental MediaPackage, see Limits in AWS Elemental MediaPackage (p. 100).

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, type a name that describes the endpoint. The ID is the primary identifier for the endpoint, and must be unique for your account in the region.
2. (Optional) For Description, type any descriptive text that helps you to identify the endpoint later.
3. For Manifest name, type 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 type 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. 64).
5. (Optional) To delay when content is available to players, type the duration (in seconds) for the delay in Time delay. The minimum time is five 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. 68).
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**. 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.
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 AWS Elemental MediaPackage to 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.

7. (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.

### Encryption Fields

Protect your content from unauthorized use through encryption. Digital rights management (DRM) systems provide keys to AWS Elemental MediaPackage for content encryption, and licenses to supported players for decryption.

**Note**
To encrypt content, you must have a DRM solution provider, and be set up to use encryption. For information, see the section called “Using Encryption” (p. 66).
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 for DASH and exactly one for the other streaming protocols. If you provide more than one system ID, enter them on separate lines, and do not separate them with commas or any other punctuation. 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. 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.

      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.

   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

   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 serve content to all requesting IP address, choose **Allow all incoming clients**.
2. To limit the IP addresses that this endpoint serves, choose **Restrict by IP address**.
3. In **Whitelist**, type the IP addresses that this endpoint serves content to.

**Streams to Include Fields**

1. (Optional) For **Stream 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. 25).
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. 33)
   - **Packager Settings Fields** (p. 33)
   - **Encryption Fields** (p. 34)
   - **Access Control Fields** (p. 34)
   - **Streams to Include Fields** (p. 35)
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 limits 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 limits, or you have already reached the maximum number of endpoints allowed on this channel. If you think you received this error wrongfully, contact AWS Support. For more information about limits in AWS Elemental MediaPackage, see Limits in AWS Elemental MediaPackage (p. 100).

**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**, type a name that describes the endpoint. The ID is the primary identifier for the endpoint, and must be unique for your account in the region.
2. (Optional) For **Description**, type any descriptive text that helps you to identify the endpoint later.
3. For **Manifest name**, type 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 type 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. 64).
5. (Optional) To delay when content is available to players, type the duration (in seconds) for the delay in **Time delay**. The minimum time is five 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 encryption. Digital rights management (DRM) systems provide keys to AWS Elemental MediaPackage for content encryption, and licenses to supported players for decryption.

**Note**
To encrypt content, you must have a DRM solution provider and be set up to use encryption. For information, see the section called "Using Encryption" (p. 66).

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 for DASH and exactly one for the other streaming protocols. If you provide more than one system ID, enter them on separate lines, and do not separate them with commas or any other punctuation. 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. 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::44445556666: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.
      
      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.

Access Control Fields

1. To serve content to all requesting IP address, choose **Allow all incoming clients**.
2. To limit the IP addresses that this endpoint serves, choose **Restrict by IP address**.
3. In **Whitelist**, type the IP addresses that this endpoint serves content to.
Streams to Include Fields

1. (Optional) For **Stream 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. 25).

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. 36)
   - Packager Settings Fields (p. 36)
   - HLS Manifest Fields (p. 36)
   - Encryption Fields (p. 37)
   - Access Control Fields (p. 38)
   - Streams to Include Fields (p. 38)

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 limits 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 limits, or you have already reached the maximum number of endpoints allowed on this channel. If you think you received this error wrongfully, contact AWS.
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, type a name that describes the endpoint. The ID is the primary identifier for the endpoint, and must be unique for your account in the region.
2. (Optional) For Description, type any descriptive text that helps you to identify the endpoint later.
3. For Manifest name, type 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 type 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. 64).
5. (Optional) To delay when content is available to players, type the duration (in seconds) for the delay in Time delay. The minimum time is five 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) 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.

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 AWS Elemental MediaPackage to 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.

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 encryption. Digital rights management (DRM) systems provide keys to AWS Elemental MediaPackage for content encryption, and licenses to supported players for decryption.

**Note**

To encrypt content, you must have a DRM solution provider, and be set up to use encryption. For information, see the section called “Using Encryption” (p. 66).

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 for DASH and exactly one for the other streaming protocols. If you provide more than one system ID, enter them on separate lines, and do not separate them with commas or any other punctuation. 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:

```plaintext
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:

```plaintext
arn:aws:iam::444455566666: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.

f. (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:

```plaintext
1800
```

### Access Control Fields

1. To serve content to all requesting IP address, choose **Allow all incoming clients**.
2. To limit the IP addresses that this endpoint serves, choose **Restrict by IP address**.
3. In **Whitelist**, type the IP addresses that this endpoint serves content to.

