Implementation Guide

Live Streaming on AWS

Live Streaming on AWS: Implementation Guide

Table of Contents

Build highly available live video streaming content using AWS Media Services and Amazon CloudFront

Publication date: *November 2016 (last update: October 2023)*

Amazon Web Services (AWS) lets broadcasters and content owners to seamlessly scale infrastructure to broadcast live content to a global audience. The Live Streaming on AWS solution helps you build highly available live video streaming content using AWS Media Services and Amazon CloudFront that is highly resilient and secure to deliver real-time viewing experiences to your customers.

This solution provides the following features:

- Encodes and packages your content for adaptive bitrate streaming across multiple screens via HTTP live streaming (HLS), Dynamic Adaptive Streaming over HTTP (*DASH*), and Common Media Application Format (CMAF) by automatically configuring AWS Elemental MediaLive and AWS Elemental MediaPackage.
- Provides an elastic, highly available, global content delivery network for live video streaming using Amazon CloudFront.

With this solution, you can run it only during a live event and then after the program ends, delete the solution's stack to ensure you only pay for the infrastructure that you use.

This implementation guide discusses architectural considerations and configuration steps for deploying Live Streaming on AWS in the AWS Cloud. It includes a link to an AWS CloudFormation template that launches and configures the AWS services required to deploy this solution using AWS best practices for security and availability.

The guide is intended for IT infrastructure architects, administrators, and DevOps professionals who have practical experience with video streaming and architecting in the AWS Cloud.

Use this navigation table to quickly find answers to these questions:

If you want to . . .	Read . . .
Know the cost for running this solution.	Cost

If you want to . . .	Read . . .
The estimated cost for running this solution in the US East (N. Virginia) Region is USD $69.74 per month.	
Understand the security considerations for this solution.	Security
Know how to plan for quotas for this solution.	Quotas
Know which AWS Regions are supported for this solution.	Supported AWS Regions
View or download the AWS CloudFormation template included in this solution to Automatically deploy the infrastructure resources (the "stack") for this solution.	AWS CloudFormation template

Features and benefits

The Live Streaming on AWS solution provides the following features:

Comprehensive output formats

Using AWS Elemental MediaPackage, this solution supports the standards and formats commonly used to stream video, such as CMAF, HLS, and DASH, for playback support on different media players.

Input redundancy

Using AWS Elemental MediaLive, this solution supports two input feeds and it's ideal for customers looking to add redundancy to their live feeds.

MediaConnect support

The solution supports AWS Elemental MediaConnect inputs providing a high-quality transport service for live video.

Flexible video content protection

Using this solution, you can apply just-in-time content protection to secure your live streams by integrating with multiple Digital Rights Management (DRM) technologies. Protection capabilities are standards-based, including support for Apple FairPlay, Widevine, and Microsoft PlayReady using AES-128 encryption.

Integration with Service Catalog AppRegistry and AWS Systems Manager Application Manager

This solution includes a Service Catalog AppRegistry resource to register the solution's CloudFormation template and its underlying resources as an application in both Service Catalog AppRegistry and AWS Systems Manager Application Manager. With this integration, you can centrally manage the solution's resources.

Use cases

Streaming media

As consumer demand for video streaming increases, media and entertainment companies are looking for secure and reliable web-based video streaming alternatives to traditional television. With Live Streaming on AWS, customers can avoid inefficient trial-and-error approaches and save time and costs for their streaming media projects.

Concepts and definitions

This section describes key concepts and defines terminology specific to this solution:

Adaptive Bit Rate (ABR)

A streaming method that adjusts the video quality based on network conditions to improve video streaming over HTTP networks.

HTTP Live Streaming (HLS)

HTTP-based streaming protocol to deliver media over the internet and developed by Apple Inc.

Dynamic Adaptive Streaming over HTTP (DASH)

HTTP-based streaming protocol (also known as MPEG-DASH) to deliver media over the internet and developed under MPEG (Motion Picture Experts Group).

Common Media Application Format (CMAF)

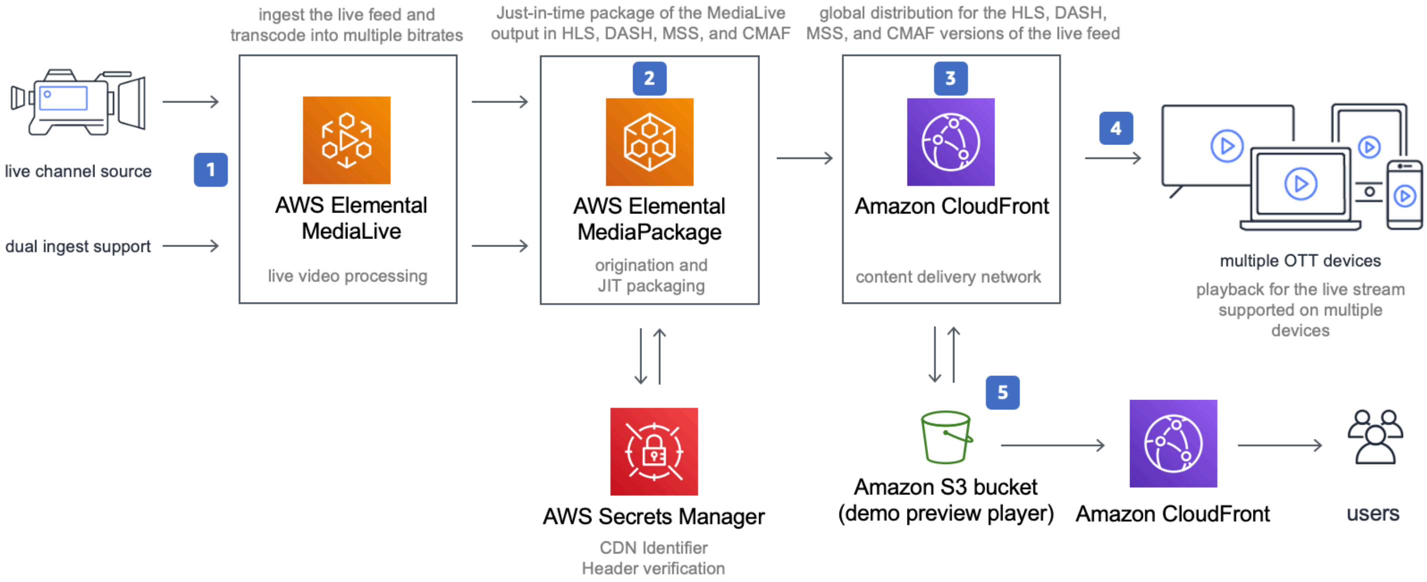
HTTP-based streaming and packaging standard to improve delivery of media over the internet, compatible with HLS and DASH, and co-developed by Apple and Microsoft. For a general reference of AWS terms, see the AWS glossary in the *AWS General Reference*.

Architecture overview

This section provides a reference implementation architecture diagram for the components deployed with this solution.

Architecture diagram

Deploying this solution with the default parameters deploys the following components in your AWS account.


ingest the live feed and transcode into multiple bitrates
Just-in-time package of the MediaLive output in HLS, DASH, MSS, and CMAF
global distribution for the HLS, DASH, MSS, and CMAF versions of the live feed
live channel source
1
dual ingest support
AWS Elemental MediaLive
live video processing
2
AWS Elemental MediaPackage
origination and JIT packaging
3
Amazon CloudFront
content delivery network
4
multiple OTT devices
playback for the live stream supported on multiple devices
AWS Secrets Manager
CDN Identifier Header verification
5
Amazon S3 bucket (demo preview player)
Amazon CloudFront
users

Live Streaming on AWS solution architecture

The solution's AWS CloudFormation template launches the AWS products and services necessary to ingest, transcode, and deliver live streaming video.

