
AWS Elemental MediaPackage

User Guide



AWS Elemental MediaPackage: User Guide

Copyright © 2019 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

What Is AWS Elemental MediaPackage?	1
Are You a First-Time User of AWS Elemental MediaPackage?	1
Concepts and Terminology	1
How AWS Elemental MediaPackage Works	2
How Input Redundancy Works	3
Features of AWS Elemental MediaPackage	4
Related Services	5
Accessing AWS Elemental MediaPackage	6
Pricing for AWS Elemental MediaPackage	6
Regions for AWS Elemental MediaPackage	6
Setting Up	7
Signing Up for AWS	7
Creating an Admin IAM User	7
Creating a Non-Admin IAM User	8
Creating a Policy for Amazon CloudFront	9
(Optional) Setting Up Encryption	10
Getting Started	11
Prerequisites	11
Step 1: Access AWS Elemental MediaPackage	11
Step 2: Create a Channel	11
Step 3: Create Endpoints	12
(Optional) Step 4: Monitor AWS Elemental MediaPackage Activity	12
Step 5: Clean Up	13
Working with Channels	14
Creating a Channel	14
Viewing Channel Details	15
Editing a Channel	15
Rotating Credentials on an Input URL	16
Deleting a Channel	16
Adding an Endpoint to a Channel	17
Working with Endpoints	18
Creating an Endpoint	18
Creating an HLS Endpoint	19
Creating a Microsoft Smooth Streaming Endpoint	22
Creating a Common Media Application Format (CMAF) Endpoint	25
Creating a DASH Endpoint	29
Viewing All Endpoints Associated with a Channel	33
Viewing a Single Endpoint	33
Editing an Endpoint	33
Deleting an Endpoint	34
Previewing an Endpoint	34
Time-shifted Viewing	36
Rules for Start and End Parameters	37
DASH Parameter Rules	37
HLS and CMAF Parameter Rules	37
Microsoft Smooth Parameter Rules	38
Using Encryption	39
Using Encrypted Content Keys with DRM	39
Rendition Groups	41
When to Use Rendition Groups	41
When Not to Use Rendition Groups	41
DASH Manifest Treatments	43
Multi-period DASH	44
How it Works	44

Compacted DASH Manifests	46
How AWS Elemental MediaPackage Compacts Manifests	48
DASH Manifest Segment Template Format	48
How It Works	49
Monitoring	50
Monitoring with CloudWatch Metrics	50
AWS Elemental MediaPackage Metrics	51
Monitoring with CloudWatch Events	56
AWS Elemental MediaPackage Events	57
Creating Event Notifications	58
Logging AWS Elemental MediaPackage API Calls with AWS CloudTrail	59
AWS Elemental MediaPackage Information in CloudTrail	59
Understanding AWS Elemental MediaPackage Log File Entries	60
Working with CDNs	62
Creating a Distribution	63
From AWS Elemental MediaPackage	63
From Amazon CloudFront	64
Viewing a Distribution	64
Editing a Distribution	64
Deleting a Distribution	64
Limits	66
Soft Limits	66
Hard Limits	66
Related Information	68
Document History	69
Earlier Updates	70
AWS Glossary	72

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 \(p. 2\)](#)
- [Features of AWS Elemental MediaPackage \(p. 4\)](#)
- [Related Services \(p. 5\)](#)
- [Accessing AWS Elemental MediaPackage \(p. 6\)](#)
- [Pricing for AWS Elemental MediaPackage \(p. 6\)](#)
- [Regions for AWS Elemental MediaPackage \(p. 6\)](#)

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 \(p. 2\)](#)
- [Concepts and Terminology \(p. 1\)](#)
- [Features of AWS Elemental MediaPackage \(p. 4\)](#)
- [Getting Started with AWS Elemental MediaPackage \(p. 11\)](#)

Concepts and Terminology

AWS Elemental MediaPackage includes the following components:

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.

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).

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).

Here is the general processing flow:

1. An upstream encoder (such as AWS Elemental MediaLive) sends an HLS live stream 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.

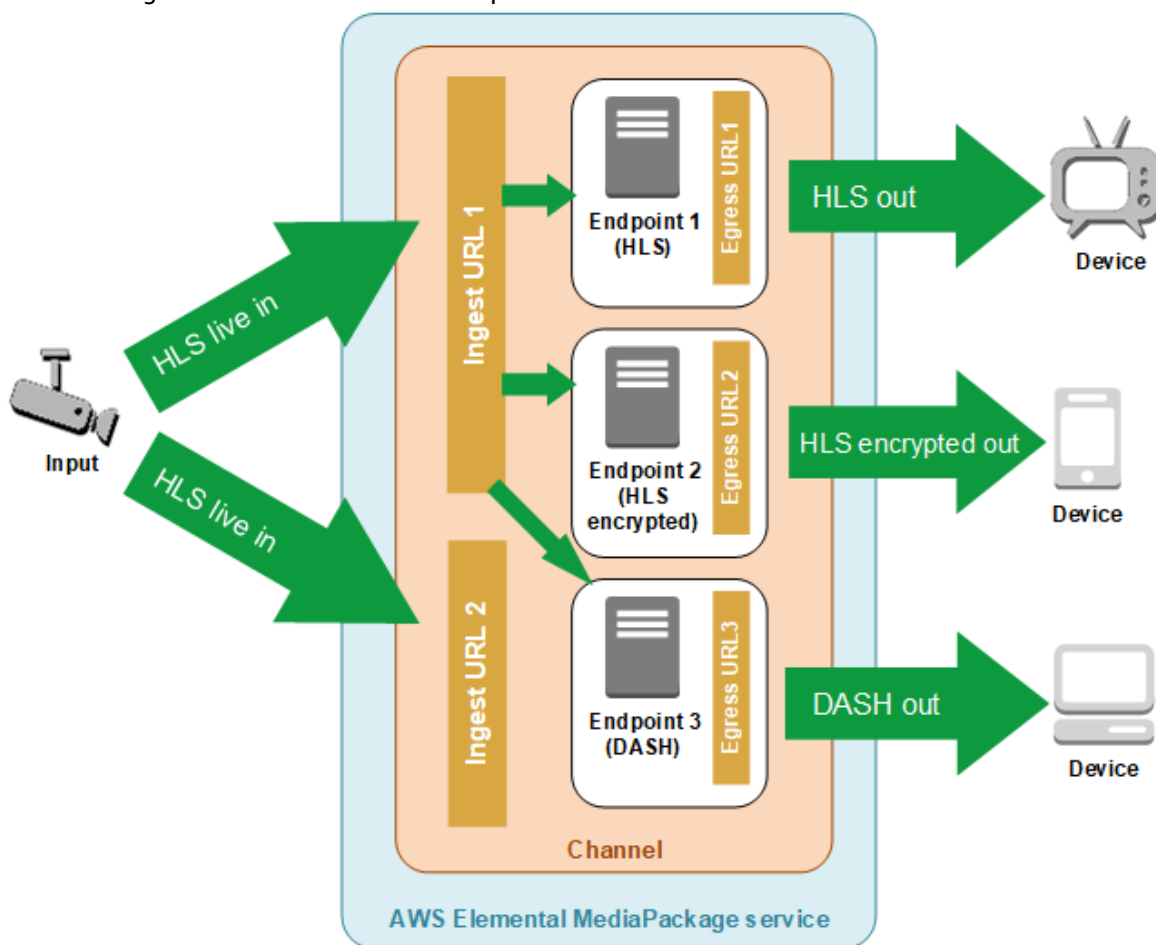
For more information, see [How Input Redundancy Works \(p. 3\)](#).

