# Live Streaming on AWS

## AWS Implementation Guide

*Tom Nightingale*

*Tom Gilman*

*Joan Morgan*

*Kiran Patel*

*Aijun Peng*

November 2016

*Last updated: July 2020 (see revisions)*

## Contents

## About this guide

This implementation guide discusses architectural considerations and configuration steps for deploying Live Streaming on AWS in the Amazon Web Services (AWS) Cloud. It includes links 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.

# Overview

Amazon Web Services (AWS) enables broadcasters and content owners to automate media supply chains, streamline content distribution, and cost-effectively broadcast live content to a global audience. With the AWS Cloud, you can seamlessly scale your infrastructure to meet demand, and pay only for what you use, helping you build highly available and cost-effective solutions for live video streaming.

AWS offers several options for delivering live streaming content on the cloud that enable you to automatically provision and dynamically scale any combination of video processing, delivery, and storage services. Customers can combine AWS Media Services with Amazon CloudFront to build a highly resilient and secure architecture that delivers an exceptional real-time viewing experience.

The Live Streaming on AWS solution automatically configures AWS Elemental MediaLive and AWS Elemental MediaPackage to encode and package 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). The solution also uses Amazon CloudFront to provide an elastic, highly available, global content delivery network for live video streaming. The solution is easy to deploy and used only during the live event. When you finish streaming, you can delete the solution's stack to help ensure that you only pay for the infrastructure that you use.

## Cost

You are responsible for the cost of the AWS services used while running this live streaming solution. As of the date of publication, the cost for running this solution in the US East (N. Virginia) Region is shown in the table below. The cost depends on the encoding profile you choose, and does not include data transfer fees, the cost for the demo HTML preview player, or Amazon CloudFront and AWS Elemental MediaPackage costs, which will vary depending on the number of users and the types of end user devices.

| Encoding profile | Total cost/hour |
| --- | --- |
| 1080 (6 encodes) | $4.00 |
| 720 (5 encodes) | $3.00 |
| 540 (4 encodes) | $2.00 |

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

## Architecture overview

Deploying this solution builds the following environment in the AWS Cloud.
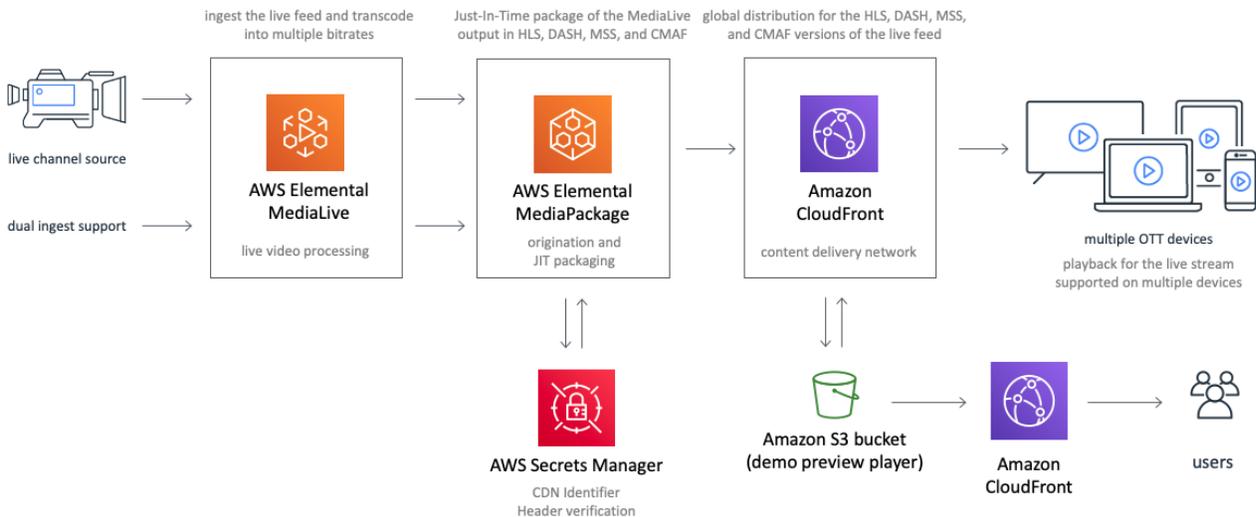


**Figure 1: Live Streaming on AWS solution architecture**

The solution's AWS CloudFormation template launches the AWS products and services necessary to ingest, transcode, and deliver live streaming video. 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.  For more information, view the Creating Adaptive Bitrate (ABR) Outputs training.

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.

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, see the [CDN Authorization in AWS Elemental MediaPackage]() in the *AWS Elemental MediaPackage User Guide*.

The CloudFront distribution delivers your live stream to viewers with low latency and high transfer speeds.