### Streams to Include Fields

1. (Optional) For **Stream 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. 25).
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. 39)
   - Packager Settings Fields (p. 40)
   - Encryption Fields (p. 41)
   - Access Control Fields (p. 42)
   - Streams to Include Fields (p. 42)
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 limits 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 limits, or you have already reached the maximum number of endpoints allowed on this channel. If you think you received this error wrongfully, contact AWS Support. For more information about limits in AWS Elemental MediaPackage, see Limits in AWS Elemental MediaPackage (p. 100).

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, type a name that describes the endpoint. The ID is the primary identifier for the endpoint, and must be unique for your account in the region.
2. (Optional) For Description, type any descriptive text that helps you to identify the endpoint later.
3. For Manifest name, type 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 type 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. 64).**

5. (Optional) To delay when content is available to players, type the duration (in seconds) for the delay in **Time delay.** The minimum time is five 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.

   - 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. 72).**

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.
   - 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. 74).

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. 70).

   **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](#).

### Encryption Fields

Protect your content from unauthorized use through encryption. Digital rights management (DRM) systems provide keys to AWS Elemental MediaPackage for content encryption, and licenses to supported players for decryption.

**Note**
To encrypt content, you must have a DRM solution provider and be set up to use encryption. For information, see the section called "Using Encryption" (p. 66).

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 for DASH and exactly one for the other streaming protocols. If you provide more than one system ID, enter them on separate lines, and do not separate them with commas or any other
Creating an Endpoint

punctuation. 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. 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.

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.

f. (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

Access Control Fields

1. To serve content to all requesting IP address, choose Allow all incoming clients.
2. To limit the IP addresses that this endpoint serves, choose Restrict by IP address.
3. In Whitelist, type the IP addresses that this endpoint serves content to.

Streams to Include Fields

1. (Optional) For Stream 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. 25).

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. 25).

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. 25).
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. 28).
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. 25).
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. 25).
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.

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. 46)
- Working with Packaging Configurations in AWS Elemental MediaPackage (p. 48)
- Working with Assets in AWS Elemental MediaPackage (p. 59)

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. 46)
- Viewing Packaging Group Details (p. 47)
- Editing a Packaging Group (p. 47)
- Deleting a Packaging Group (p. 48)
- Adding a Packaging Configuration to a Packaging Group (p. 48)

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.
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 limits 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 limits, 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, contact AWS Support. For more information about limits in MediaPackage, see Limits in AWS Elemental MediaPackage (p. 100).

**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**

You can't edit a packaging group. To make changes, create a new group and delete the original.

- To create a group, see Creating a Packaging Group (p. 46).
- To delete a group, see Deleting a Packaging Group (p. 48).
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. 59).
- To delete an asset, see Deleting an Asset (p. 62).

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. 49).

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 distribution network (CDN), such as Amazon CloudFront. The configuration outputs 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
Creating a Packaging Configuration

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

To create a packaging configuration, 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 Packaging_configurations in the AWS Elemental MediaPackage 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 MediaPackage might get picked up for inclusion in diagnostic logs or Amazon CloudWatch Events.

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 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. 50)
Creating a Packaging Configuration

- Manifest Settings Fields (p. 50)
- Stream Selection Fields (p. 51)
- Encryption Fields (p. 51)

7. Choose Save.

If you exceed the limits 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 limits, 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, contact AWS Support. For more information about limits in MediaPackage, see Limits in AWS Elemental MediaPackage (p. 100).

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. 68).

4. (Optional) To tell the service to repeat the key before every segment of the manifest, select Repeat EXT-X-KEY. 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.

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 encryption. Digital rights management (DRM) systems provide keys to AWS Elemental MediaPackage for content encryption, and licenses to supported players for decryption.

**Note**
To encrypt content, you must have a DRM solution provider, and be set up to use encryption. For information, see the section called “Using Encryption” (p. 66).

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.

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:

   ```
   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:
4. For **System IDs**, enter unique identifiers for your streaming protocol and DRM system. Provide up to two IDs for DASH and exactly one for 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](#)
   - [Manifest Settings Fields](#)
   - [Stream Selection Fields](#)
   - [Encryption Fields](#)
7. Choose **Save**.

If you exceed the limits 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 limits, 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, contact **AWS Support**. For more information about limits in MediaPackage, see **Limits in AWS Elemental MediaPackage** (p. 100).

### 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.
Creating a 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

You can't change the values on **Manifest layout** and **Segment template format**. AWS Elemental MediaPackage supports only the default format for outbound DASH manifests for VOD content.

**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 encryption. Digital rights management (DRM) systems provide keys to AWS Elemental MediaPackage for content encryption, and licenses to supported players for decryption.

**Note**

To encrypt content, you must have a DRM solution provider, and be set up to use encryption. For information, see the section called "Using Encryption" (p. 66).

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 two IDs for DASH and exactly one for 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)**

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 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. 54)
   - Manifest Settings Fields (p. 55)
   - Stream Selection Fields (p. 55)
   - Encryption Fields (p. 55)
7. Choose **Save**.

If you exceed the limits 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 limits, 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, contact **AWS Support**. For more information about limits in MediaPackage, see **Limits in AWS Elemental MediaPackage** (p. 100).

**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 encryption. Digital rights management (DRM) systems provide keys to AWS Elemental MediaPackage for content encryption, and licenses to supported players for decryption.

**Note**

To encrypt content, you must have a DRM solution provider, and be set up to use encryption. For information, see the section called "Using Encryption" (p. 66).

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.

   The following example shows a URL:
Creating a Packaging Configuration

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 two IDs for DASH and exactly one for 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](https://dashif.org/specifications/dash-implementation-profiles/). 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)**

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. 56)
   - **Manifest Settings Fields** (p. 57)
   - **Stream Selection Fields** (p. 57)
   - **Encryption Fields** (p. 58)
7. Choose **Save**.

If you exceed the limits 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 limits, 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, contact [AWS Support](https://aws.amazon.com/support/). For more information about limits in MediaPackage, see [Limits in AWS Elemental MediaPackage](https://aws.amazon.com/mediapackage/#limits) (p. 100).

**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) To tell the service to repeat the key before every segment of the manifest, select **Repeat EXT-X-KEY**. 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.

### 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 encryption. Digital rights management (DRM) systems provide keys to AWS Elemental MediaPackage for content encryption, and licenses to supported players for decryption.

**Note**

To encrypt content, you must have a DRM solution provider, and be set up to use encryption. For information, see the section called "Using Encryption" (p. 66).

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.

   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::44445556666:role/SpekeAccess
   ```

3. For **System IDs**, enter unique identifiers for your streaming protocol and DRM system. Provide up to two IDs for DASH and exactly one for 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.

### 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)**

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. 46).
- To delete a configuration, see Deleting a Packaging Configuration (p. 48).

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.

Topics
- Ingesting an Asset (p. 60)
- Viewing Asset Details (p. 62)
- Editing an Asset (p. 62)
- Deleting an Asset (p. 62)

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.

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. 60)
   - Asset Details Fields (p. 61)
   - Packaging Settings Field (p. 62)
5. Choose Ingest assets.

If you exceed the limits 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 limits, 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, contact AWS Support. For more information about limits in MediaPackage, see Limits in AWS Elemental MediaPackage (p. 100).

Asset Access Fields

The following fields describe how AWS Elemental MediaPackage accesses the source content.

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. 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 or .m3u8 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**

```
MovieNight20171126093045
```

**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/.
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. 60).
- To delete an asset, see Deleting a Packaging Configuration (p. 62).

**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.
Features in AWS Elemental MediaPackage

The following sections describe the features that are available in AWS Elemental MediaPackage and how they work.

**Topics**

- Time-shifted Viewing Reference in AWS Elemental MediaPackage (p. 64)
- Using Encryption in AWS Elemental MediaPackage (p. 66)
- Rendition Groups Reference in AWS Elemental MediaPackage (p. 68)
- DASH Manifest Options in AWS Elemental MediaPackage (p. 69)

Time-shifted Viewing Reference in AWS Elemental MediaPackage

Time-shifted viewing is available with only 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. AWS Elemental MediaPackage allows a content retention window of up to 336 hours (14 days) for time-shifted viewing. Time-shifted functionality is controlled by the MediaPackage endpoint and by the start and end parameters provided in the content request URL.

**To enable time-shifted viewing**

1. Enable time-shifted viewing by typing a value for **Startover time** 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 are sent to this endpoint, AWS Elemental MediaPackage generates a manifest within the window that is indicated in the request. 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 nine hours of content.

   For packager-specific rules about how you can notate the parameters, see [Rules for Start and End Parameters](p. 65).

   The start and end parameters determine the time boundaries of the manifest. Expected behaviors are as follows:

   **Note**
   In all cases, the maximum manifest length is nine hours.
• 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. 6).

• 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. You can use a start time that's up to 9 hours in the past.

• 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.