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 \(p. 50\)](#).

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.



How Input Redundancy Works

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 \(p. 14\)](#). 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 \(p. 18\)](#).

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 help enabling time delay, see [Creating an Endpoint \(p. 18\)](#).

3. You create an input and channel in AWS Elemental MediaLive, and you add two HLS output groups to the channel in MediaLive, as described in [Getting Started](#) in the *AWS Elemental MediaLive User Guide*. When you're setting up MediaLive, remember these important points:
 - The streams must have identical encoder settings for input redundancy to work correctly and playback to be maintained.
 - The input loss behavior on the channel in AWS Elemental MediaLive must not be set to emit a black frame 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 behavior in MediaLive, see [Complete the General Settings](#) in the *AWS Elemental MediaLive User Guide*.
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.

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

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 [How Input Redundancy Works \(p. 3\)](#).

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

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 \(p. 36\)](#).

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 spec to applicable playback devices.

Whitelisting

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 \(p. 22\)](#).

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 \(p. 2\)](#).

- **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 \(p. 7\)](#).

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 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 \(p. 7\)](#)
- [Creating an Admin IAM User \(p. 7\)](#)
- [Creating a Non-Admin IAM User \(p. 8\)](#)
- [\(Optional\) Setting Up Encryption \(p. 10\)](#)

Signing Up for AWS

If you do not have an AWS account, use the following procedure to create one.

To sign up for AWS

1. Open <https://aws.amazon.com/>, and then choose **Create an AWS Account**.

Note

If you previously signed in to the AWS Management Console using AWS account root user credentials, choose **Sign in to a different account**. If you previously signed in to the console using IAM credentials, choose **Sign-in using root account credentials**. Then choose **Create a new AWS account**.

2. Follow the online instructions.

Part of the sign-up procedure involves receiving a phone call and entering a verification code using 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 \(p. 8\)](#).

To create an IAM user for yourself and add the user to an Administrators group

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 of the console, choose **Users**, and then choose **Add user**.
3. For **User name**, type **Administrator**.
4. Select the check box next to **AWS Management Console access**, select **Custom password**, and then type the new user's password in the text box. You can optionally select **Require password reset** to force the user to create a new password the next time the user signs in.
5. Choose **Next: Permissions**.
6. On the **Set permissions** page, choose **Add user to group**.
7. Choose **Create group**.
8. In the **Create group** dialog box, for **Group name** type **Administrators**.
9. For **Filter policies**, select the check box for **AWS managed - job function**.
10. In the policy list, select the check box for **AdministratorAccess**. Then choose **Create group**.
11. 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.
12. Choose **Next: Tags** to add metadata to the user by attaching tags as key-value pairs.
13. 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 to restrict users' permissions to specific AWS resources, go to [Access Management](#) and [Example Policies](#).

For information about creating users with limited permissions, see [Creating a Non-Admin IAM User \(p. 8\)](#).