1. AWS Elemental MediaLive ingests two input feeds and transcodes your content into two adaptive bitrate (ABR) HTTP Live Streaming (HLS) streams as output.

   Two feeds are ingested for MediaLive redundancy. Each MediaLive transcodes a single ingest feed into ABR outputs.
2. AWS Elemental MediaPackage ingests the MediaLive ABR output and packages the live stream into HTTP Live Streaming (HLS), Dynamic Adaptive Streaming over HTTP (DASH) and Common Media Application Format (CMAF) formats that are delivered to three MediaPackage custom endpoints.

These three formats were selected because that is what popular streaming devices support. MediaPackage is an origin server that prepares the content for delivery to phones, tablets, smart TVs, and other devices.

3. An Amazon CloudFront distribution is configured to use the MediaPackage custom endpoints as its origin and includes a CDN Identifier custom HTTP header to authenticate requests. MediaPackage only fulfills playback requests that are authorized between MediaPackage and CloudFront using the CDN Identifier. This CDN Identifier is created as part of the CloudFormation deployment and securely stored in AWS Secrets Manager. For details, refer to the CDN Authorization in AWS Elemental MediaPackage in the *AWS Elemental MediaPackage User Guide*.
4. The CloudFront distribution delivers your live stream to viewers with low latency and high transfer speeds.
5. A demo HTML preview player is available to help you test the solution. The player is a static website hosted in an Amazon Simple Storage Service (Amazon S3) bucket. Amazon CloudFront is used to restrict access to the solution's website bucket contents.

You can configure this solution to ingest Real-time Transport Protocol (RTP), Real-Time Messaging Protocol (RTMP), HTTP Live Streaming (HLS), or AWS Elemental MediaConnect flows. This solution also includes three encoding profiles.

Note

AWS CloudFormation resources are created from AWS Cloud Development Kit (AWS CDK) constructs.

AWS Well-Architected

We designed this solution with best practices from the AWS Well-Architected Framework, which helps customers design and operate reliable, secure, efficient, and cost-effective workloads in the cloud.

This section describes how we applied the design principles and best practices of the Well-Architected Framework when building this solution.

Operational excellence

This section describes how the principles and best practices of the operational excellence pillar were applied when designing this solution.

The Live Streaming on AWS solution tracks all assets via AWS CloudTrail, and logs from AWS Elemental MediaLive, MediaPackage, and Amazon CloudFront provide observability into the infrastructure and the rest of the solution components.

Security

This section describes how the principles and best practices of the security pillar were applied when designing this solution. To help reduce latency and improve security, Live Streaming on AWS includes an Amazon CloudFront distribution with an origin access identity, which is a special CloudFront user that helps restrict access to the solution's website bucket contents.

AWS Elemental MediaPackage only fulfills playback requests that are authorized between MediaPackage and CloudFront using a CDN identifier. This CDN identifier is created as part of the AWS CloudFormation deployment and securely stored in AWS Secrets Manager. For more details, refer to the CDN authorization in AWS Elemental MediaPackage in the *AWS Elemental MediaPackage User Guide*.

Reliability

This section describes how the principles and best practices of the reliability pillar were applied when designing this solution.

The solution supports primary and secondary live streams throughout AWS Elemental MediaLive and AWS Elemental MediaPackage providing the redundancy that customers require for critical live video events.

Performance efficiency

This section describes how the principles and best practices of the performance efficiency pillar were applied when designing this solution.

This solution uses AWS Elemental MediaLive, MediaPackage, and MediaConnect, which are currently available in specific AWS Regions only. If you use MediaConnect as input, you must deploy this solution in the same Region as your MediaConnect flows.

The Live Streaming on AWS solution is automatically tested and reviewed by solution architects and subject matter experts for areas to experiment and improve.

Cost optimization

This section describes how the principles and best practices of the cost optimization pillar were applied when designing this solution.

The cost for running this solution varies based on a number of factors, including the encoded profile selected, the bitrate of the live stream and the number of viewers.

Using Application Manager, customers can measure the efficiency of the workloads, and the costs associated with delivery.

Sustainability

This section describes how the principles and best practices of the sustainability pillar were applied when designing this solution. To minimize the environmental impact of backend services, Live Streaming on AWS uses managed and serverless services. Customers can run this solution only during a live event and delete the stack after the program ends, thereby reducing the carbon footprint as compared to the footprint of continually operating on-premises servers.

Architecture details

This section describes the components and AWS services that make up this solution and the architecture details on how these components work together.

HTML preview player

A demo HTML preview player is available to help you test the solution. The player is a static website hosted in an Amazon S3 bucket. It is pre-populated with the URLs that point to the newly created customer stream. The HTML/JavaScript application plays back the HLS, DASH, MSS, and CMAF streams. In addition, the solution can be configured to ingest a Demo HLS feed hosted on AWS. Customize the HTML in the `DemoBucket` S3 bucket to suit your needs. For details about putting the HLS video into a webpage, refer to the Apache 2.0 Video.JS open-source project.

To check output on the player, start the AWS Elemental MediaLive channel. Even if you selected no for the **Start MediaLive Channel** CloudFormation template parameter, you can go to the MediaLive channel to start the player when you are ready to start testing.

URL_PULL (HLS) input configuration

`URL_PULL` provides the option to ingest an HTTP live streaming (HLS) stream over HTTP or HTTPS. The following parameters are required to configure the solution to ingest an HLS stream:

Source Input Type - `URL_PULL`.

Primary Source URL - The HTTP(s) link to the HLS stream manifest file. The default value is a demo stream from AWS.

Primary Source Username - Only required if you have basic authentication setup on your source HLS stream.

Primary Source Password - Only required if you have basic authentication setup on your source HLS stream.

Secondary Source URL - The HTTP(s) link to the HLS stream manifest file. The default value is a demo stream from AWS.

Secondary Source Username - Only required if you have basic authentication setup on your source HLS stream.

Secondary Source Password - Only required if you have basic authentication setup on your source HLS stream.

Encoding Profile - Select the profile that best matches your source resolution.

Start MediaLive Channel - If your device is ready to stream, select `yes`. Otherwise, select no — you can start the AWS Elemental MediaLive channel through the AWS Management Console when you're ready to stream.

> **Note**
>
> For a full list of input types and configuration details, refer to the Creating an input topic in the *AWS Elemental MediaLive User Guide*.

RTMP_PULL input configuration

`RTMP_PULL` provides the option to ingest an RTMP stream. The following parameters are required to configure the solution to ingest an RTMP stream:

Source Input Type - `RTMP_PULL`.

Primary Source URL - The RTMP link to the primary source stream, for example `rtmp://203.0.113.20:1935/primary`.

Primary Source Username - Only required if you have basic authentication setup on your source stream.

Primary Source Password - Only required if you have basic authentication setup on your source stream.

Secondary Source URL - The RTMP link to the primary source stream, for example `rtmp://203.0.113.20:1935/secondary`.

Secondary Source Username - Only required if you have basic authentication setup on your source stream.

Secondary Source Password - Only required if you have basic authentication setup on your source stream.

Encoding Profile - Select the profile that best matches your source resolution.

