
Video on Demand on AWS Foundation Implementation Guide



Video on Demand on AWS Foundation: Implementation Guide

Copyright © 2021 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

Home	1
Overview	2
Cost	2
Example cost for a 60-minute source video	2
Architecture	3
Components	5
Ingest	5
Encoding	5
Processing	5
Considerations	7
Customization	7
Regional deployments	7
Template	8
Automated deployment	9
Launch the stack	9
Security	11
Amazon S3 bucket policy	11
IAM roles	11
Additional resources	12
Appendix A: Working with multiple job settings files	13
Appendix B: Troubleshooting	14
Appendix C: Operational metrics	15
Source code	16
Contributors	17
Revisions	18
Notices	19

Video on Demand on AWS Foundation

AWS Solutions Implementation Guide

Publication date: **November 2020**

This implementation guide discusses architectural considerations and configuration steps for deploying the Video on Demand on AWS Foundation solution 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 on demand workflows 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 deliver media content to a global audience. With AWS, 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 video-on-demand workflows.

AWS offers its customers several methods to cost-effectively deliver video content to a global audience in the AWS Cloud. Using these methods, you can dynamically scale any combination of video processing, delivery, and storage services. [AWS Elemental MediaConvert](#) transcodes media files from their source format into versions that will playback on devices like smartphones, tablets, and PCs; and [Amazon CloudFront](#) accelerates delivery of your video content to end users. You can leverage these AWS services to build a highly available, resilient architectures that ingest, store, process, and deliver video content on demand.

Video on Demand on AWS Foundation is a reference implementation that automatically provisions the AWS services necessary to build a scalable, distributed video-on-demand workflow. This solution combines [Amazon Simple Storage Service](#) (Amazon S3), a highly-scalable, reliable, and low-latency object store, MediaConvert for production grade transcoding, and Amazon CloudFront for global distribution. This solution also leverages [AWS Lambda](#), a service that allows you to run code without provisioning or managing servers, and [Amazon CloudWatch](#) to track encoding jobs in MediaConvert, and [Amazon Simple Notification Service](#) (Amazon SNS) to send notifications for completed jobs.

This solution is designed to help you quickly begin encoding video files with MediaConvert. Out of the box, this solution provides a sample MediaConvert job settings JSON file, which is used to transcode videos uploaded to an Amazon S3 bucket. You can customize the architecture to meet your specific needs. For more information, refer to [Customization \(p. 7\)](#).

If you are looking to build out more complex workflows with options around ingest processing and publishing video content, AWS also offers the [Video on Demand on AWS](#) solution.

Cost

You are responsible for the cost of the AWS services used while running this solution. The total cost for running this solution depends on the size of your videos, the number of outputs created, and the number of views the published content receives through CloudFront. The majority of the cost will come from MediaConvert. For more information about MediaConvert pricing, refer to [AWS Elemental MediaConvert Pricing](#). Prices are subject to change. For full details, refer to the pricing webpage for each AWS service you will be using in this solution.

Example cost for a 60-minute source video

With the solution deployed in the US East (N. Virginia) Region and using the default encoding settings to process a 60-minute source video, it will cost approximately \$11.82. This estimate is based on the pricing available at the time of publishing this solution and may vary depending on the source video size and format.

The video outputs in this example use the following MediaConvert settings: AVC codec, 1 pass quality, 30 fps, with the DeInterlacer enabled.

Professional tier costs for the MediaConvert settings used in this example:

Output	Cost
SD resolution	\$0.012/min
HD resolution	\$0.024/min
Audio only	\$0.005/min

Outputs:

- a. File group: 1 HD with captions
- b. HLS profile: 2 HD and 3 SD with captions
- c. DASH profile: 2 HD and 3 SD with captions
- d. 1 audio only output for the DASH profile

Calculations for each output:

- a. $(1 \times 60 \times \$0.024) = \1.44
 - b. $(2 \times 60 \times \$0.024) + (3 \times 60 \times \$0.012) = \$5.04$
 - c. $(2 \times 60 \times \$0.024) + (3 \times 60 \times \$0.012) = \$5.04$
 - d. $(1 \times 60 \times \$0.005) = \0.30
- Total cost = \$11.82 (\$1.44 + \$5.04 + \$5.04 + \$0.30)

The costs for CloudFront and Amazon S3 storage will vary depending on the number and format of outputs created and the number of requests to view the content delivered through CloudFront.