Topics

• DASH Parameter Rules (p. 65)
• HLS and CMAF Parameter Rules (p. 66)
• Microsoft Smooth Parameter Rules (p. 66)

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/v1/997cbb27697d4863bb65488133bff26f/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/v1/997cbb27697d4863bb65488133bff26f/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

**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/v1/1f76b3b4f394c44a485c0e4e560afe50e/start/1513717228/end/1513720828/drama.ism/Manifest
```

Using Encryption in AWS Elemental MediaPackage

Protect your content from unauthorized use through encryption. Digital rights management (DRM) systems provide keys to AWS Elemental MediaPackage for content encryption, and licenses to supported players for decryption.
Using Encrypted Content Keys with DRM

Encrypted content keys are available with only live workflows in AWS Elemental MediaPackage.

For the most secure DRM encryption solution, use encrypted content keys in addition to encrypted content. To use encrypted content keys, you must import suitable certificates into the AWS Certificate Manager. For information about ACM, see the AWS Certificate Manager User Guide.

Run AWS Certificate Manager in the same Region as you run AWS Elemental MediaPackage.

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. 28).

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.
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, AWS Elemental 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.

Topics

• When to Use Rendition Groups (p. 68)
• When Not to Use Rendition Groups (p. 68)

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:

• 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, AWS Elemental 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

  AWS Elemental 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 audio 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 bit rate selection).

To limit the tracks available to a player, use the stream selection options from the AWS Elemental MediaPackage console or the REST API.

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.

Default DASH manifest

The following is a truncated example of a DASH manifest with no treatments:

```xml
<MPD>
  <Period>
    <AdaptationSet>
      <Representation>
        <SegmentTemplate>
          <SegmentTimeline>
            <S />
          </SegmentTimeline>
        </SegmentTemplate>
      </Representation>
    </AdaptationSet>
  </Period>
</MPD>
```

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. 70).
- Reduce the length of the manifest to make processing and playback more efficient. See Compacted DASH Manifests (p. 72).
- Control what segment information is used in the media URL in the SegmentTemplate properties. See DASH Manifest Segment Template Format (p. 74).

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. 39).
- 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 \((scte35 ptsAdjustment + scte35 ptsTime) / (EventStream timescale)\).

**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" availExpected="4">
          <scte35:Program><scte35:SpliceTime ptsTime="3783780"/></scte35:Program>
        </scte35:SpliceInsert>
      </scte35:SpliceInfoSection>
    </Event>
  </EventStream>
```
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>
            <S t="361301" d="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">
        <SegmentTemplate timescale="48000" media="index_audio_2_0_$Number$.mp4?m=1528413503" initialization="index_audio_2_0_init.mp4?m=1528413503" startNumber="6" presentationTimeOffset="0">
          <SegmentTimeline>
            <S t="578305" d="96256" r="3"/>
            <S t="1058561" d="96256" r="5"/>
            <S t="1636097" d="96256" r="3"/>
          </SegmentTimeline>
        </SegmentTemplate>
      </Representation>
    </AdaptationSet>
  </Period>
  <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" availExpected="4">
            <scte35:Program><scte35:SpliceTime ptsTime="3783780"/></scte35:Program>
          </scte35:SpliceInsert>
        </scte35:SpliceInfoSection>
      </Event>
    </EventStream>
  </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="183265" tier="4095">
      <scte35:TimeSignal>
        <scte35:SpliceTime ptsTime="1350000"/>
      </scte35:TimeSignal>
      <scte35:SegmentationDescriptor segmentationEventId="1073741825" segmentationEventCancelIndicator="false" segmentationDuration="450000">
        <scte35:DeliveryRestrictions webDeliveryAllowedFlag="false" noRegionalBlackoutFlag="true" archiveAllowedFlag="true" deviceRestrictions="3"/>
        <scte35:SegmentationUpid segmentationUpidType="1" segmentationUpidLength="3" segmentationTypeId="48" segmentNum="0" segmentsExpected="0" segment="012345"/>
      </scte35:SegmentationDescriptor>
    </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_5_0_$Number$.mp4?m=1543947824" initialization="index_video_5_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 its 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:

```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"/>

  <Representation id="2" width="854" height="480" frameRate="30/1" bandwidth="1000000" codecs="avc1.640029"/>
</AdaptationSet>
```
For information about compacting a DASH manifest, see How AWS Elemental MediaPackage Compacts Manifests (p. 74).

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.
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. 75).

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. 39). 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:

   ```xml
   <SegmentTemplate timescale="30" media="index_video_1_0_\$Time$.mp4?m=1122792372" initialization="index_video_1_0_init.mp4?m=1122792372" startNumber="2937928">
   ...<SegmentTimeline>...
   </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.

   ```xml
   <SegmentTemplate timescale="30000" media="155_video_1_2_\$Time$.mp4?m=1545421124" initialization="155_video_1_2_init.mp4?m=1545421124" startNumber="710">
   >>>>>> mainline
   <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 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. 76)
- duration Attribute with Compacted DASH Manifests (p. 78)

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. 39). 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 13500 (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" bitstreamSwitchings="true">
     <Representation id="1" width="852" height="480" frameRate="30/1" bandwidth="1200000" codecs="avc1.4D401F">
   ```
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:

\[
((\text{wall clock time} - \text{availabilityStart Time}) / (\text{duration} / \text{timescale})) + \text{startNumber}
\]

**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 \((600 \text{ seconds elapsed time}) / (3 \text{ second segment durations}) = 200 \text{ elapsed segments. Adding those segments to the 175032 start segment makes the most recent segment 175232.}\)

**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.
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. 72).

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