Start MediaLive Channel - If your device is ready to stream, select `yes`. Otherwise, select no — you can start the AWS Elemental MediaLive channel through the AWS Management Console when you're ready to stream.

> **Note**
>
> For a full list of input types and configuration details, refer to the Creating an input topic in the *AWS Elemental MediaLive User Guide*.

RTMP_PUSH and RTP_PUSH input configuration

`RTMP_PUSH` and `RTP_PUSH` provide the option to push a transport stream (TS) to AWS Elemental MediaLive. In both options, the following parameters are required to configure the solution:

Source Input Type - `RTP_PUSH` or `RTMP_PUSH`

Input Security Group CIDR Block - A valid CIDR block used to create a security group to restrict access to the MediaLive input.

Encoding Profile - Select the profile that best matches your source resolution.

Start MediaLive Channel - If your device is ready to stream, select `yes`. Otherwise, select no — you can start the MediaLive channel through the AWS Management Console when you're ready to stream.

> **Note**
>
> For a full list of input types and configuration details, refer to the Creating an input topic in the *AWS Elemental MediaLive User Guide*.

MEDIACONNECT input configuration

`MEDIACONNECT` provides the option to ingest a stream from Elemental MediaConnect. The following parameters are required to configure the solution to ingest from MediaConnect:

Source Input Type - `MEDIACONNECT`.

Primary MediaConnect ARN - The ARN of the primary source stream, for example: `arn:aws:mediaconnect:uswest1:111122223333:flow:1bgf67:primary`. This MediaConnect flow must be in a different Availability Zone as the secondary stream.

Secondary MediaConnect ARN - The ARN of the secondary source stream, for example: `arn:aws:mediaconnect:uswest1:111122223333:flow:1bgf67:secondary`. This MediaConnect flow must be in a different Availability Zone as the primary stream.

Encoding Profile - Select the profile that best matches your source resolution.

Start MediaLive Channel - If your device is ready to stream, select `yes`. Otherwise, select no — you can start the AWS Elemental MediaLive channel through the AWS Management Console when you're ready to stream.

> **Note**
>
> For a full list of input types and configuration details, refer to the Creating an input topic in the *AWS Elemental MediaLive User Guide*.

AWS services in this solution

AWS service	Description
Amazon Elemental MediaLive	**Core.** It ingests two input feeds and transcode s your content into two adaptive bitrate (ABR) HTTP Live Streaming (HLS) streams as outputs.
Amazon Elemental Mediapackage	**Core.** It ingests the MediaLive ABR output and packages the live stream into HLS, Dynamic Adaptive Streaming over HTTP (DASH), and Common Media Application Format (CMAF) formats that are delivered to three MediaPack age custom endpoints.
Amazon CloudFront	**Core.** It uses the MediaPackage custom endpoints and its origin and includes a CDN

AWS service	Description
	identifier custom HTTP header to authenticate requests.
[Amazon Secrets Manager]()	**Core.** It stores securely the CDN identifier from Amazon CloudFront.
[Amazon S3]()	**Core.** It stores a demo HTML preview player to help you test the solution.
[AWS Systems Manager]()	**Supporting.** Provides application-level resource monitoring and visualization of resource operations and cost data.

Plan your deployment

This section describes the cost, Security, Regions, and quota considerations for planning your deployment.

Cost

You are responsible for the cost of the AWS services used while running this solution. As of this revision, the cost for running this solution in the US East (N. Virginia) with:

- Approximately 1,000 viewers for a one-hour live event using a standard definition (SD)-540p encoding profile is approximately **$2.50** for live encoding and packaging + **$67.24** for 791GB distribution = **$69.74 for the one-hour event**.
- Approximately 10,000 viewers for a one-hour live event using a high definition (HD)-1080p encoding profile is approximately **$13.04** for live encoding and packaging + **$1,492.56** for 18,017GB distribution = **$1,505.60 for the one-hour event**.
- Approximately 100 viewers for 200 hours live events each month using a high definition (HD)-1080p encoding profile is approximately **$904.69** for live encoding and packaging + **$2,934.00** for 36,035 GB distribution = **$3,838.69 for the total of 200 hours each month.**

These cost estimates depend on many factors, which are detailed in the following cost examples.

> **Note**
>
> The examples provided are likely higher than the actual costs of running this solution. The intent was to provide a guide to the pricing that is easily understood. Where assumptions were needed, we used factors that were straightforward to calculate and also likely be more expensive than the actual cost.

For an additional cost example for streaming a live event, refer to the FAQs about live streaming on AWS blog in the *AWS Media Blog*.

We recommend creating a budget through AWS Cost Explorer to help manage costs. Prices are subject to change. For full details, refer to the pricing webpage for each AWS service used in this solution.

Cost example 1

Cost example 1 covers a use case of approximately 1,000 viewers viewing a live event for about one hour with a **SD-540p** encoding profile selected in the AWS CloudFormation template. This cost example is based on the following factors:

- Pricing Region: US-East-1, assuming standard pricing (no free-tier or discounts).
- Viewers consume the highest bitrate: Note that bitrate consumption is a mix of all the streams, but the highest bitrate is used in the calculation to show the upper cost range. Additionally, the Quality-Defined Variable Bitrate (QVBR) and variable video complexity can result in an output bandwidth that is 10-50% lower in price than the estimate provided in Table 1.
- 99% cache/hit ratio between the content delivery network (CDN) and AWS Elemental MediaPackage.

> **Note**
>
> Storage of the test player is not included in this cost estimate.

Table 1 summarizes the total pricing for the live streaming event. Tables 2 through 4 break down the cost for each AWS service.

Table 1: Cost breakdown for 1,000 viewers for a one-hour live event

AWS service	Function	Cost per hour [USD]
AWS Elemental MediaLive	Input and outputs for channel	$1.99
AWS Elemental MediaPackage	Ingest of channel	$0.11
	Packaging and origination	$0.40
Amazon CloudFront	Distribution	$67.24
	Total:	**$69.74/hour**

AWS Elemental MediaLive pricing

Table 2 breaks down the AWS Elemental MediaLive pricing which assumes HD AVC input and SD AVC outputs with less than 10 Mbps bitrate and less than 30 frames per second (fps) frame rate.

Table 2: MediaLive pricing

Input / Output	Cost per hour [USD]
HD Input (AVC 10-20Mbps)	$0.234
Output 1 SD (512x288, 400k)	$0.44
Output 2 SD (640x360, 800k)	$0.44
Output 3 SD (768x432, 1200k)	$0.44
Output 4 SD (960x540, 1800k)	$0.44
Total:	**$1.99/hour**

AWS Elemental MediaPackage pricing

MediaPackage charges $0.03 per Gigabyte (GB) per hour for standard Live ingest, based on the aggregate bitrate of all live input streams. This example has four input streams, so the rate of GB streaming per hour is determined by:

1. Adding the bitrate of all streams in either kbps or Mbps:
   - 400k + 800k + 1200k + 1800k = 4200 kbps
   - .4 + .8 + 1.2 + 1.8 = 4.2 Mbps
2. Convert kbps or Mbps to Gigabits (Gbit) per second (ps). 1024 Mb = 1 Gbit:
   - 4.2 Mbps / 1024 = 0.0041015625 Gbits ps
3. Convert Gbits to GB. 1 Gbit = 0.125 (or 1/8) GB.
   - 0.0041015625 Gbits ps * 0.125 = 0.0005126953125 GBps
4. Convert GB per second to GB per hour:
   - 0.0005126953125 GBps * 60s * 60mins = 1.845703125 or 1.85 GB/hour