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:

```
{
  "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 \(p. 9\)](#).

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 \(CDNs\) \(p. 62\)](#).

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. Type the following JSON policy document:

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

6. Choose **Review policy**.

Note

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

7. On the **Review policy** page, type a **Name** and a **Description** (optional) 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.

(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 [Using Encryption](#) (p. 39).

Getting Started with 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 \(p. 11\)](#)
- [Step 1: Access AWS Elemental MediaPackage \(p. 11\)](#)
- [Step 2: Create a Channel \(p. 11\)](#)
- [Step 3: Create Endpoints \(p. 12\)](#)
- [\(Optional\) Step 4: Monitor AWS Elemental MediaPackage Activity \(p. 12\)](#)
- [Step 5: Clean Up \(p. 13\)](#)

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 \(p. 7\)](#), 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**.
2. For **ID**, type a name that describes the channel, such as **channelHLS1**. The ID is the primary identifier for the channel, and must be unique for your account in the region. Supported characters are letters, numbers, underscore (`_`), and dash (`-`). You cannot use spaces in the ID.
3. Keep the defaults for the remaining fields, and then choose **Create channel**.

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

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

AWS Elemental MediaPackage securely generates the user names and passwords when it creates the channel. If you need to change these credentials, see [Rotating Credentials on an Input URL \(p. 16\)](#).

Provide the information from these fields to the person in charge of the upstream encoder. In the stream configuration in the encoder, this person must type the destination as the input URL, and the WebDAV credentials as the channel's user name and password. The upstream encoder must push WebDAV over HTTPS to AWS Elemental MediaPackage, and include these credentials. If you're using input redundancy, the input streams to this channel must have identical encoder settings. For more information about setting up source streams for input redundancy, see [How Input Redundancy Works \(p. 3\)](#).

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**, type 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 region. Supported characters are letters, numbers, underscore (_), and dash (-). You cannot 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 type 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.

To view metrics using the CloudWatch console

Metrics are grouped first by the service namespace, and then by the various dimension combinations within each namespace.

1. 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. 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 Metrics \(p. 51\)](#).

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 to be deleted.
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 of 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.

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 \(p. 14\)](#)
- [Viewing Channel Details \(p. 15\)](#)
- [Editing a Channel \(p. 15\)](#)
- [Rotating Credentials on an Input URL \(p. 16\)](#)
- [Deleting a Channel \(p. 16\)](#)
- [Adding an Endpoint to a Channel \(p. 17\)](#)

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 [How Input Redundancy Works \(p. 3\)](#).

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 \(p. 66\)](#).

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 [How Input Redundancy Works \(p. 3\)](#).

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 \(p. 63\)](#).

Note

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

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

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

To edit a channel (console)

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

Rotating Credentials on an Input URL

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

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

To rotate credentials (console)

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

AWS Elemental MediaPackage displays the new credentials.

Deleting a Channel

Delete a channel to stop AWS Elemental MediaPackage from receiving further content. You must delete the channel's endpoints (as described in [Deleting an Endpoint \(p. 34\)](#)) 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 of 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 [Working with Endpoints \(p. 18\)](#).

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 \(p. 18\)](#)
- [Viewing All Endpoints Associated with a Channel \(p. 33\)](#)
- [Viewing a Single Endpoint \(p. 33\)](#)
- [Editing an Endpoint \(p. 33\)](#)
- [Deleting an Endpoint \(p. 34\)](#)
- [Previewing an Endpoint \(p. 34\)](#)

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 [How Input Redundancy Works \(p. 3\)](#).

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 \(p. 19\)](#)
- [Creating a Microsoft Smooth Streaming Endpoint \(p. 22\)](#)
- [Creating a Common Media Application Format \(CMAF\) Endpoint \(p. 25\)](#)
- [Creating a DASH Endpoint \(p. 29\)](#)

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 \(p. 15\)](#).
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 \(p. 19\)](#)
 - [Packager Settings Fields \(p. 20\)](#)
 - [Encryption Fields \(p. 20\)](#)
 - [Access Control Fields \(p. 22\)](#)
 - [Streams to Include Fields \(p. 22\)](#)
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 \(p. 66\)](#).

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 \(p. 36\)](#).
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 \(p. 41\)](#).
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 [Using Encryption \(p. 39\)](#).

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. **Resource ID** – Identifier that you define for the content, which is sent 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 following example shows a resource ID:

```
MovieNight20171126093045
```

- b. **System IDs** – 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. **URL** – The URL from the API Gateway proxy that you set up to talk to your key server. The API Gateway must reside in the same AWS Region as MediaPackage.

The following example shows a URL:

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

- d. **Role ARN** – 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.

- f. **Encryption method** – Choose **Sample-AES** for Apple HLS Fairplay or **AES-128** for Apple HLS AES-128.
- g. **(Optional) Constant initialization vector** – A 128-bit, 16-byte hex value represented by a 32-character string that is used with the key for encrypting content.
- h. **(Optional) Key rotation interval** – 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) Repeat EXT-X-KEY** – Boolean that indicates whether 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 choose **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. Choose this according to the needs of the player. Choosing this 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 \(p. 15\)](#).
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 \(p. 23\)](#)
 - [Packager Settings Fields \(p. 23\)](#)
 - [Encryption Fields \(p. 24\)](#)
 - [Access Control Fields \(p. 24\)](#)
 - [Streams to Include Fields \(p. 25\)](#)
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 \(p. 66\)](#).

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 \(p. 36\)](#).
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 [Using Encryption](#) (p. 39).

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. **Resource ID** – Identifier that you define for the content, which is sent 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 following example shows a resource ID:

```
MovieNight20171126093045
```

- b. **System IDs** – 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. **URL** – The URL from the API Gateway proxy that you set up to talk to your key server. The API Gateway must reside in the same AWS Region as MediaPackage.

The following example shows a URL:

```
https://lwm2dx1f33.execute-api.us-west-2.amazonaws.com/SpekeSample/copyProtection
```

- d. **Role ARN** – 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.

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 \(p. 15\)](#).
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 \(p. 26\)](#)
 - [Packager Settings Fields \(p. 26\)](#)
 - [HLS Manifest Fields \(p. 27\)](#)
 - [Encryption Fields \(p. 27\)](#)
 - [Access Control Fields \(p. 28\)](#)
 - [Streams to Include Fields \(p. 28\)](#)
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 \(p. 66\)](#).

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 \(p. 36\)](#).
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 [Using Encryption \(p. 39\)](#).

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. **Resource ID** – Identifier that you define for the content, which is sent 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 following example shows a resource ID:

```
MovieNight20171126093045
```

- b. **System IDs** – 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. **URL** – The URL from the API Gateway proxy that you set up to talk to your key server. The API Gateway must reside in the same AWS Region as MediaPackage.

The following example shows a URL:

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

- d. **Role ARN** – 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.

- f. **(Optional) Key rotation interval** – 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.

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 \(p. 15\)](#).
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 \(p. 31\)](#)
 - [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 \(p. 66\)](#).

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 \(p. 36\)](#).
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.
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 \(p. 46\)](#).

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**, 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.
 - If you choose **Time**, 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.

For more information about the variables used in the `SegmentTemplate` tag, see [DASH Manifest Segment Template Format \(p. 48\)](#).

- 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 \(p. 44\)](#).

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 [Using Encryption \(p. 39\)](#).

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. **Resource ID** – Identifier that you define for the content, which is sent 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 following example shows a resource ID:

```
MovieNight20171126093045
```

- b. **System IDs** – 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. **URL** – The URL from the API Gateway proxy that you set up to talk to your key server. The API Gateway must reside in the same AWS Region as MediaPackage.

The following example shows a URL:

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

- d. **Role ARN** – 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.

- f. **(Optional) Key rotation interval** – 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 \(p. 15\)](#).

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 \(p. 15\)](#).
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 \(p. 15\)](#).
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 \(p. 18\)](#).
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 \(p. 15\)](#).
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 \(p. 15\)](#).

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.

Time-shifted Viewing Reference 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 six hours of content.

For packager-specific rules about how you can notate the parameters, see [Rules for Start and End Parameters \(p. 37\)](#).

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 six 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. For example, there is no `EXT-X-ENDLIST` tag on HLS manifests. Otherwise, if the end time is in the past, the tags in the manifest are consistent with a video on demand (VOD) manifest.

- 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 6 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 \(p. 37\)](#)
- [HLS and CMAF Parameter Rules \(p. 37\)](#)
- [Microsoft Smooth Parameter Rules \(p. 38\)](#)

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&2018-04-04T02:15:00-08:00
```

- Path elements – start and end parameters are included in the path of the request URL

Example HLS

```
https://cf98fa7b2ee4450e.mediapackage.us-east-1.amazonaws.com/out/  
v1/064134724fd74667ba294657a674ae72/start/1513717228/end/1513720828/comedy.m3u8
```

Example CMAF

```
https://cf98fa7b2ee4450e.mediapackage.us-east-1.amazonaws.com/out/  
v1/064134724fd74667ba294657a674ae72/manifest_id/start/1522807213/end/1522800013/news.m3u8
```

Microsoft Smooth Parameter Rules

Start and end parameters in the URL request for Microsoft Smooth Streaming content can be included as path elements in the URL.

- Path elements – start and end parameters are included in the path of the request URL

Example

```
https://cf98fa7b2ee4450e.mediapackage.us-east-1.amazonaws.com/out/  
v1/1f76b3b4f94c44a485c0e4e560afe50e/start/1513717228/end/1513720828/drama.ism/Manifest
```

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.

Note

To encrypt content, you must have a DRM solution provider.

- For an overview, see <https://docs.aws.amazon.com/speke/latest/documentation/what-is-speke.html#services-architecture>.
- To get set up, see <https://docs.aws.amazon.com/speke/latest/documentation/customer-onboarding.html>.

Your DRM solution provider can help you get set up to use DRM encryption in AWS Elemental MediaPackage.

Using Encrypted Content Keys with DRM

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 [Working with Endpoints](#) (p. 18).

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 on CMAF endpoints.

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 \(p. 41\)](#)
- [When Not to Use Rendition Groups \(p. 41\)](#)

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:

- On CMAF endpoints, 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 output DASH manifests.

Default DASH manifest

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

```
<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 manifest:

- Separate the manifest into multiple periods, to allow ad breaks. See [DASH Manifest Options in AWS Elemental MediaPackage \(p. 44\)](#).
- Reduce the length of the manifest to make processing and playback more efficient. See [Compacted DASH Manifests \(p. 46\)](#).

- Control what segment information is used in the media URL in the `SegmentTemplate` properties. See [DASH Manifest Segment Template Format \(p. 48\)](#).

Multi-period DASH 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 \(p. 29\)](#).
- For more information about how multi-period DASH works in AWS Elemental MediaPackage, see the following *How it Works* section.

How it Works

To use the multi-period DASH feature, the input to AWS Elemental MediaPackage must have SCTE-35 ad marker messages. These messages inform MediaPackage of where to create period boundaries. This is how MediaPackage processes those messages:

1. AWS Elemental MediaPackage detects the SCTE-35 messages from the input source.
2. Using the attributes of the SCTE-35 messages, AWS Elemental MediaPackage calculates where the boundaries are between the end of the main content and the ads. This calculation is $(\text{scte35 ptsAdjustment} + \text{scte35 ptsTime}) / (\text{EventStream timescale})$.

Example

In the following example, the period starts at 44.075 seconds because $(183003 + 3783780) / 90000 = 44.075$:

```
<Period start="PT44.075S" id="21">
  <EventStream timescale="90000" schemeIdUri="urn:scte:scte35:2013:xml">
    <Event>
      <scte35:SpliceInfoSection protocolVersion="0" ptsAdjustment="183003" tier="4095">
        <scte35:SpliceInsert spliceEventId="1000" spliceEventCancelIndicator="false"
          outOfNetworkIndicator="true" spliceImmediateFlag="false" uniqueProgramId="7"
          availNum="1" availsExpected="4">
          <scte35:Program><scte35:SpliceTime ptsTime="3783780"/></scte35:Program>
        </scte35:SpliceInsert>
      </scte35:SpliceInfoSection>
    </Event>
  </EventStream>
  .
  .
</Period>
```

3. AWS Elemental MediaPackage inserts the `EventStream`, `Event`, and `scte35` tags with additional information into the manifest and surrounds the ad period with a `Period` tag, as shown in the preceding example. MediaPackage groups all adaptation sets before the first ad period into a period,

and any subsequent adaptation sets after the ad are grouped into a period, until the next SCTE-35 marker. Here is a complete manifest example with multiple periods. It uses SpliceInsert SCTE-35 ad markers:

Example