```
<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"/>
  <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>
```

Compacted manifest with `duration` attribute

```
<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" duration="60000"/>
  <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>
```
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. 79)
- Monitoring AWS Elemental MediaPackage with Amazon CloudWatch Events (p. 88)
- Logging AWS Elemental MediaPackage API Calls with AWS CloudTrail (p. 91)

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**

79
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. 80)
- AWS Elemental MediaPackage VOD Content Metrics (p. 85)

## 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></td>
<td>Valid dimension:</td>
</tr>
<tr>
<td></td>
<td>• Combination of IngestEndpoint and OriginEndpoint</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>
<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>
</tbody>
</table>
### Metric

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>- Smallest individual output request (in bytes) made to AWS Elemental MediaPackage.</td>
</tr>
<tr>
<td>SampleCount</td>
<td>- Number of requests that is used in the statistical calculation.</td>
</tr>
<tr>
<td>Sum</td>
<td>- 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

#### EgressRequestCount

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

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
## Metric: EgressResponseTime

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

### Valid statistics:
- **Average**: average amount of time \((\text{Sum}/\text{SampleCount})\) 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**
### Metric Description

<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>
<tr>
<td></td>
<td><strong>Valid statistics:</strong></td>
</tr>
<tr>
<td></td>
<td>• Average – average bytes (Sum/SampleCount) that AWS Elemental MediaPackage receives over the configured interval.</td>
</tr>
<tr>
<td></td>
<td>• Maximum – largest individual input request (in bytes) made to AWS Elemental MediaPackage.</td>
</tr>
<tr>
<td></td>
<td>• Minimum – smallest individual input 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 receives over the configured interval.</td>
</tr>
<tr>
<td></td>
<td><strong>Valid dimensions:</strong></td>
</tr>
<tr>
<td></td>
<td>• Channel</td>
</tr>
<tr>
<td></td>
<td>• Combination of Channel and IngestEndpoint</td>
</tr>
<tr>
<td></td>
<td>• No dimension</td>
</tr>
</tbody>
</table>
### Live Content Metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IngressResponseTime</td>
<td>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</td>
</tr>
</tbody>
</table>

### 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. Value: The autogenerated GUID of the channel. Can be used alone or with other dimensions: • 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>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<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</td>
</tr>
<tr>
<td></td>
<td>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>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<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>
<tr>
<td></td>
<td>Valid statistics:</td>
</tr>
<tr>
<td></td>
<td>• Average – average amount of time (Sum/SampleCount) that it takes AWS Elemental MediaPackage to process output requests over the configured interval.</td>
</tr>
<tr>
<td></td>
<td>• Maximum – longest amount of time (in milliseconds) that it takes AWS Elemental MediaPackage to process an output request and provide a response.</td>
</tr>
<tr>
<td></td>
<td>• Minimum – shortest amount of time (in milliseconds) that it takes AWS Elemental MediaPackage to process an output request and provide a response.</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 amount of time that it takes AWS Elemental MediaPackage to process output requests over the configured interval.</td>
</tr>
<tr>
<td></td>
<td>Valid dimensions:</td>
</tr>
<tr>
<td></td>
<td>• PackagingConfiguration</td>
</tr>
</tbody>
</table>

**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>
<tr>
<td></td>
<td>Value: The autogenerated GUID of the configuration.</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 configuration.</td>
</tr>
</tbody>
</table>
### 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, and specific details about the event.

#### Monitoring AWS Elemental MediaPackage with Amazon 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.

For more information about creating rules in CloudWatch Events, see the [Amazon CloudWatch Events User Guide](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/CloudWatch_GettingStarted.html).

For a list of events that AWS Elemental MediaPackage emits, see [AWS Elemental MediaPackage Events](p. 88).

### Dimension | Description
--- | ---
| | With the `statusCodeRange` dimension to show metrics for the specified configuration that is associated with the specified status code.

#### StatusCodeRange

Metrics are shown for the specified status code range.

Value: 2xx, 3xx, 4xx, or 5xx.

Can be used alone or with other dimensions:

- Alone to show all output requests for the specified status range.
- With the `channel` dimension to show output requests for all endpoints that are associated with the specified channel, with the specified status code range.
- 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.
endpoint affected, and more. 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.

Topics
- Input Notification Events (p. 89)
- Key Provider Notification Events (p. 90)

Input Notification Events

You get input notification events if one of these actions occurs:

- For live content, a channel in AWS Elemental MediaPackage exceeds the limit for the number of input streams. For information about limits, see Limits in AWS Elemental MediaPackage (p. 100).
- For live content, AWS Elemental 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. 4).

- For video on demand (VOD) content, an asset in AWS Elemental MediaPackage changes status. You get notifications for the following events:
  - IngestStart
  - IngestError
  - IngestComplete

Example Maximum Input Streams Exceeded Event