((sum of stream bitrates in Kbps) / 1024 (Mbps conversion)) / 1024 (Gbit conversion) * .125 (bits to Bytes conversion) * 60 seconds * 60 minutes = GB / hour

Table 3: MediaPackage ingest pricing

Input / Output	kbps
Output 1 SD (512x288, 400k)	400
Output 2 SD (640x360, 800k)	800
Output 3 SD (768x432, 1200k)	1,200
Output 4 SD (960x540, 1800k)	1,800
Total kbps:	4,200
Convert kbps to GB/hour 4,200 kbps / 8 (bits to bytes) / 1024 (kbps to Mbps) / 1024 (Mbps to Gbps) x 60 (seconds) x 60 (minutes)	1.85 GB/hour
GB/hour with redundancy	3.70 GB/hour
Cost (GB/hour * $0.03)	**$0.11/hour**

Table 4: MediaPackage packaging and origin pricing

AWS service	Cost per hour [USD]
Total MediaPackage with 1% of egress (7.91 GB x $0.05)	$0.40

Note

We assume that MediaPackage serves 1% of the traffic to CloudFront and viewers, and the rest is served by CloudFront. Table 5 provides the 1% of egress for the MediaPackage estimate in Table 4.

Viewer traffic pricing

The cost estimate for viewer traffic assumes that all viewers get the highest bitrate for the one-hour live streaming event.

Table 5: Viewer traffic pricing

AWS service	Function	Total
Amazon CloudFront	Average Mbps per viewer	1.8
	Total MB per sec (1000 x 1.8 / 8)	225
	Total Egress per hour (225 / 1024 x 60 x 60)	791 GB/hour
	Total cost (791 GB * $0.085):	**$67.24/hour**

Cost example 2

Cost example 2 covers a use case of approximately 10,000 viewers viewing a live event for about one hour with an HD-1080p encoding profile selected in the CloudFormation template. This cost example is based on the following factors:

- Pricing Region: US-East-1, assuming standard pricing (no free-tier or discounts).
- Viewers consume the highest bitrate: Note that bitrate consumption is a mix of all the streams, but the highest bitrate is used in the calculation to show the upper cost range. Additionally, the QVBR and variable video complexity can result in an output bandwidth that is 10-50% lower in price than the estimate provided in Table 6.
- 99% cache/hit ratio between the CDN and AWS Elemental MediaPackage.

Note

Storage of the test player is not included in this cost estimate.

Table 6 summarizes the total pricing for the live streaming event. Tables 7 through 10 breaks down the cost for each AWS service.

Table 6: Cost breakdown for 10,000 viewers for a one-hour live event

AWS service	Function	Cost per hour [USD]
AWS Elemental MediaLive	Input and outputs for channel	$3.75
AWS Elemental MediaPackage	Ingest of channel	$0.28
	Packaging and origination	$9.01
Amazon CloudFront	Distribution	$1,492.56
	Total:	**$1,505.60/hour**

AWS Elemental MediaLive pricing

Table 7 breaks down the AWS Elemental MediaLive pricing which assumes HD AVC input and both SD and HD AVC outputs with less than 10 Mbps bit rate and less than 30 fps frame rate.

Table 7: MediaLive pricing

Input / Output	Cost per hour [USD]
HD Input (AVC 10-20Mbps)	$0.234
Output 1 SD (512x288, 400k)	$0.44
Output 2 SD (640x360, 800k)	$0.44
Output 3 SD (768x432, 1200k)	$0.44

Input / Output	Cost per hour [USD]
Output 4 SD (960x540, 1800k)	$0.44
Output 5 HD (1280x720p, 2700k)	$0.88
Output 6 HD (1920x1080p, 4100k)	$0.88
Total:	**$3.75/hour**

AWS Elemental MediaPackage ingest pricing

As detailed in Cost Example 1, MediaPackage pricing is based on GB ingested per second across all streams of live output. Table 8 shows the conversion from kbps to Mbps to Gbits/second to GB/hour.

Table 8: MediaPackage ingest pricing

Input / output	kbps
Output 1 SD (512x288, 400k)	400
Output 2 SD (640x360, 800k)	800
Output 3 SD (768x432, 1200k)	1,200
Output 4 SD (960x540, 1800k)	1,800
Output 5 (1280x720p_2700k)	2,700
Output 6 (1920x1080p_4100k)	4,100
Total kbps:	11,000
Convert kbps to GB/hour 11000 kbps / 8 (bits to bytes) / 1024 (kbps to Mbps) / 1024 (Mbps to Gbps) x 60 (seconds) x 60 (minutes)	4.83

Input / output	kbps
GB/hour with redundancy	9.66
Total (GB/hour * $0.03)	**$0.28/hour**

Table 9: MediaPackage packaging and origin pricing

AWS service	Cost per hour [USD]
Total MediaPackage with 1% of egress (180 GB x $0.05)	$9.01

> **Note**
>
> We assume that MediaPackage serves 1% of the traffic to CloudFront and viewers, and the rest is served by CloudFront. Table 10 provides the 1% of egress for the MediaPackage estimate in Table 9.

Estimating the viewer traffic

The cost estimate for viewer traffic assumes that all viewers get the highest bitrate for the one-hour live streaming event.

Table 10: Viewer traffic pricing

AWS service	Function	Total
Amazon CloudFront	Average Mbps per viewer	4.1
	Total MB per sec (10,000 x 4.1 / 8)	5125
	Total Egress per hour (5125 / 1024 x 60 x 60)	18,017 GB/hour

AWS service	Function	Total
	Total cost 10,240 GB * $0.085 + 7,777 GB * 0.08):	**$1,531.49/hour**

Note

Pricing is per minute, with a minimum of 10 minutes. Prices are subject to change. For full details, refer to the pricing webpage for each AWS service you will be using in this solution.

Cost example 3

Cost example 3 covers a use case of approximately 100 viewers viewing live events for about two hundred hours every month with an HD-1080p encoding profile selected in the CloudFormation template. This cost example is based on the following factors::

- Pricing Region: US-East-1, assuming standard pricing (no free-tier or discounts).
- Viewers consume the highest bitrate: Note that bitrate consumption is a mix of all the streams, but the highest bitrate is used in the calculation to show the upper cost range. Additionally, the QVBR and variable video complexity can result in an output bandwidth that is 10-50% lower in price than the estimate provided in Table 11.
- 99% cache/hit ratio between the CDN and AWS Elemental MediaPackage.

Note

Storage of the test player is not included in this cost estimate.

Table 11summarizes the total pricing for the live streaming event. Tables 7 through 10 breaks down the cost for each AWS service.

Table 11: Cost breakdown for 10,000 viewers for a one-hour live event

AWS service	Function	Cost per 200 hours [USD]
AWS Elemental MediaLive	Input and outputs for channel	$750.00

AWS service	Function	Cost per 200 hours [USD]
AWS Elemental MediaPackage	Ingest of channel	$58.01
	Packaging and origination	$96.68
Amazon CloudFront	Distribution	$2,934.00
	Total:	**$3838.69/hour**

AWS Elemental MediaLive pricing

Table 12 breaks down the AWS Elemental MediaLive pricing which assumes HD AVC input and both SD and HD AVC outputs with less than 10 Mbps bit rate and less than 30 fps frame rate.

