Video on Demand on AWS

Implementation Guide
# Table of Contents

Solution overview .................................................................................................................. 1
Cost ........................................................................................................................................... 2
  Example cost for a 60-minute source video ........................................................................ 2
Architecture ............................................................................................................................... 4
Components .............................................................................................................................. 5
  Encoding options .................................................................................................................. 5
    Quality-defined variable bitrate mode ............................................................................. 5
    Accelerated transcoding .................................................................................................... 5
    Frame capture .................................................................................................................... 5
  MediaPackage ....................................................................................................................... 5
  Error handling ..................................................................................................................... 6
  Ingest step functions .......................................................................................................... 6
    Video-only workflow ........................................................................................................ 6
    Metadata and video workflow .......................................................................................... 6
  Processing step functions .................................................................................................. 7
  Publishing step functions ................................................................................................... 7
Integration with AWS Service Catalog AppRegistry and AWS Systems Manager Application Manager .................................................................................................................. 7
Security ...................................................................................................................................... 8
  Amazon S3 bucket policy ..................................................................................................... 8
  IAM roles ............................................................................................................................. 8
Considerations ......................................................................................................................... 9
  Customization ..................................................................................................................... 9
  Solution updates ................................................................................................................ 9
  Regional deployments ....................................................................................................... 9
Template .................................................................................................................................... 10
Deployment ............................................................................................................................. 11
  Launch the stack ................................................................................................................ 11
Monitoring this solution with AppRegistry ........................................................................ 14
  Activate CloudWatch Application Insights ..................................................................... 14
  Activate AWS Cost Explorer ......................................................................................... 15
  Activate cost allocation tags associated with the solution ............................................. 16
Resources .................................................................................................................................. 17
Update the stack ..................................................................................................................... 18
Ingest workflow ..................................................................................................................... 19
Processing workflow ............................................................................................................ 20
Publishing workflow ............................................................................................................ 21
Workflow configuration ......................................................................................................... 22
Metadata file .......................................................................................................................... 23
MediaConvert templates ....................................................................................................... 24
Uninstall the solution ............................................................................................................. 25
  Using the AWS Management Console ........................................................................... 25
  Using AWS Command Line Interface ............................................................................. 25
  Deleting the Amazon S3 Buckets .................................................................................. 25
  Deleting the Amazon DynamoDB tables ......................................................................... 26
  Deleting the CloudWatch Logs ....................................................................................... 26
Operational metrics .................................................................................................................. 27
Source code ........................................................................................................................... 28
Revisions .................................................................................................................................. 29
Contributors .......................................................................................................................... 31
Notices ....................................................................................................................................... 32
AWS glossary .......................................................................................................................... 33
Deploy a reference implementation to build a scalable, distributed video-on-demand workflow

Publication date: April 2017 (last update (p. 29): July 2023)

The Video on Demand on AWS solution automatically provisions the AWS services necessary to build a scalable, distributed workflow to ingest, store, process, and deliver video content on demand. It ingests metadata files and source videos, processes the videos for playback on a wide range of devices, stores the transcoded media files, and delivers the videos to end users through CloudFront.

This solution uses the following AWS services to build a highly available and resilient architecture:

- **AWS Elemental MediaConvert** to transcode media files from their source format into versions that play back smartphones, tablets, PCs and other devices.
- **AWS Elemental MediaPackage** (optional) to create video streams formatted to play on several devices from a single video input.
- **Amazon CloudFront** to accelerate delivery of your video content to end users.
- **AWS Step Functions** to build applications from individual components that each perform a discrete function.
- **AWS Lambda** to run code without provisioning or managing servers.
- **Amazon Simple Storage Service** (Amazon S3) for object storage.
- **Amazon DynamoDB** to track source and destination file metadata and progress through the workflow.
- **Amazon CloudWatch** to track encoding jobs.
- **Amazon Simple Queue Service** (Amazon SQS) to capture the workflow output.
- **Amazon Simple Notification Service** (Amazon SNS) to send publishing, encoding, and error notifications.

This solution provides an example architecture to build a global consumer video workflow on AWS. By default, the solution can encode MP4, MPG, M4V, M2TS, and MOV files. You can customize the architecture to encode any media file type supported by AWS Elemental MediaConvert. For more information, refer to the section called “Customization” (p. 9).

This implementation guide discusses architectural considerations and configuration steps for deploying Video on Demand on AWS in the 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.
Cost

You are responsible for the cost of the AWS services used while running this solution, which can vary based on the following factors:

- The size of your videos.
- The number of outputs created.
- The number of views the published content receives through CloudFront.

AWS Elemental MediaConvert composes the majority of the cost. For more information about MediaConvert pricing, refer to [AWS Elemental MediaConvert Pricing](#).

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.

Example cost for a 60-minute source video

As of the latest revision, the estimated cost for using this solution with the default encoding settings to process a 60-minute source video in the US East (N. Virginia) Region is approximately $4.23. This estimate may vary depending on the source video size and format. The following test was run with a 1080p 60-minute input video file. This does not include Amazon Simple Storage Service (Amazon S3) storage costs, which vary depending on input file size. 4K video input will increase costs.