Architecture overview

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

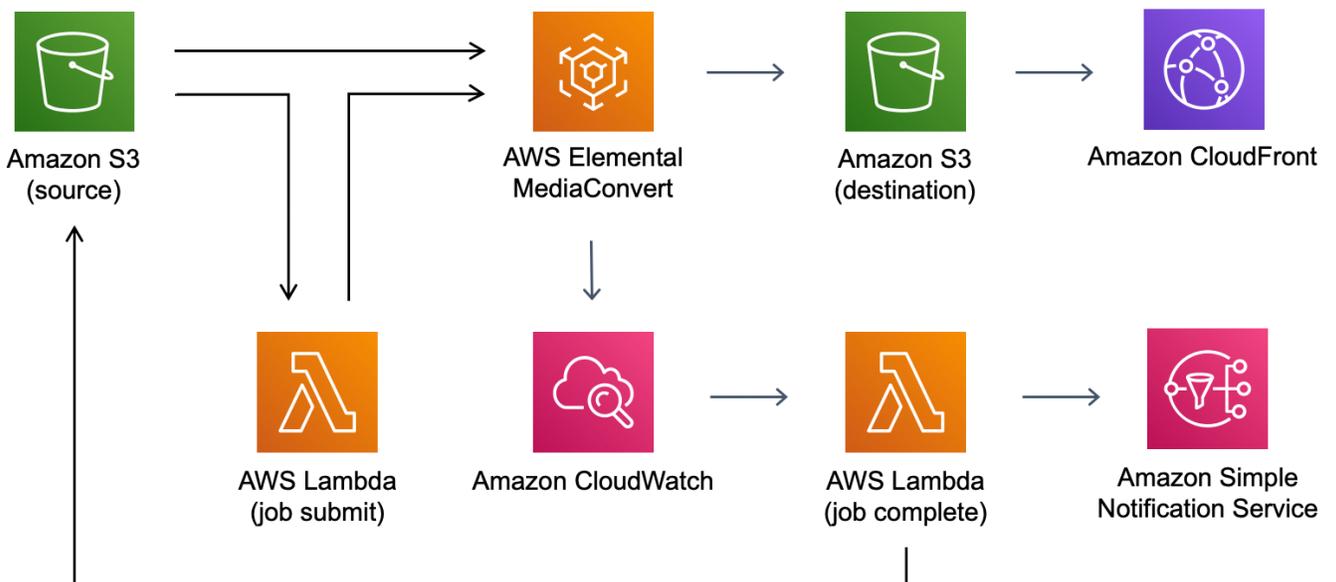


Figure 1: Video on Demand on AWS Foundation architecture

The architecture diagram represents the following flow:

- The AWS CloudFormation template deploys a workflow that ingests source videos, transcodes the videos into multiple Adaptive Bitrate Formats (ABR) and delivers the content through CloudFront.
- The solution creates a source Amazon S3 bucket to store the source video files, and a destination bucket to store the outputs from MediaConvert.
- A sample job settings file, used to define the encoding settings for MediaConvert, is uploaded to the source S3 bucket.
- The solution includes two AWS Lambda functions: a job submit function to create the encoding jobs in MediaConvert and a job complete function to process the outputs.
- Amazon CloudWatch tracks encoding jobs in MediaConvert and triggers the Lambda job complete function.
- An Amazon SNS topic is deployed to send notifications of completed jobs, and CloudFront is configured with the destination S3 bucket as the origin for global distribution of the transcoded video content.

Solution components

Ingest

To trigger the video processing workflow, you must upload the source video assets to the source S3 bucket through standard tools. For example, the AWS Management Console, AWS Command Line Interface (Snowball client/AWS CLI), or third-party tools that interface with Amazon Simple Storage Service (Amazon S3).

By default, this solution creates an `assets01` folder in the root of the source S3 bucket with a `job-settings.json` file. Each time you upload a video to the `assets01` folder, or any other folder that you created, an Amazon CloudWatch Events rule invokes the `job-submit` Lambda function. This function receives the details for the source video from the event, applies the settings contained in the job settings file in the same top-level folder as the uploaded video in S3, and submits a job to AWS Elemental MediaConvert using the processed job settings file.

To track the job in MediaConvert, the name of the workflow defined at deployment and a globally unique identifier (GUID) created by the `job-submit` Lambda function, are included in the job submitted to MediaConvert.

Encoding

The sample job settings file created as part of the AWS CloudFormation deployment has the [Quality Variable Bit Rate \(QVBR\)](#) rate control enabled with [accelerated transcoding](#) set to `PREFERRED` and generates the following outputs:

- MP4 @ 1920 x 1080
- HLS Adaptive Bit Rate (ABR) with 5 renditions @ 1920 x 1080, 1280 x 720, 960 x 540, 640 x 360, 480 x 270
- DASH Adaptive Bit Rate (ABR) with 5 renditions @ 1920 x 1080, 1280 x 720, 960 x 540, 640 x 360, 480 x 270

To change the settings, you can update or replace the job settings file in Amazon S3 with your own settings. For details, refer to [Exporting and importing AWS Elemental MediaConvert jobs](#) in the *MediaConvert User Guide*. Only export jobs that have successfully run and completed to ensure validity of job settings file.