Table 12: MediaLive pricing

Input / Output	Cost per hour [USD]
HD Input (AVC 10-20Mbps)	$0.234 * 200 = $46.80
Output 1 SD (512x288, 400k)	$0.438 * 200 = $87.60
Output 2 SD (640x360, 800k)	$0.438 * 200 = $87.60
Output 3 SD (768x432, 1200k)	$0.438 * 200 = $87.60
Output 4 SD (960x540, 1800k)	$0.438 * 200 = $87.60
Output 5 HD (1280x720p, 2700k)	$0.882 * 200 = $176.40
Output 6 HD (1920x1080p, 4100k)	$0.882 * 200 = $176.40
Total:	**$750 for 200 hours**

AWS Elemental MediaPackage ingest pricing

As detailed in Cost Example 1, MediaPackage pricing is based on GB ingested per second across all streams of live output. Table13 shows the conversion from kbps to Mbps to Gbits/second to GB/hour.

Table 13: MediaPackage ingest pricing

Input / output	kbps
Output 1 SD (512x288, 400k)	400
Output 2 SD (640x360, 800k)	800
Output 3 SD (768x432, 1200k)	1,200
Output 4 SD (960x540, 1800k)	1,800
Output 5 (1280x720p_2700k)	2,700
Output 6 (1920x1080p_4100k)	4,100
Total kbps:	11,000
Convert kbps to GB/hour Convert kbps to GB/200 hours (11,000 kbps / 10 (kbps to Mbps) / 1024 (Mbps to Gbps) / 8 (Gbps to GB/s) x 60 (seconds) x 60 (minutes) x 200 (hours)	966.79
GB/200 hours with redundancy	1,933.59
Total (GB/hour * $0.03)	**$58.01**

Table 14: MediaPackage packaging and origin pricing

AWS service	Cost per hour [USD]
Total MediaPackage with 1% of egress (180 GB x $0.05)	$96.68

> **Note**
>
> We assume that MediaPackage serves 1% of the traffic to CloudFront and viewers, and the rest is served by CloudFront. Table 15 provides the 1% of egress for the MediaPackage estimate in Table 14.

Estimating the viewer traffic

The cost estimate for viewer traffic assumes that all viewers get the highest bitrate for the one-hour live streaming event.

Table 10: Viewer traffic pricing

AWS service	Function	Total
Amazon CloudFront	Average Mbps per viewer	4.1
	Total MB per sec (10,000 x 4.1 / 8)	51.25
	Total Egress per hour (5125 / 1024 x 60 x 60)	36,035 GB/ 200 hours
Total cost 10,240 GB * $0.085 + 25,795 GB * 0.08):		**$2,934.00/ 200 hours**

> **Note**
>
> Pricing is per minute, with a minimum of 10 minutes. Prices are subject to change. For full details, refer to the pricing webpage for each AWS service you will be using in this solution.

Security

When you build systems on AWS infrastructure, security responsibilities are shared between you and AWS. This shared responsibility model reduces your operational burden because AWS operates, manages, and controls the components including the host operating system, virtualization layer, and physical security of the facilities in which the services operate. For more information about AWS security, visit AWS Cloud Security.

Amazon CloudFront

This solution deploys a static website hosted in an Amazon S3 bucket. To help reduce latency and improve security, this solution includes an Amazon CloudFront distribution with an origin access identity, which is a special CloudFront user that helps restrict access to the solution's website bucket contents. For more information, refer to Restricting Access to Amazon S3 Content by Using an Origin Access Identity.

Supported AWS Regions

This solution uses AWS Elemental MediaLive, MediaPackage, and MediaConnect, which are currently available in specific AWS Regions only. Therefore, you must launch this solution in an AWS Region where these services are available. For the most current service availability by Region, refer to the AWS Regional Services List. If you use MediaConnect as input, you must deploy this solution in the same Region as your MediaConnect flows.

Quotas

Service quotas, also referred to as limits, are the maximum number of service resources or operations for your AWS account.

Quotas for AWS services in this solution

Make sure you have sufficient quota for each of the services implemented in this solution. For more information, refer to AWS service quotas.

Click one of the following links to go to the page for that service. To view the service quotas for all AWS services in the documentation without switching pages, view the information in the Service endpoints and quotas page in the PDF instead.

AWS CloudFormation quotas

Your AWS account has CloudFormation quotas that you should be aware of when launching the stack for this solution. By understanding these quotas, you can avoid limitation errors that would prevent you from deploying this solution successfully. For more information, refer to AWS CloudFormation quotas in the *AWS CloudFormation Users Guide*.

Deploy the solution

This solution uses AWS CloudFormation templates and stacks to automate its deployment. The CloudFormation template specifies the AWS resources included in this solution and their properties. The CloudFormation stack provisions the resources that are described in the template.

Deployment consideration

The Live Streaming on AWS solution configures AWS Elemental MediaLive with one of three progressive, 30 frames-per-second encoding profiles. You can choose one of the following encoding profiles.

- **HD-1080p profile** - 1920x1080, 1280x720, 960x540, 768x432, 640x360, 512x288
- **HD-720p profile** - 1280x720, 960x540, 768x432, 640x360, 512x288
- **SD-540p profile** - 960x540, 768x432, 640x360, 512x288

> ⚠ **Important**
>
> This solution includes an option to send anonymized operational metrics to AWS. We use this data to better understand how customers use this solution and related services and products. AWS owns the data gathered though this survey. Data collection is subject to the AWS Privacy Policy. To opt out of this feature, download the template, modify the AWS CloudFormation mapping section, and then use the AWS CloudFormation console to upload your template and deploy the solution. For more information, refer to the Anonymized data collection section of this guide.

AWS CloudFormation template

live-streaming-on-aws.template - Use this template to launch the solution and all associated components. The default configuration deploys an AWS Lambda function, an AWS Elemental MediaLive input and channel, an AWS Elemental MediaPackage channel, two Amazon CloudFront distributions, and an Amazon Simple Storage Service (Amazon S3) bucket for the demo HTML preview player, but you can also customize the template based on your specific needs.

Launch the stack

Before you launch the solution, review the cost, architecture, security, and other considerations discussed in this guide. Follow the step-by-step instructions in this section to configure and deploy the solution into your account.

Time to deploy: Approximately 20 minutes

This automated AWS CloudFormation template deploys the Live Streaming on AWS solution in the AWS Cloud.

1. Sign in to the AWS Management Console and select the button to launch the `live-streaming-on-aws` AWS CloudFormation template.

2. The template launches in the US East (N. Virginia) Region by default. To launch this solution in a different AWS Region, use the Region selector in the console navigation bar.

   > **Note**
   >
   > This solution uses the AWS Elemental MediaLive, MediaPackage, and MediaConnect services, which are currently available in specific AWS Regions only. Therefore, you must launch this solution in an AWS Region where these services are available. For the most current service availability by Region, see the AWS Regional Service List.

3. On the **Create stack** page, verify that the correct template URL is in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your Live Streaming on AWS solution stack.
5. Under **Parameters**, review the parameters for the template, and modify them as necessary.

   This solution uses the following default values. Refer to URL_PULL (HLS) input configuration, RTMP_PULL input configuration, RTMP_PUSH and RTP_PUSH input configuration, and MEDIACONNECT input configuration for detailed instruction for setting up each input type.