```
<?xml 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="963329" d="95232"/>
            <S t="1058561" d="96256" r="5"/>
            <S t="1636097" d="95232"/>
            <S t="1731329" 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" availsExpected="4">
            <scte35:Program><scte35:SpliceTime ptsTime="3783780"/></scte35:Program>
          </scte35:SpliceInsert>
        </scte35:SpliceInfoSection>
      </Event>
    </EventStream>
    <AdaptationSet mimeType="video/mp4" segmentAlignment="true"
      subsegmentAlignment="true" startWithSAP="1" subsegmentStartsWithSAP="1"
      bitstreamSwitching="true">
      <Representation id="1" width="960" height="540" frameRate="30000/1001"
        bandwidth="1000000" codecs="avc1.4D401F">
        <SegmentTemplate timescale="30000" media="index_video_1_0_${Number$.mp4?
m=1528413503" initialization="index_video_1_0_init.mp4?m=1528413503" startNumber="22"
        presentationTimeOffset="1322261">
          <SegmentTimeline>
            <S t="1322261" d="60060" r="13"/>
          </SegmentTimeline>
        </SegmentTemplate>
      </Representation>
    </AdaptationSet>
  </Period>
</MPD>
```

```
</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_$.mp4?
m=1528413503" initialization="index_audio_2_0_init.mp4?m=1528413503" startNumber="22"
presentationTimeOffset="2115617">
      <SegmentTimeline>
        <S t="2116353" d="96256"/>
        <S t="2212609" d="95232"/>
        <S t="2307841" d="96256" r="5"/>
        <S t="2885377" d="95232"/>
        <S t="2980609" d="96256" r="4"/>
      </SegmentTimeline>
    </SegmentTemplate>
  </Representation>
</AdaptationSet>
</Period>
</MPD>
```

If your input has TimeSignal SCTE-35 ad markers instead of SpliceInsert, the EventStream within the ad period looks like this:

```
<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">012345</
scte35:SegmentationUpid>
      </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 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:

```
<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 their elements are collapsed into one and moved to the `AdaptationSet`. The playback device understands that each representation in this adaptation set uses this same template:

```
<AdaptationSet mimeType="video/mp4" segmentAlignment="true" subsegmentAlignment="true"
startWithSAP="1" subsegmentStartsWithSAP="1" 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>
```

For information about compacting a DASH manifest, see [How AWS Elemental MediaPackage Compacts Manifests \(p. 48\)](#).

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 `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 in

the URL of the `media` attribute instead of `$Number$`. 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. 49).

How It Works

To use the `$Time$` variable, from the AWS Elemental MediaPackage console, choose **Time** for **Segment template format** on the DASH endpoint. AWS Elemental MediaPackage takes the following actions:

1. When AWS Elemental MediaPackage generates the DASH manifest, it uses the `$Time$` variable in the `media` value of the `SegmentTemplate` object, as shown in the following example:

Example

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

2. When a playback device requests segments, it uses the URL defined in the `media` attribute and replaces the variable with the availability start time of the segment that is requested.

Important

The value that replaces the variable must be an exact `t` value of a segment. If the request uses an arbitrary timestamp, AWS Elemental MediaPackage doesn't seek the closest segment.

Example

The following is an example of a segment template from a representation. It uses the `$Time$` variable.

```
<SegmentTemplate timescale="30000" media="155_video_1_2_-$Time$.mp4?m=1545421124"
initialization="155_video_1_2_init.mp4?m=1545421124" startNumber="710">
  <SegmentTimeline>
    <S t="255197799" d="360360" r="8"/>
    <S t="258441039" d="334334"/>
  </SegmentTimeline>
</SegmentTemplate>
```

The request URL for the first segment is `155_video_1_2_255197799.mp4`. With a 360360 duration, the next segment request is `155_video_1_2_255558159.mp4`, and so on through the ninth segment.

The final segment request is `155_video_1_2_258441039.mp4`.

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 \(p. 50\)](#)
- [Monitoring AWS Elemental MediaPackage with Amazon CloudWatch Events \(p. 56\)](#)
- [Logging AWS Elemental MediaPackage API Calls with AWS CloudTrail \(p. 59\)](#)

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 region, go to the **Channels** page.
 - For metrics on a specific channel and all of its endpoints, go to the channel's details page.
 - For metrics on a specific endpoint and its channel, go to the endpoint's details page.
3. (Optional) To refine the metrics view, choose **Open in CloudWatch**.

To view metrics using the CloudWatch console

Metrics are grouped first by the service namespace, and then by the various dimension combinations within each namespace.

1. Sign in to the AWS Management Console and open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Metrics**.
3. Under **All metrics**, choose the **AWS/MediaPackage** namespace.
4. Choose the metric dimension to view the metrics (for example, choose `channel` to view metrics per channel).

To view metrics using the AWS CLI

At a command prompt, use the following command:

```
aws cloudwatch list-metrics --namespace "AWS/MediaPackage"
```

AWS Elemental MediaPackage Metrics

The `AWS/MediaPackage` namespace includes the following metrics. AWS Elemental MediaPackage publishes metrics to CloudWatch every minute, if not sooner.

Metric	Description
<code>ActiveInput</code>	<p>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.</p> <p>Units: None</p> <p>Valid dimension:</p> <ul style="list-style-type: none"> • Combination of <code>Channel</code> and <code>OriginEndpoint</code>
<code>EgressBytes</code>	<p>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.</p> <p>Units: Bytes</p> <p>Valid statistics:</p> <ul style="list-style-type: none"> • <code>Average</code> – average bytes (<code>Sum/SampleCount</code>) that AWS Elemental MediaPackage outputs over the configured interval. • <code>Maximum</code> – largest individual output request (in bytes) made to AWS Elemental MediaPackage. • <code>Minimum</code> – smallest individual output request (in bytes) made to AWS Elemental MediaPackage. • <code>SampleCount</code> – number of requests that is used in the statistical calculation.

Metric	Description
	<ul style="list-style-type: none"> • Sum – total number of bytes that AWS Elemental MediaPackage outputs over the configured interval. <p>Valid dimensions:</p> <ul style="list-style-type: none"> • Channel • Combination of Channel and OriginEndpoint • No dimension
EgressRequestCount	<p>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</p> <p>Valid statistics:</p> <ul style="list-style-type: none"> • Sum – total number of output requests that AWS Elemental MediaPackage receives. <p>Valid dimensions:</p> <ul style="list-style-type: none"> • Channel • Combination of Channel and OriginEndpoint • StatusCodeRange • Combination of Channel and StatusCodeRange • Combination of Channel, OriginEndpoint, and StatusCodeRange • No dimension

Metric	Description
EgressResponseTime	<p>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.</p> <p>Units: Milliseconds</p> <p>Valid statistics:</p> <ul style="list-style-type: none">• Average – average amount of time (Sum/SampleCount) that it takes AWS Elemental MediaPackage to process output requests over the configured interval.• Maximum – longest amount of time (in milliseconds) that it takes AWS Elemental MediaPackage to process an output request and provide a response.• Minimum – shortest amount of time (in milliseconds) that it takes AWS Elemental MediaPackage to process an output request and provide a response.• SampleCount – number of requests that is used in the statistical calculation.• Sum – total amount of time that it takes AWS Elemental MediaPackage to process output requests over the configured interval. <p>Valid dimensions:</p> <ul style="list-style-type: none">• Channel• Combination of Channel and OriginEndpoint

Metric	Description
IngressBytes	<p>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.</p> <p>Units: Bytes</p> <p>Valid statistics:</p> <ul style="list-style-type: none">• Average – average bytes ($\text{Sum}/\text{SampleCount}$) that AWS Elemental MediaPackage receives over the configured interval.• Maximum – largest individual input request (in bytes) made to AWS Elemental MediaPackage.• Minimum – smallest individual input request (in bytes) made to AWS Elemental MediaPackage.• SampleCount – number of requests that is used in the statistical calculation.• Sum – total number of bytes that AWS Elemental MediaPackage receives over the configured interval. <p>Valid dimensions:</p> <ul style="list-style-type: none">• Channel• Combination of Channel and OriginEndpoint• No dimension

Metric	Description
IngressResponseTime	<p>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</p> <p>Valid statistics:</p> <ul style="list-style-type: none"> • 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. <p>Valid dimensions:</p> <ul style="list-style-type: none"> • Channel • Combination of Channel and OriginEndpoint • No dimension

AWS Elemental MediaPackage Dimensions

You can filter the `AWS/MediaPackage` data using the following dimensions.

Dimension	Description
No Dimension	Metrics are aggregated and shown for all channels, endpoints, or status codes.
Channel	<p>Metrics are shown only for the specified channel.</p> <p>Value: The autogenerated GUID of the channel.</p> <p>Can be used alone or with other dimensions:</p> <ul style="list-style-type: none"> • Alone to show metrics for only the specified channel.

Dimension	Description
	<ul style="list-style-type: none"> With the <code>originEndpoint</code> dimension to show metrics for the specified endpoint that is associated with the specified channel.
<code>OriginEndpoint</code>	<p>Metrics are shown for the specified channel and endpoint combination.</p> <p>Value: The autogenerated GUID of the endpoint.</p> <p>Must be used in conjunction with the <code>channel</code> dimension.</p>
<code>StatusCodeRange</code>	<p>Metrics are shown for the specified status code range.</p> <p>Value: <code>2xx</code>, <code>3xx</code>, <code>4xx</code>, or <code>5xx</code>.</p> <p>Can be used alone or with other dimensions:</p> <ul style="list-style-type: none"> Alone to show all output requests for the specified status range. With the <code>channel</code> dimension to show output requests for all endpoints that are associated with the specified channel, with the specified status code range. With the <code>channel</code> and <code>originEndpoint</code> dimensions to show output requests with a specific status code range on the specified endpoint that is associated with the specified channel.

Monitoring AWS Elemental MediaPackage with Amazon CloudWatch Events

Amazon CloudWatch Events enable 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 Amazon EC2 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](#).

For a list of events that AWS Elemental MediaPackage emits, see [AWS Elemental MediaPackage Events](#) (p. 57).

AWS Elemental MediaPackage Events

AWS Elemental MediaPackage integrates with Amazon CloudWatch Events to notify you of certain events that affect your channels and endpoints. Each event is represented in [JSON \(JavaScript Object Notation\)](#) and contains the event name, the date and time when the event occurred, the channel or endpoint affected, and more. 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. 57)
- [Key Provider Notification Events](#) (p. 58)

Input Notification Events

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

- 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. 66).
- 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 [How Input Redundancy Works](#) (p. 3).

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/262ff182d46d4b399fcabea1364df682"
  ],
  "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)"
  }
}
```

Example Input Switch Event

```
{
  "id": "8f9b8e72-0b31-e883-f19c-aec84742f3ce",
  "detail-type": "MediaPackage Input Notification",
  "source": "aws.mediapackage",
  "account": "aws_account_id",
  "time": "2018-07-16T17:29:36Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:mediapackage:us-
east-1:aws_account_id:origin_endpoints/82d6b9bc04cb4612b487963d6c8d0f1a"
  ],
  "detail": {
    "event": "InputSwitchEvent",
    "message": "Origin endpoint experienced an Input Switch Event"
  }
}
```