Processing

A CloudWatch Events rule is configured to trigger the `job-complete` Lambda function each time an encoding job starts up, completes successfully, or fails in MediaConvert. This function retrieves the details of the job from the event and generates the CloudFront URLs for the MediaConvert outputs. The details for the input file, job settings, and outputs are then added to a `jobs-manifest.json` file stored at the root of the source S3 bucket.

```
{  
  "Jobs":
```

```
{
  "jobId-0001": {
    "FileInput": "s3://SOURCE_BUCKET/example.mp4",
    "JobSettings": {...},
    "Outputs": {
      "HLS": "https://cloudfront.net/...",
      "DASH": "https://cloudfront.net/..."
    }
  },
  "jobId-0002": {
    "FileInput": "s3://SOURCE_BUCKET/example2.mp4",
    "JobSettings": {...},
    "Outputs": {
      "HLS": "https://cloudfront.net/...",
      "DASH": "https://cloudfront.net/..."
    }
  }
}
...
}
```

Video on Demand on AWS Foundation jobs manifest file

Figure 2: Video on Demand on AWS Foundation jobs manifest file

The job-complete Lambda function also sends a summary of the job and the outputs to the Amazon Simple Notification Service (Amazon SNS) topic created at deployment. Any errors from the encoding process are also captured by the Lambda function and sent to the SNS topic.

Design considerations

Customization

This solution is a serverless architecture that can be easily updated and extended for your specific video processing needs. For example, adding or replacing Amazon Simple Notification Service (Amazon SNS) with [Amazon Simple Queue Service \(Amazon SQS\)](#) to allow upstream workflows to subscribe and get notifications on the workflow outputs. You can also add multiple folders and job settings files in the source S3 bucket to accommodate different use cases. Refer to [Appendix A \(p. 13\)](#) for details on using multiple job settings files.

Regional deployments

This solution uses AWS Elemental MediaConvert which is available in specific AWS Regions only. Therefore, you must deploy this solution in a Region that supports this service. For the most current service availability by Region, refer to [AWS Regional Services](#).

AWS CloudFormation template

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

[View
Template](#)

video-on-demand-on-aws-foundation.template: Use this template to launch the Video on Demand on AWS Foundation solution and all associated components. The default configuration deploys AWS Lambda functions, Amazon Simple Storage Service (Amazon S3) buckets, AWS Elemental MediaConvert, Amazon CloudWatch Logs, Amazon CloudWatch Events rules, Amazon Simple Notification Service (Amazon SNS) topics, and an Amazon CloudFront distribution. You can also customize the template based on your specific needs.

Automated deployment

Before you launch the automated deployment, review the architecture and other considerations discussed in this guide. Follow the step-by-step instructions in this section to configure and deploy the Video on Demand on AWS Foundation solution into your account.

Time to deploy: Approximately 10 minutes

Launch the stack

The automated AWS CloudFormation template deploys the Video on Demand on AWS Foundation solution in the AWS Cloud.

Note

You are responsible for the cost of the AWS services used while running this solution. See the [Cost \(p. 2\)](#) 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 select the button below to launch the `video-on-demand-on-aws-foundation.template` AWS CloudFormation template.



You can also [download the template](#) as a starting point for your own implementation.

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 AWS Elemental MediaConvert, which is available in specific AWS Regions only. Therefore, you must deploy this solution in a Region that supports this service. For the most current service availability by Region, refer to [AWS Regional Services](#).

3. On the **Select template** page, verify that you selected the correct template and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack.
5. Under **Parameters**, review the parameters for the template and modify them as necessary.

This solution uses the following default value.

Parameter	Default	Description
Notification Email Address	<Requires input>	A valid email address to receive Amazon SNS notifications.

6. Choose **Next**.
7. On the **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 see a status of **CREATE_COMPLETE** in approximately 10 minutes.

After the stack is created, Amazon SNS sends three subscription notifications to the admin email address with links to enable encoding, publishing, and error notifications.

10. In the subscription notification emails, select each link to enable SNS notifications.

Note

In addition to the AWS Lambda functions that create solution resources and manage the workflow, this solution includes the `custom-resource` Lambda function, which runs only during initial configuration or when resources are updated or deleted.

When running this solution, the `custom-resource` Lambda function is inactive. However, do not delete the function, since it is necessary to manage associated resources.

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 Cloud Security](#).