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, which are ABR stream sets ranging in output resolution from 270p to 1080p, that you can set during initial configuration.

This solution also deploys a demo HTML preview player that you can use to 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.

# Deployment considerations

## Encoding profiles

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

## Regional deployment

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, see [AWS service offerings by Region](). If you use MediaConnect as input, you must deploy this solution in the same Region as your MediaConnect flows.

# AWS CloudFormation template

This solution uses AWS CloudFormation to automate the deployment of the Live Streaming on AWS solution in the AWS Cloud. It includes the following AWS CloudFormation template, which you can download before deployment:

**View 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.

# Automated deployment

Before you launch the solution, please review the architecture, configuration, network 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

## Launch the stack

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

> **Note**:  You are responsible for the cost of the AWS services used while running this solution. See the Cost section for more details. For full details, see the pricing webpage for each AWS service you will be using in this solution.

1. Sign in to the AWS Management Console and click the button to the right to launch the `live-streaming-on-aws` AWS CloudFormation template. You can also download the template as a starting point for your own implementation.

   **Launch Solution**

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 AWS service offerings by Region.

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. See Appendices A through D for detailed instruction for setting up each input type.

| Parameter | Default | Description |
| --- | --- | --- |
| **LIVE STREAM SOURCE** | | |
| **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://d15an60oa eed9r.cloudfront. net/live_stream_v 2/sports_reel_wit h_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://d3h5srgm8 b0t83.cloudfront. net/live_stream_v 2/sports_reel_wit h_markers.m3u8` | The secondary (backup) source URL for the live feed. By default, this parameter contains the secondary demo source URL. |
| **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** | | |

| Parameter | Default | Description |
|---|---|---|
| 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. |
| 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 | `true` | Choose whether to start the MediaLive channel when the solution is created. We recommend starting the channel if you will use the demo 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 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 see a status of CREATE_COMPLETE in roughly 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, you will see both Lambda functions 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, terminate the resources created by this solution to help ensure that you don't incur unnecessary AWS charges.

# Security

When you build systems on AWS infrastructure, security responsibilities are shared between you and AWS. This shared model can reduce your operational burden as AWS operates, manages, and controls the components from the host operating system and virtualization layer down to the physical security of the facilities in which the services operate. For more information about security on AWS, visit the AWS Security Center.

## 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, see Restricting Access to Amazon S3 Content by Using an Origin Access Identity.

# Additional resources

## AWS services

- AWS Elemental MediaLive

- AWS Elemental MediaPackage

- AWS Elemental MediaConnect

- Amazon CloudFront

- AWS CloudFormation

- AWS Lambda

## AWS blogs

- Connecting AWS Elemental Live On-Premises to AWS Media Services in the Cloud

- Connecting OBS Studio to AWS Media Services in the Cloud

- Connecting FFmpeg Using RTP to AWS Media Services in the Cloud

- Connecting FFmpeg Using RTMP to AWS Media Services in the Cloud

- Connecting VLC Media Playing Using RTP to AWS Media Services in the Cloud

## Tutorials and workshops

- AWS Live Streaming and Live-to-VOD Workshop

- Build a Secure Packager and Encoder Key Exchange (SPEKE) Reference Server

- Build a Frame-Accurate Live-to-VOD Workflow with Image Recognition

# Appendix A: 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 **true**. Otherwise, select **false**—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*.

# Appendix B: 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 **true**. Otherwise, select **false**—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*.

# Appendix C: 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 **true**. Otherwise, select **false**—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*.

# Appendix D: 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 **true**. Otherwise, select **false**—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*.

# Appendix E: Collection of operational metrics

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

- **Solution ID:** The AWS solution identifier

- **Unique ID (UUID):** Randomly generated, unique identifier for each live streaming solution deployment

- **Timestamp:** Data-collection timestamp

- **Launch Data:** The AWS Region where the stack was launched

- **Source Input Type:** The input type you selected at launch

- **Source Input Codec:** The codec you selected at launch

- **Source Input Resolution:** The resolution you selected at launch

AWS owns the data gathered through this survey.  Data collection is subject to the [AWS Privacy Policy](). To opt out of this feature, modify the AWS CloudFormation template mapping section as follows:

```
AnonymousData:
    SendAnonymousData:
      Data: Yes
```

to

```
AnonymousData:
    SendAnonymousData:
      Data: No
```

# Source code

This solution includes two source code options, NodeJS and Python. Use the **Source Code** AWS CloudFormation parameter to choose the source code you want. You can visit our [GitHub repository](#) to download the templates and scripts for this solution, and to share your customizations with others.

# Document 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, see the [changelog file](#) in the GitHub repository. |

**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's 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.

Live Streaming on AWS is licensed under the terms of the Apache License Version 2.0 available at https://www.apache.org/licenses/LICENSE-2.0.