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:

```
{
  "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 \(p. 57\)](#) and [Key Provider Notification Events \(p. 58\)](#).

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 save it.

Logging AWS Elemental MediaPackage API Calls with AWS CloudTrail

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:

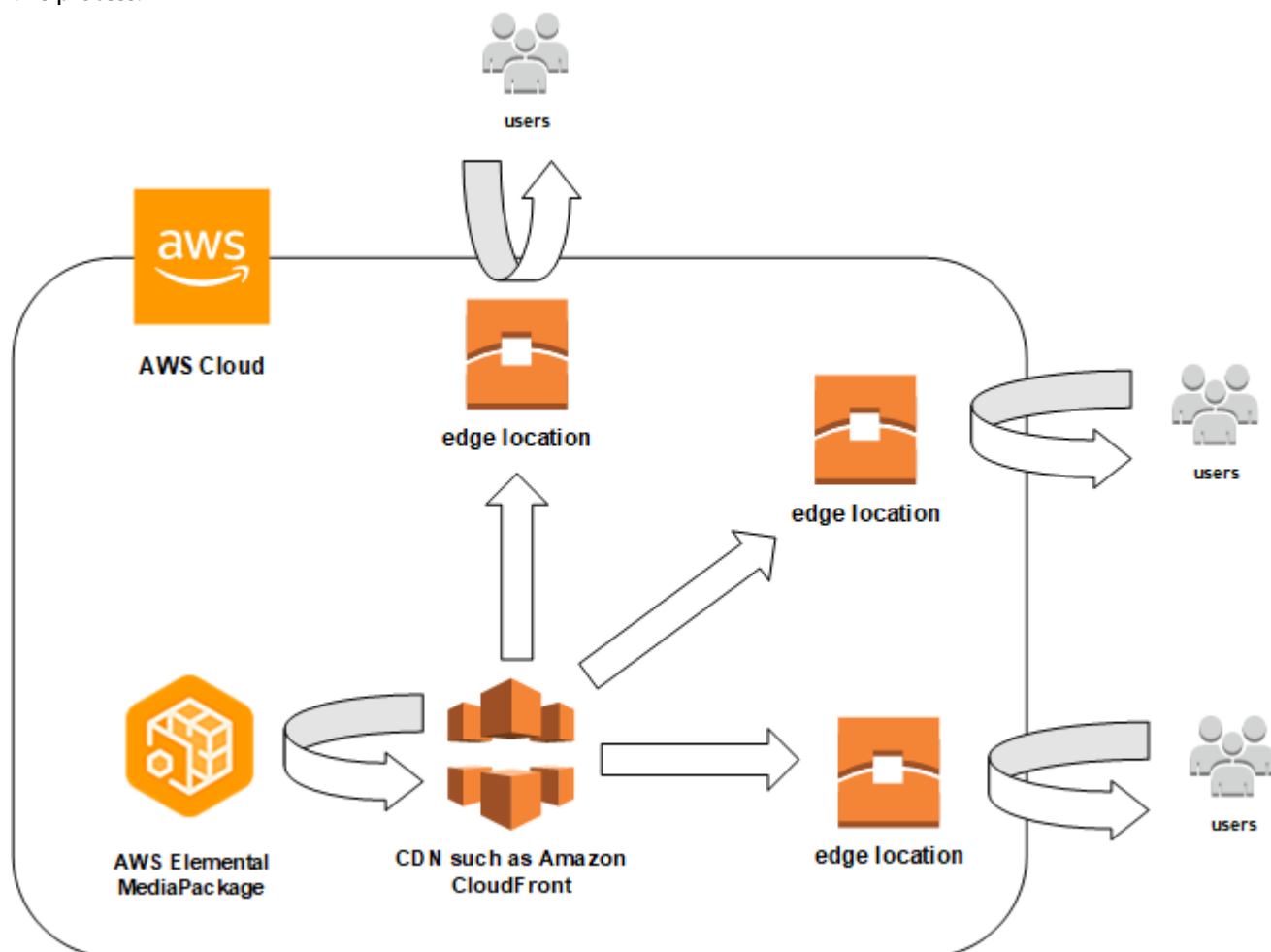
```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "ABCDEFGHIJKL123456789",
    "arn": "arn:aws:sts::444455556666:assumed-role/Admin/testUser",
    "accountId": "444455556666",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2018-12-18T00:50:58Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "ABCDEFGHIJKL123456789",
        "arn": "arn:aws:iam::444455556666:role/Admin",
        "accountId": "444455556666",
        "userName": "Admin"
      }
    }
  },
  "eventTime": "2018-12-18T00:50:59Z",
  "eventSource": "mediapackage.amazonaws.com",
  "eventName": "UpdateChannel",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.17",
```


AWS Elemental MediaPackage User Guide
Understanding AWS Elemental
MediaPackage Log File Entries