Parameter	Default	Description
LIVE STREAM SOURCE		

Parameter	Default	Description
Source Input Type	`URL_PULL`	Specify the input type for AWS Elemental MediaLive : `RTP_PUSH`, `RTMP_PUSH` , `RTMP_PULL` , `URL_PULL`, or `MEDIACONNECT` .
URL_PULL and RTML_PULL CONFIGURATION		
Primary Source URL	`https://d15an60oaeed9r.cloudfront.net/live_stream_v2/sports_reel_with_markers.m3u8`	The primary source URL for the live feed. By default, this parameter contains the primary demo source URL.
Primary Source Username	<Optional input>	If authentication is required to access the source, enter the username.
Primary Source Password	<Optional input>	If authentication is required to access the source, enter the password.
Secondary Source URL	`https://d3h5srgm8b0t83.cloudfront.net/live_stream_v2/sports_reel_with_markers.m3u8`	The secondary (backup) source URL for the live feed. By default, this parameter contains the secondary demo source URL.

Parameter	Default	Description
Secondary Source Username	<Optional input>	If authentication is required to access the secondary source, enter the username.
Secondary Source Password	<Optional input>	If authentication is required to access the secondary source, enter the password.
RTP_PUSH and RTMP_PUSH CONFIGURATION		
Input CIDR Block	*`<Requires input>`*	Specify the CIDR block for the MediaLive security group for push input types.
MEDIACONNECT CONFIGURATION		
Primary MediaConnect ARN	<Optional input>	The primary source MediaConnect flow for the live feed. You can create the flow in the MediaConnect console. To provide redundancy, create the primary and secondary flows in different Availability Zones.

Parameter	Default	Description
Secondary MediaConnect ARN	<Optional input>	The secondary source MediaConnect flow for the live feed. You can create the flow in the MediaConnect console. To provide redundancy, create the primary and secondary flows in different Availability Zones.
ENCODING OPTIONS		
Encoding Profile	`720`	Specify the encoding profile to use with MediaLive.
Start MediaLive Channel	`yes`	Choose whether to start the MediaLive channel when the solution is created. We recommend starting the channel if you will use the the section called "HTML preview player".

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template will create AWS Identity and Access Management (IAM) resources.
9. Choose **Create** to deploy the stack.

   You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a **CREATE_COMPLETE** status in approximately 20 minutes.

To test the live stream playback, navigate to the AWS CloudFormation stack **Outputs** tab, select the **Demo Console URL** from the **Value** column.

> **Note**
>
> In addition to the primary AWS Lambda function, this solution includes the `solution-helper` Lambda function, which runs only during initial configuration or when resources are updated or deleted.
> When running this solution, both Lambda functions are displayed in the AWS Lambda console, do not delete the `solution-helper` function because it is necessary to manage associated resources.

Once the live streaming event is complete, stop the resources created by this solution to help ensure that you don’t incur unnecessary AWS charges.

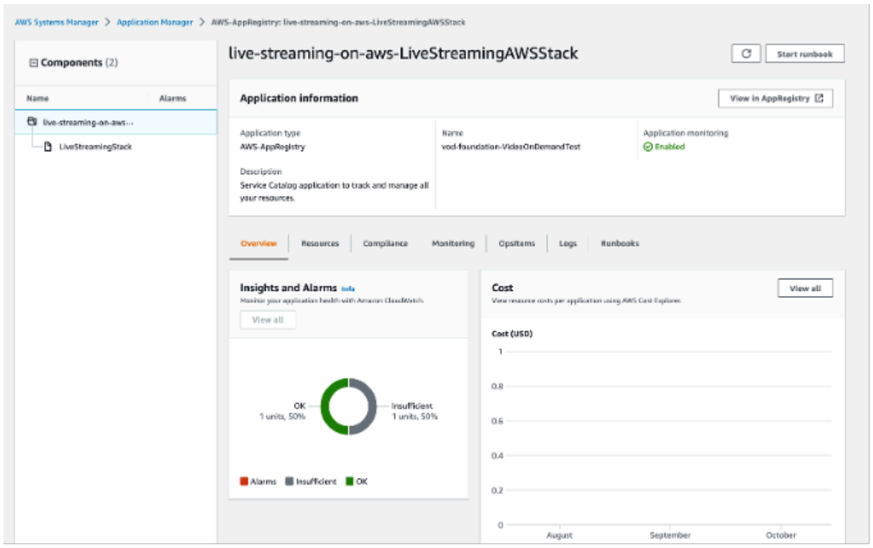
Monitoring the solution with AWS Service Catalog AppRegistry

The Live Streaming on AWS solution includes a Service Catalog AppRegistry resource to register the CloudFormation template and underlying resources as an application in both AWS Service Catalog AppRegistry and AWS Systems Manager Application Manager.

AWS Systems Manager Application Manager gives you an application-level view into this solution and its resources so that you can:

- Monitor its resources, costs for the deployed resources across stacks and AWS accounts, and logs associated with this solution from a central location.
- View operations data for the resources of this solution in the context of an application, such as deployment status, CloudWatch alarms, resource configurations, and operational issues.

The following figure depicts an example of the application view for the Live Streaming on AWS stack in Application Manager.



AWS Systems Manager > Application Manager > AWS-AppRegistry: live-streaming-on-aws-LiveStreamingAWSStack

live-streaming-on-aws-LiveStreamingAWSStack

Application information

Application type: AWS-AppRegistry

Name: vod-foundation-VideoOnDemandTest

Application monitoring: Enabled

Description: Service Catalog application to track and manage all your resources.

Insights and Alarms: Monitor your application health with Amazon CloudWatch.

Cost: View resource costs per application using AWS Cost Explorer.

Live Streaming on AWS stack in Application Manager

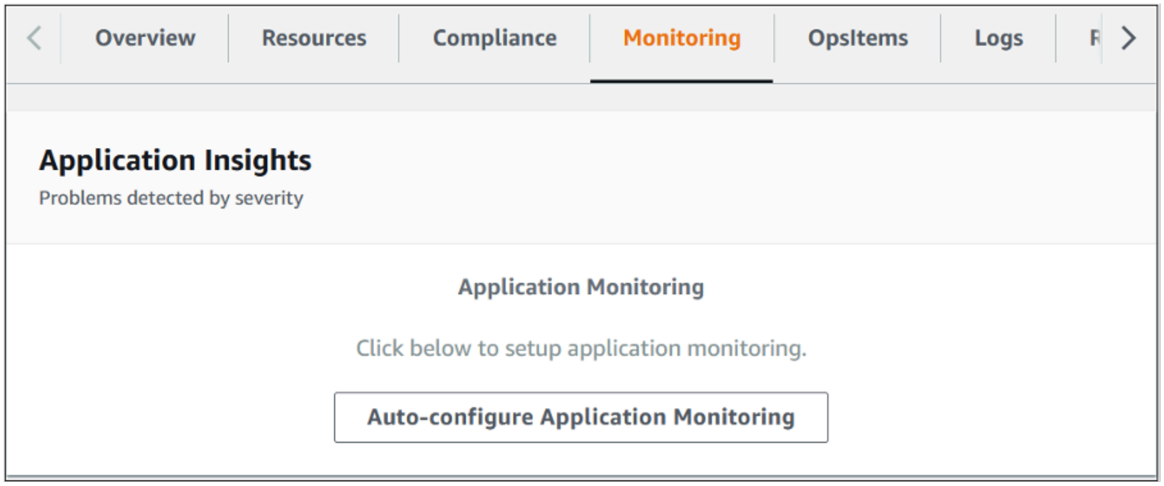
> **Note**
>
> You must activate CloudWatch Application Insights, AWS Cost Explorer, and cost allocation tags associated with this solution. They are not activated by default.