```
{
"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/262fef182d464b399fcbea1364df682"
],
"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,...,ex_15.m3u8,index_16.m3u8,index_17.m3u8,index_18.m3u8,index_19.m3u8,index_20.m3u8,index_21.m3u8,index_22.m3u8,index_23.m3u8"
}
}
```

Example Input Switch Event

```
{
"id": "8f9b8e72-0b31-e883-f19c-aec84742f3ce",
"detail": {
  "event": "InputSwitch",
  "oldValue": “index_1.m3u8”,
  "newValue": “index_2.m3u8”
}
}
```
Creating Event Notifications

Example Ingest Status Event

```
{
  "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"
  }
}
```

Key Provider Notification Events

When you're using content encryption on an endpoint, AWS Elemental MediaPackage can't reach the key provider. For information about DRM and encryption, see https://docs.aws.amazon.com/speke/latest/documentation/.

```
{
  "id": "7bf73129-1428-4cd3-a780-98ds273d1602",
  "detail-type": "MediaPackage Key Provider 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: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 detail-type from the event, as described in Input Notification Events (p. 89) and Key Provider Notification Events (p. 90).
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:

- Overview for Creating a Trail
- CloudTrail Supported Services and Integrations
- Configuring Amazon SNS Notifications for CloudTrail
- Receiving CloudTrail Log Files from Multiple Regions and Receiving CloudTrail Log Files from Multiple Accounts

All MediaPackage actions are logged by CloudTrail and are documented in the AWS Elemental MediaPackage API Reference. For example, calls to the `CreateChannel`, `CreateOriginEndpoint`, and `RotateIngestEndpointCredentials` operations generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or AWS Identity and Access Management (IAM) user credentials
- Whether the request was made with temporary security credentials for a role or federated user
- Whether the request was made by another AWS service

For more information, see the CloudTrail userIdentity Element.

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:

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefgHiJkL123456789",
        "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": "ABCDEFGHJKLMN789",
    "arn": "arn:aws:iam::444455556666:role/Admin",
    "accountId": "444455556666",
    "userName": "Admin"
  }
},
"eventTime": "2018-12-18T00:50:59Z",
"eventSource": "mediapackage.amazonaws.com",
"eventName": "UpdateChannel",
"awsRegion": "us-west-2",
"sourceIPAddress": "203.0.113.17",
"userAgent": "aws-cli/1.15.71 Python/3.6.5 Darwin/17.7.0 botocore/1.10.70",
"requestParameters": {
  "description": "updated cloudtrail description",
  "id": "cloudtrail-test"
},
"responseElements": {
  "description": "updated cloudtrail description",
  "hlsIngest": {
    "ingestEndpoints": [
      {
        "username": "***",
        "url": "https://mediapackage.us-west-2.amazonaws.com/in/v2/8d0ca97840d94b18b37ad292c131bcad/8d0ca97840d94b18b37ad292c131bcad/channel",
        "password": "***",
        "id": "8d0ca97840d94b18b37ad292c131bcad"
      },
      {
        "username": "***",
        "url": "https://mediapackage.us-west-2.amazonaws.com/in/v2/8d0ca97840d94b18b37ad292c131bcad/9c17f979598543b9be24345d63b3ad30/channel",
        "password": "***",
        "id": "9c17f979598543b9be24345d63b3ad30"
      }
    ]
  }
},
"requestID": "fc158262-025e-11e9-8360-6b7f705fbb8a",
"eventType": "AwsApiCall",
"recipientAccountId": "444455556666"
Tagging AWS Elemental MediaPackage Resources

Tagging is available with only live workflows in AWS Elemental MediaPackage.

A *tag* is a metadata label that you assign or that AWS assigns to an AWS resource. Each tag consists of a *key* and a *value*. For tags that you assign, you define the key and value. For example, you might define the key as *stage* and the value for one resource as *test*.

Tags help you do the following:

- Identify and organize your AWS resources. Many AWS services support tagging, so you can assign the same tag to resources from different services to indicate that the resources are related. For example, you could assign the same tag to an AWS Elemental MediaPackage channel and endpoint that you assign to an AWS Elemental MediaTailor configuration.
- Track your AWS costs. You activate these tags on the AWS Billing and Cost Management dashboard. AWS uses the tags to categorize your costs and deliver a monthly cost allocation report to you. For more information, see Use Cost Allocation Tags in the AWS Billing and Cost Management User Guide.

For tips on using tags, see the AWS Tagging Strategies post on the AWS Answers blog.

The following sections provide more information about tags for AWS Elemental MediaPackage.