```
"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"
      }
    ]
  },
  "id": "cloudtrail-test",
  "arn": "arn:aws:mediapackage:us-
west-2:444455556666:channels/8d0ca97840d94b18b37ad292c131bcad"
},
"requestID": "fc158262-025e-11e9-8360-6bff705fbba5",
"eventID": "e9016b49-9a0a-4256-b684-eed9bd9073ab",
"readOnly": false,
"eventType": "AwsApiCall",
"recipientAccountId": "444455556666"
}
```

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. 63)
- [Viewing a Distribution](#) (p. 64)
- [Editing a Distribution](#) (p. 64)

- [Deleting a Distribution \(p. 64\)](#)

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 console, or manually from the CloudFront console. The following sections describe each approach.

Topics

- [Creating a Distribution from AWS Elemental MediaPackage \(p. 63\)](#)
- [Creating a Distribution from Amazon CloudFront \(p. 64\)](#)

Creating a Distribution from AWS Elemental MediaPackage

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 \(p. 9\)](#).

To create a distribution from the AWS Elemental MediaPackage console

1. Start a new channel as described in [the section called "Creating a Channel" \(p. 14\)](#), 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 \(p. 15\)](#).

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 \(p. 18\)](#). 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 \(p. 15\)](#), 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. For information about requesting an increase to soft limits, see [AWS Service Limits](#). Hard limits cannot be changed.

Soft Limits

The following table describes limits in AWS Elemental MediaPackage that can be increased. For information about changing limits, see [AWS Service Limits](#).

For some customers, your account limit might be below these advertised 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.

Resource	Default Limit
Maximum Channels	<p>10</p> <p>Note Increasing your channel limit doesn't always mean that you also need to increase your endpoints. For example, if you need 14 channels and want to serve HLS, HLS encrypted, and DASH content from each channel, you need only three endpoints for each channel (one for each output type). The default endpoint limit is 10 so, although you do need a channel limit increase, you don't need to increase your endpoint limit. You won't exceed the limit of 10 endpoints <i>per channel</i>.</p>
Maximum Endpoints per Channel	<p>10</p> <p>This is a <i>per channel</i> limit. Each endpoint represents the output package that you use. If one channel serves HLS, HLS encrypted, DASH, DASH encrypted, Microsoft Smooth, and Microsoft Smooth encrypted content, then that channel has six endpoints and falls within the 10 endpoints limit. If you have 10 channels set up this same way, then you still haven't exceeded the limit because each channel uses only 6 endpoints.</p>

Hard Limits

The following table describes limits within AWS Elemental MediaPackage that can't be increased.

Resource or Operation	Limit
Input Stream Limits	20 streams per channel, and 10 tracks per stream
Maximum Content Retention	336 hours (14 days)
Maximum Live Manifest Length	5 minutes
Maximum Time-shifted Manifest Length	6 hours
Request Rates per Channel	<ul style="list-style-type: none">• Input: 50 requests per second• Output: 200 requests per second
REST API Requests	<ul style="list-style-type: none">• Steady state: 5 requests per second• Bursting: 50 requests per second

AWS Elemental MediaPackage Related Information

The following table lists related resources that you'll find useful as you work with AWS Elemental MediaPackage.

Resource	Description
Classes and Workshops	Links to role-based and specialty courses as well as self-paced labs to help sharpen your AWS skills and gain practical experience.
AWS Developer Tools	Links to developer tools, SDKs, IDE tool kits, and command line tools for developing and managing AWS applications.
AWS Whitepapers	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.
AWS Support Center	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.
AWS Support	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.
Contact Us	A central contact point for inquiries concerning AWS billing, account, events, abuse, and other issues.
AWS Site Terms	Detailed information about our copyright and trademark; your account, license, and site access; and other topics.

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:** February 6, 2019

update-history-change	update-history-description	update-history-date
Added DASH manifest treatment information. (p. 43)	Added <i>DASH Manifest Options</i> topic to discuss the ways that you can modify output DASH manifests.	February 6, 2019
Added AWS CloudTrail logging information. (p. 59)	Added <i>Logging AWS Elemental MediaPackage API Calls with AWS CloudTrail</i> topic to discuss using CloudTrail to log actions in the AWS Elemental MediaPackage API.	December 21, 2018
Added information about compact DASH manifests. (p. 46)	Added a <i>Compacted DASH Manifests</i> topic to discuss how compacting DASH output manifests works in AWS Elemental MediaPackage.	December 18, 2018
Added information about DASH manifest SegmentTemplate format options. (p. 48)	Added the <i>DASH Manifest Segment Template Format</i> topic to discuss how to change variables in the media URL in the SegmentTemplate element of the DASH manifest.	December 18, 2018
Updated content retention window limit. (p. 66)	AWS Elemental MediaPackage now retains content for 336 hours (14 days).	November 13, 2018
Added content key encryption to DRM encryption (p. 39)	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.	November 8, 2018
Added input redundancy information. (p. 3)	Added <i>How Input Redundancy Works</i> topic to discuss how AWS Elemental MediaPackage can	August 28, 2018

	receive two identical streams for back-up purposes. .	
Added Amazon CloudFront console integration information. (p. 62)	Added sections about working with distributions in CloudFront, including how to create a distribution from the AWS Elemental MediaPackage console.	August 3, 2018
Added information about multi-period DASH. (p. 44)	Added <i>Multi-period DASH in AWS Elemental MediaPackage</i> topic to discuss the purpose and functionality of multiple periods in DASH manifests.	July 18, 2018
Added content delivery network (CDN) information. (p. 62)	Added <i>Working with CDNs</i> topic to discuss how AWS Elemental MediaPackage works with CDNs such as Amazon CloudFront.	May 31, 2018
Added information about creating event notifications. (p. 56)	Added <i>Creating Event Notifications</i> topic to discuss using Amazon CloudWatch Events and Amazon Simple Notification Service to notify you of new events.	January 22, 2018

Earlier Updates

The following table describes important changes in each release of the *AWS Elemental MediaPackage User Guide* before May 2018.

Change	Description	Date
Initial document creation.	New document.	November 27, 2017
Corrected links and added whitelisting.	Corrected links to the AWS Elemental MediaPackage console and AWS Elemental MediaPackage API Reference. In <i>Working with Endpoints</i> , added reference to access control fields.	December 1, 2017
Added IAM policy information specific to AWS Elemental MediaPackage.	In Setting Up AWS Elemental MediaPackage (p. 7) , added instructions for creating non-admin roles with limited permissions.	December 13, 2017
Added hard limit information.	In Limits in AWS Elemental MediaPackage (p. 66) , added information about limits that can't be changed (hard limits).	December 20, 2017

Change	Description	Date
Updated IAM policy information.	In Setting Up AWS Elemental MediaPackage (p. 7) , added information about policies specific to AWS Elemental MediaPackage.	January 5, 2018
Added Amazon CloudWatch Events information.	Added the section called "Monitoring with CloudWatch Events" (p. 56) section about the CloudWatch Events that AWS Elemental MediaPackage supports.	February 14, 2018
Added CMAF endpoint information.	Added Creating a Common Media Application Format (CMAF) Endpoint (p. 25) section for new output type.	April 6, 2018
Updated feature functionality.	In Features of AWS Elemental MediaPackage (p. 4) , added feature support for HDR-10.	April 30, 2018
Added content delivery network (CDN) information.	Added topic Working with Content Delivery Networks (CDNs) (p. 62) to discuss how AWS Elemental MediaPackage works with CDNs such as Amazon CloudFront.	May 31, 2018

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*.