Activate CloudWatch Application Insights

1. Sign in to the Systems Manager console.
2. In the navigation pane, choose **Application Manager**.
3. In **Applications**, choose **AppRegistry applications**.
4. In **AppRegistry applications**, search for the application name for this solution and select it.

The next time you open Application Manager, you can find the new application for your solution in the **AppRegistry application** category.

5. In the **Components** tree, choose the application stack you want to activate.
6. In the **Monitoring** tab, in **Application Insights**, select **Auto-configure Application Monitoring**.



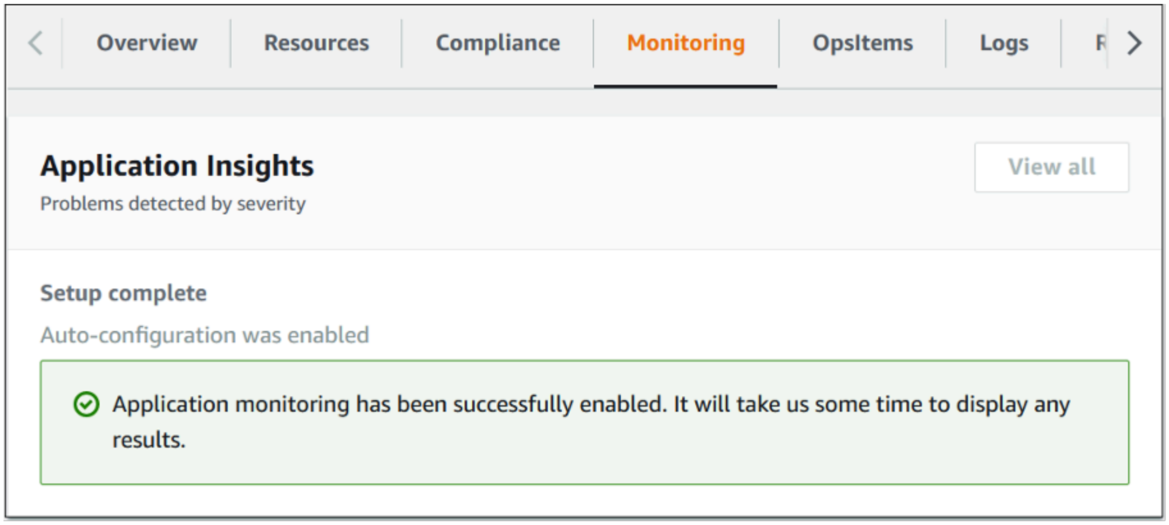
Application Insights

Problems detected by severity

Application Monitoring

Click below to setup application monitoring.

Monitoring for your applications is now activated and the following status box appears:



Application Insights

Problems detected by severity

Setup complete

Auto-configuration was enabled

Application monitoring has been successfully enabled. It will take us some time to display any results.

Activate AWS Cost Explorer

You can see the overview of the costs associated with the application and application components within the Application Manager console through integration with AWS Cost Explorer which must be first activated. Cost Explorer helps you manage costs by providing a view of your AWS resource costs and usage over time. To activate Cost Explorer for the solution:

1. Sign in to the AWS Cost Management console.
2. In the navigation pane, select **Cost Explorer**.
3. On the **Welcome to Cost Explorer** page, choose **Launch Cost Explorer**.

The activation process can take up to 24 hours to complete. Once activated, you can open the Cost Explorer user interface to further analyze cost data for the solution.

Activate cost allocation tags associated with the solution

After you activate Cost Explorer, you must activate a cost allocation tag to see the costs for this solution. The cost allocation tags can only be activated from the management account for the organization. To activate cost allocation tags:

1. Sign in to the AWS Billing and Cost Management console.
2. In the navigation pane, select **Cost Allocation Tags**.
3. On the Cost allocation tags page, filter for the `AppManagerCFNStackKey` tag, then select the tag from the results shown.

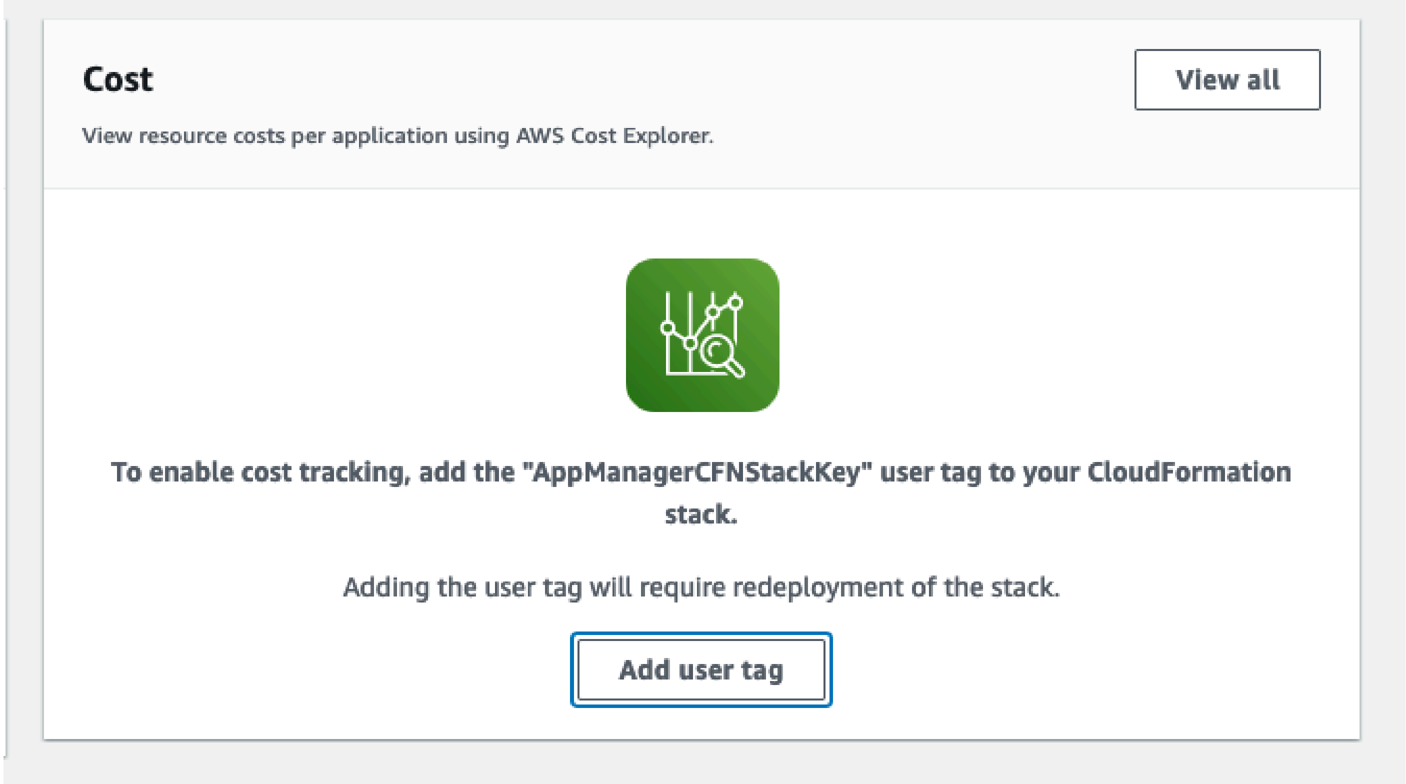
4. Choose **Activate**.

The activation process can take up to 24 hours to complete and the tag data to appear.