Supported Resources in AWS Elemental MediaPackage

The following resources in AWS Elemental MediaPackage support tagging:

- Channels
- Endpoints

Video on demand (VOD) resources don't support tagging. This includes assets, packaging groups, and packaging configurations.

For information about adding and managing tags, see Managing Tags (p. 95).

Tag Restrictions

The following basic restrictions apply to tags on AWS Elemental MediaPackage resources:

- Only live content workflows are supported in MediaPackage
- 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

Managing Tags

Tags are made up of the Key and Value properties on a resource. 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 Resources topic in the AWS Elemental MediaPackage API 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. 97)
- Viewing a Distribution (p. 98)
- Editing a Distribution (p. 98)
- Deleting a Distribution (p. 98)
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. 97)
- Creating a Distribution from Amazon CloudFront (p. 98)

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 Creating a Policy for Amazon CloudFront (p. 12).

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. 24), 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. 26).

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. 28). 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. 25), 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.
Limits in AWS Elemental MediaPackage

The following sections provide information about the limits in AWS Elemental MediaPackage.

Topics
- Live Content Limits (p. 100)
- VOD Content Limits (p. 101)

Live Content Limits

This section describes the limits for live content in AWS Elemental MediaPackage. For information about requesting an increase to soft limits, see AWS Service Limits. Hard limits can’t be changed.

Live Soft Limits

The following table describes limits in AWS Elemental MediaPackage for live content that can be increased. For information about changing limits, see AWS Service Limits.

For some customers, your account limit might be below these published limits. If you believe that you encountered a Resource limit exceeded error wrongfully, create a Service Limit Increase case and provide details such as how many channels or endpoints you have and what you were trying to do.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Default Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Channels</td>
<td>10</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td></td>
</tr>
<tr>
<td>Increasing your channel limit</td>
<td></td>
</tr>
<tr>
<td>doesn’t always mean that you also</td>
<td></td>
</tr>
<tr>
<td>need to increase your endpoints.</td>
<td></td>
</tr>
<tr>
<td>For example, if you need 14</td>
<td></td>
</tr>
<tr>
<td>channels and want to serve</td>
<td></td>
</tr>
<tr>
<td>HLS, HLS encrypted, and DASH</td>
<td></td>
</tr>
<tr>
<td>content from each channel, you</td>
<td></td>
</tr>
<tr>
<td>need only 3 endpoints for each</td>
<td></td>
</tr>
<tr>
<td>channel (one for each output</td>
<td></td>
</tr>
<tr>
<td>type). The default endpoint limit</td>
<td></td>
</tr>
<tr>
<td>is 10 so, although you do need a</td>
<td></td>
</tr>
<tr>
<td>channel limit increase, you don’t</td>
<td></td>
</tr>
<tr>
<td>need to increase your endpoint</td>
<td></td>
</tr>
<tr>
<td>limit. You won’t exceed the</td>
<td></td>
</tr>
<tr>
<td>limit of 10 endpoints <em>per channel</em>.</td>
<td></td>
</tr>
<tr>
<td>Maximum Endpoints per Channel</td>
<td>10</td>
</tr>
<tr>
<td>This is a <em>per channel</em> limit.</td>
<td></td>
</tr>
<tr>
<td>Each endpoint represents the</td>
<td></td>
</tr>
<tr>
<td>output package that you use.</td>
<td></td>
</tr>
<tr>
<td>If one channel serves HLS, HLS</td>
<td></td>
</tr>
<tr>
<td>encrypted, DASH, DASH encrypted,</td>
<td></td>
</tr>
<tr>
<td>Microsoft Smooth, and Microsoft</td>
<td></td>
</tr>
<tr>
<td>Smooth encrypted content, then</td>
<td></td>
</tr>
<tr>
<td>that channel has 6 endpoints and</td>
<td></td>
</tr>
<tr>
<td>falls within the 10 endpoints</td>
<td></td>
</tr>
</tbody>
</table>
Live Hard Limits

The following table describes limits in AWS Elemental MediaPackage for live content that can't be increased.

<table>
<thead>
<tr>
<th>Resource or Operation</th>
<th>Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Stream Limits</td>
<td>20 streams per channel, and 10 tracks per stream</td>
</tr>
<tr>
<td>Maximum Content Retention</td>
<td>336 hours (14 days)</td>
</tr>
<tr>
<td>Maximum Live Manifest Length</td>
<td>5 minutes</td>
</tr>
<tr>
<td>Maximum Time-shifted Manifest Length</td>
<td>9 hours</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 Limits

This section describes the limits for video on demand (VOD) content in AWS Elemental MediaPackage. For information about requesting an increase to soft limits, see AWS Service Limits. Hard limits can't be changed.