Amazon S3 bucket policy

The Amazon Simple Storage Service (Amazon S3) buckets for AWS Elemental MediaConvert output includes a policy that allows access from Amazon CloudFront. Because the Amazon CloudFront endpoints are publicly accessible, the MediaConvert output bucket is also publicly accessible when accessed via CloudFront. For information on how to secure Amazon CloudFront, refer to [Serving Private Content through CloudFront](#) in the *Amazon CloudFront Developer Guide*.

IAM roles

AWS Identity and Access Management (IAM) roles enable customers to assign granular access policies and permissions to services and users in the AWS Cloud. The Video on Demand on AWS Foundation solution creates several IAM roles, including a role that grants AWS Elemental MediaConvert access to Amazon Simple Storage Service. This role is necessary to allow the services to operate in your account.

Additional resources

- | | |
|--|--|
| <ul style="list-style-type: none">• Amazon CloudFront• Amazon CloudWatch• Amazon Simple Notification Service• Amazon Simple Storage Service | <ul style="list-style-type: none">• AWS CloudFormation• AWS Elemental MediaConvert• AWS Lambda |
|--|--|

Appendix A: Working with multiple job settings files

By default, this solution creates an `assets01` folder in the root of the Amazon Simple Storage Service (Amazon S3) bucket with a `job-settings.json` file. Each time a video is uploaded to this folder, it triggers a workflow to apply job settings to the job created in AWS Elemental MediaConvert.

Note

The S3 event notification is configured to trigger the workflow when a video is uploaded with one of the following lowercase file extensions: `.mp4`, `.mpg`, `.m4v`, `.m2ts`, and `.mov`.

To support different job settings, create additional folders at the root of the source S3 bucket and include different job settings for each folder. For example:

```
assets01/  
  job-settings.json  
  video-01.mp4  
assets02/  
  job-settings.json  
  video-02.mp4  
  subfolder/video-03.mpg
```

With the above configuration in the source S3 bucket, `video-01.mp4` files are encoded with the settings stored in the `assets01` folder. Meanwhile, `video-02.mp4` and `video-03.mpg` files are encoded using the settings in the `assets02` folder.

Note

There are no specific requirements for the folder names, except you must name the settings file `job-settings.json`.

Although you can export a completed job from MediaConvert to use as a job settings file, this solution does not support [input stitching](#) or [input clipping](#), because it only triggers on the upload of one video file source.

If `AccelerationSettings` is not defined in the job settings JSON file, this solution will automatically add this and set it to `PREFERRED`.

We recommend that you provide a [custom name for your output groups](#), especially if your job settings include more than one output group of the same type, for example, three HLS output groups. The output group name is used as part of the output destination path and having distinct names makes it easier to locate where each output is being written.

Appendix B: Troubleshooting

The email address you provided when deploying this solution gets notifications both when jobs complete successfully on AWS Elemental MediaConvert and when jobs fail. It also gets notifications about errors that might have occurred while trying to submit a job or process the output from a job.

If notified about a MediaConvert job failure, navigate to the MediaConvert console and select the job ID of the job that failed. This takes you to the **Job Summary** page. The **Overview** section includes an error message with more information on why the job failed. Also, on this page there are [AWS Elemental MediaConvert error codes](#) that you can look up on the *MediaConvert User Guide* for details on how to address the issue.

If the error is not a MediaConvert job failure, possibly one of the two Lambda functions, `job_submit` or `job_complete`, encountered an error. The email you received has an `ErrorDetails` link that takes you directly to the CloudWatch logs generated by the failed function. The logs have additional details on why it failed.

Note

When overriding the sample `job-settings.json`, we recommend exporting job settings from a MediaConvert job that has successfully completed. Invalid encoding settings will result in the `job_submit` Lambda function to fail.

Appendix C: Collection of operational metrics

This solution includes an option to send anonymous operational metrics to AWS. We use this data to better understand how customers use this solution and related services and products. When enabled, the following information is collected and sent to AWS each time a video is processed:

- **Solution ID:** The AWS solution identifier.
- **Unique ID (UUID):** Randomly generated, unique identifier for each live streaming solution deployment.
- **Timestamp:** Data-collection timestamp.
- **Job Settings:** The job settings with the source and destination object paths removed. This helps us understand what output groups customers are looking for.

AWS owns the data gathered via this survey. Data collection will be 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

Visit our [GitHub repository](#) to download the templates and scripts for this solution, and to share your customizations with others.

Contributors

The following individuals contributed to this document:

- Tom Nightingale
- Joan Morgan

Revisions

Date	Change
November 2020	Initial release

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 current AWS 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. The responsibilities and liabilities of AWS 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.

The Video on Demand on AWS Foundation solution is licensed under the terms of the Apache License Version 2.0 available at [The Apache Software Foundation](#).