Confirm cost tags associated with the solution

After you activate cost allocation tags associated with the solution, you must confirm the cost allocation tags to see the costs for this solution. To confirm cost allocation tags:

1. Sign in to the Systems Manager console.
2. In the navigation pane, choose **Application Manager**.
3. In **Applications**, choose the application name for this solution and select it.
4. In the **Overview** tab, in **Cost**, select **Add user tag**.



Cost

View resource costs per application using AWS Cost Explorer.

To enable cost tracking, add the "AppManagerCFNStackKey" user tag to your CloudFormation stack.

Adding the user tag will require redeployment of the stack.

5. On the **Add user tag** page, enter `confirm`, then select **Add user tag**.

The activation process can take up to 24 hours to complete and the tag data to appear.

Uninstall the solution

You can uninstall the Live Streaming on AWS solution from the AWS Management Console or by using the AWS Command Line Interface. You must manually delete the Amazon Simple Storage Service (Amazon S3) bucket created by this solution. AWS Solutions Implementations do not automatically delete the S3 bucket in case you have stored data to retain.

Using the AWS Management Console

1. Sign in to the AWS CloudFormation console.
2. On the **Stacks** page, select this solution's installation stack.
3. Choose **Delete**.

Using AWS Command Line Interface

Determine whether the AWS Command Line Interface (AWS CLI) is available in your environment. For installation instructions, refer to What Is the AWS Command Line Interface in the *AWS CLI User Guide*. After confirming that the AWS CLI is available, run the following command.

```
$ aws cloudformation delete-stack --stack-name <installation-stack-name>
```

Delete the Amazon S3 buckets

This solution is configured to retain the solution-created Amazon S3 buckets if you decide to delete the AWS CloudFormation stack to prevent accidental data loss. After uninstalling the solution, you can manually delete the S3 buckets if you do not need to retain the data. Follow these steps to delete the Amazon S3 buckets.

1. Sign in to the Amazon S3 console.
2. Choose **Buckets** from the left navigation pane.
3. Locate the *<stack-name>* S3 buckets.
4. Select the S3 bucket and choose **Delete**.

To delete the S3 buckets using AWS CLI, run the following command. You will not need to empty the bucket first when using the `--force` option.

```
$ aws s3 rb s3://<bucket-name> --force
```

Developer guide

This section provides the source code for the solution.

Source code

Visit our GitHub repository to download the templates and scripts for this solution, and to share your customizations with others. The Live Streaming on AWS template is generated using the AWS CDK. Refer to the README.md file for additional information.

Reference

This section includes information about an optional feature for collecting unique metrics for this solution, and a list of builders who contributed to this solution.

Anonymized data collection

This solution includes an option to send anonymized operational metrics to AWS. We use this data to better understand how customers use this solution and related services and products. When invoked, the following information is collected and sent to AWS:

- **Solution ID -** The AWS solution identifier
- **Unique ID (UUID) -** Randomly generated, unique identifier for each Live Streaming on AWS deployment
- **Timestamp -** Data-collection timestamp
- **Example: Instance Data:** Count of the state and type of instances that are managed by the EC2 Scheduler in each AWS Region

Example data:

Running: {`t2.micro:2`}, {`m3.large:2`}

Stopped: {`t2.large:1`}, {`m3.xlarge:3`}

AWS owns the data gathered though this survey. Data collection is subject to the AWS Privacy Policy. To opt out of this feature, complete the following steps before launching the AWS CloudFormation template.

1. Download the AWS CloudFormation template to your local hard drive.
2. Open the AWS CloudFormation template with a text editor.
3. Modify the AWS CloudFormation template mapping section from:

   ```
   AnonymizedData:
       SendAnonymizedData:
         Data: Yes
   ```

   to:

```
AnonymizedData:
    SendAnonymizedData:
      Data: No
```

4. Sign in to the AWS CloudFormation console.
5. Select **Create stack**.
6. On the **Create stack** page, specify template section, select **Upload a template file**.
7. Under **Upload a template file**, choose **Choose file** and select the edited template from your local drive.
8. Choose **Next** and follow the steps in Launch the stack in the Deploy the solution section of this guide.

Contributors

- Tom Nightingale
- Tom Gilman
- Joan Morgan
- Eddie Goynes
- Kiran Patel
- Aijun Peng
- San Dim Ciin

Revisions

Date	Change
November 2016	Initial release
March 2017	Added an AWS CloudFormation parameter for the Elemental Live server version.
April 2017	Changed all Elemental references to AWS Elemental to reflect the new name.
September 2018	Added information about AWS Elemental MediaLive and MediaPackage functionality, and encoding profiles.
December 2018	Added information about the Amazon CloudFront distribution for the static website hosted in the Amazon S3 bucket.
March 2019	Added information about MediaConnect input and CMAF format functionality.
December 2019	Updated Cost information; updated the Lambda run-times to Node 12.x and Python 3.8.
July 2020	Updated encoding settings and appendices with input configuration details. For a detailed description of the changes from version 2.3.0 to version 2.4.0, refer to the CHANGELOG.md file in the GitHub repository.
June 2021	Documentation updates only: Expanded the cost estimates to provide more granular information, clarified the collection of operational metrics information, and updated guide organization to improve readability.
August 2021	Release version 3.0.0: Updated with new features and bug fixes. For more information, refer to the CHANGELOG.md file in the GitHub repository.

Date	Change
September 2021	Documentation updates only: Provided additional details about the demo preview player.
November 2021	Release version 3.1.0: bug fixes. For more information, refer to the CHANGELOG.md file in the GitHub repository.
January 2022	Release version 3.1.1: bug fixes. For more information, refer to the CHANGELOG.md file in the GitHub repository.
March 2022	Release version 3.1.2: bug fixes. For more information, refer to the CHANGELOG.md file in the GitHub repository.
July 2022	Release version 4.0.0: solution conversion to AWS CDK. For more information, refer to the CHANGELOG.md file in the GitHub repository.
August 2022	Release version 4.0.1: Minor update and bug fixes. For more information, refer to the CHANGELOG.md file in the GitHub repository.
November 2022	Release version 4.1.0: Added AppRegistry support for the solution. For more information, refer to the CHANGELOG.md file in the GitHub repository.
April 2023	Release version 4.2.0: Updated solution to use CDK 2 from CDK 1, updated AppRegistry, and security updates. Also, added a couple of new sections in the documentation, such as solution features and benefits, use cases, concepts and definitions, and the Well-Architected design framework. For more information, refer to the CHANGELOG.md file in the GitHub repository.
April 2023	Release version 4.2.1: Mitigated impact caused by new default settings for S3 Object Ownership (ACLs disabled) for all new S3 buckets. For more information, refer to the CHANGELOG.md file in the GitHub repository.

Date	Change
August 2023	Release version 4.2.2: Updated npm packages and lambda runtimes to NodeJS 18 and AWS SDK v3. Updated Cost table and added additional cost examples. Updated Cost table and added additional cost examples. For more information, refer to the CHANGELOG.md file in the GitHub repository.
October 2023	Release version 4.2.3: Updated package versions to resolve security vulnerabilities. For more information, refer to the CHANGELOG.md file in the GitHub repository.
November 2023	Documentation update: Added Confirm cost tags associated with the solution to the Monitoring the solution with AWS Service Catalog AppRegistry section.

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents AWS current product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided "as is" without warranties, representations, or conditions of any kind, whether express or implied. AWS responsibilities and liabilities to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.