VOD Soft Limits

The following table describes limits in AWS Elemental MediaPackage for VOD content that can be increased. For information about changing limits, see AWS Service Limits.

For some customers, your account limit might be below these published limits. If you believe that you encountered a Resource limit exceeded error wrongfully, create a Service Limit Increase case and provide details such as how many channels or endpoints you have and what you were trying to do.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Default Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Packaging Groups</td>
<td>10</td>
</tr>
</tbody>
</table>

**Note**

Increasing your packaging group limit 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
Maximum Packaging Configurations per Packaging Group 10
This is a per packaging group limit. 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 limit. If you have 10 packaging groups set up this same way, then you still haven't exceeded the limit because each group uses only 6 configurations.

Maximum Assets per Packaging Group 1000
This is a per packaging group limit. If you have 1100 assets spread out over multiple packaging groups, then you still haven't exceeded the limit if each group has no more than 1000 assets.

VOD Hard Limits
The following table describes limits within AWS Elemental MediaPackage for VOD content that can't be increased.

<table>
<thead>
<tr>
<th>Resource or Operation</th>
<th>Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Stream Limits</td>
<td>20 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>
<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>

102
# 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
- **Latest documentation update:** May 17, 2019

<table>
<thead>
<tr>
<th>update-history-change</th>
<th>update-history-description</th>
<th>update-history-date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Added video on demand (VOD) topics. (p. 1)</td>
<td>Throughout guide, added topics about working with VOD content: <em>VOD Content Processing, Allowing AWS Elemental MediaPackage to Access Amazon Simple Storage Service, VOD Content Delivery, Delivering VOD Content, VOD Content Metrics, and VOD Content Limits.</em></td>
<td>May 17, 2019</td>
</tr>
<tr>
<td>Added further information about DASH manifest SegmentTemplate format options. (p. 74)</td>
<td>Added the <em>Duration Attribute</em> 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. 101)</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. 6)</td>
<td>Added the <em>Live and VOD Manifest Reference</em> topic that explains when AWS Elemental MediaPackage serves a live or VOD manifest.</td>
<td>April 16, 2019</td>
</tr>
<tr>
<td>Added tagging information. (p. 94)</td>
<td>Added <em>Tagging Resources</em> 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. 74)</td>
<td>Added the <em>DASH Manifest Segment Template Format</em> 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. 69)</td>
<td>Added <em>DASH Manifest Options</em> 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. 91)</td>
<td>Added <em>Logging AWS Elemental MediaPackage API Calls with AWS CloudTrail</em> topic to discuss using CloudTrail to log actions in the AWS Elemental MediaPackage API.</td>
<td>December 21, 2018</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Added information about compact DASH manifests. (p. 72)</td>
<td>Added a <em>Compacted DASH Manifests</em> 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. 101)</td>
<td>AWS Elemental MediaPackage now retains content for 336 hours (14 days).</td>
<td>November 13, 2018</td>
</tr>
<tr>
<td>Added content key encryption to DRM encryption (p. 67)</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. (p. 4)</td>
<td>Added <em>How Input Redundancy Works</em> topic to discuss how AWS Elemental MediaPackage can receive two identical streams for back-up purposes.</td>
<td>August 28, 2018</td>
</tr>
<tr>
<td>Added Amazon CloudFront console integration information. (p. 96)</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. (p. 70)</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. (p. 96)</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>
</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>Corrected links and added whitelisting.</td>
<td>Corrected links to the AWS Elemental MediaPackage console and AWS Elemental MediaPackage API Reference. In Working with Endpoints, added reference to access control fields.</td>
<td>December 1, 2017</td>
</tr>
<tr>
<td>Added IAM policy information specific to AWS Elemental MediaPackage.</td>
<td>In Setting Up AWS Elemental MediaPackage (p. 10), 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 Limits in AWS Elemental MediaPackage (p. 100), 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. 10), added information about policies specific to AWS Elemental MediaPackage.</td>
<td>January 5, 2018</td>
</tr>
<tr>
<td>Added Amazon CloudWatch Events information.</td>
<td>Added the section called “Monitoring with CloudWatch Events” (p. 88) section about the CloudWatch Events that AWS Elemental MediaPackage supports.</td>
<td>February 14, 2018</td>
</tr>
<tr>
<td>Added CMAF endpoint information.</td>
<td>Added Creating a Common Media Application Format (CMAF) Endpoint (p. 35) section for new output type.</td>
<td>April 6, 2018</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>Updated feature functionality.</td>
<td>In <em>Features of AWS Elemental MediaPackage</em> (p. 7), added feature support for HDR-10.</td>
<td>April 30, 2018</td>
</tr>
<tr>
<td>Added content delivery network (CDN) info.</td>
<td>Added topic <em>Working with Content Delivery Networks (CDNs)</em> (p. 96) 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.