<table>
<thead>
<tr>
<th>AWS service</th>
<th>Dimensions</th>
<th>Cost [USD]</th>
</tr>
</thead>
<tbody>
<tr>
<td>MediaConvert</td>
<td>Output profile: AVC codec 1 pass quality 30 fps</td>
<td>$4.23</td>
</tr>
<tr>
<td>Amazon CloudFront</td>
<td>Using the price of $0.085 per GB for CloudFront. A 60-minute video with the default job settings streamed to 1,000 users would cost approximately: 0.75 MB/s * 1000 users * 3600 seconds ~= 2700 GB/hour. 2700 GB/hour * $0.085 = $229.50 an hour</td>
<td>$229.50</td>
</tr>
<tr>
<td>AWS Step Functions</td>
<td>Free tier cost is negligible even beyond free tier</td>
<td>$0.00</td>
</tr>
<tr>
<td>AWS Lambda</td>
<td>A 60-minute video will invoke around 24 lambda functions. When all free tier is used up, $0.0000002 * 24 requests = $0.0000048</td>
<td>$0.0000048</td>
</tr>
</tbody>
</table>
Example cost for a 60-minute source video

<table>
<thead>
<tr>
<th>AWS service</th>
<th>Dimensions</th>
<th>Cost [USD]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon S3</td>
<td>A 60-minute video will at most use 9GB of storage on S3 depending on the complexity of the video content. $0.023 per GB * 9GB = $0.207.</td>
<td>$0.21</td>
</tr>
<tr>
<td>Amazon DynamoDB</td>
<td>Free tier cost is negligible even beyond free tier</td>
<td>$0.00</td>
</tr>
<tr>
<td>Amazon CloudWatch</td>
<td>Free tier cost is negligible even beyond free tier</td>
<td>$0.00</td>
</tr>
<tr>
<td>Amazon Simple Queue Service (Amazon SQS)</td>
<td>Free tier cost is negligible even beyond free tier</td>
<td>$0.00</td>
</tr>
<tr>
<td>Amazon Simple Notification Service (Amazon SNS)</td>
<td>Free tier cost is negligible even beyond free tier</td>
<td>$0.00</td>
</tr>
</tbody>
</table>

**Total:** $233.94

(Optional) AWS Elemental MediaPackage $0.05

The following table lists the professional tier costs for the MediaConvert settings used in this example:

<table>
<thead>
<tr>
<th>Output</th>
<th>Cost [USD]</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD resolution</td>
<td>$0.0075/min</td>
</tr>
<tr>
<td>HD resolution</td>
<td>$0.024/min</td>
</tr>
</tbody>
</table>

Outputs with calculations:

HLS profile: 3 SD and 2 HD (3 x 60 x $0.0075) + (2 x 60 x $0.024)

Total cost = $1.35 + %2.88 = $4.23

The costs for CloudFront and Amazon S3 storage varies depending on the number and format of outputs created and the number of requests to view the content delivered through CloudFront. To calculate your average cost, use the data size of your output multiplied by the average number of viewers for your stream.
Architecture overview

The AWS CloudFormation template deploys a workflow that ingests source videos, or source videos and metadata files. When you upload a source video only, encoding options are defined in the AWS CloudFormation template at launch, and are applied to every video the solution encodes. When you upload a source video and metadata file, encoding parameters for each source video are defined in the metadata file, allowing customers to apply encoding options on a video-by-video basis.

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

1. An Amazon S3 bucket for source media files. Content is eventually stored in Amazon S3 Glacier, according to the Amazon S3 lifecycle policy.
2. AWS Step Functions, which creates ingest, processing, and publishing step functions.
3. AWS Elemental MediaConvert to transcode media files from their source format.
4. AWS Lambda functions that perform the work of each step, and process error messages. For a description of the 10 Lambda functions, refer to the README.md file on GitHub.
5. An Amazon DynamoDB table stores data captured through the workflow.
6. Amazon CloudWatch for logging and Amazon CloudWatch Events rules for AWS Elemental MediaConvert notifications.
7. Amazon SNS topics to send encoding, publishing, and error notifications.
8. AWS Elemental MediaPackage (optional) to create video streams formatted to play on several devices from a single video input.
9. An Amazon S3 bucket for storing destination media files.
10. An Amazon CloudFront distribution to deliver your video content to end users.
11. An Amazon SQS queue to capture the workflow outputs.

Video on Demand on AWS architecture

The AWS CloudFormation template deploys the following architecture:

1. An Amazon S3 bucket for source media files. Content is eventually stored in Amazon S3 Glacier, according to the Amazon S3 lifecycle policy.
2. AWS Step Functions, which creates ingest, processing, and publishing step functions.
3. AWS Elemental MediaConvert to transcode media files from their source format.
4. AWS Lambda functions that perform the work of each step, and process error messages. For a description of the 10 Lambda functions, refer to the README.md file on GitHub.
5. An Amazon DynamoDB table stores data captured through the workflow.
6. Amazon CloudWatch for logging and Amazon CloudWatch Events rules for AWS Elemental MediaConvert notifications.
7. Amazon SNS topics to send encoding, publishing, and error notifications.
8. AWS Elemental MediaPackage (optional) to create video streams formatted to play on several devices from a single video input.
9. An Amazon S3 bucket for storing destination media files.
10. An Amazon CloudFront distribution to deliver your video content to end users.
11. An Amazon SQS queue to capture the workflow outputs.
Solution components

Encoding options

The Video on Demand on AWS solution leverages AWS Elemental MediaConvert job templates to define the solution's encoding options. The solution can encode your source videos into H.264 and H.265; SD, HD, and 4K MP4; and SD and HD HTTP Live Streaming (HLS), and Dynamic Adaptive Streaming over HTTP (DASH). The workflow can be configured to encode all videos in the same or to use metadata files to apply encoding settings on a video-by-video basis.

The workflow selects one of the three templates, based on the resolution of the source video. You can also customize the solution to work with any valid MediaConvert template. For more information, refer to MediaConvert templates (p. 24).

Quality-defined variable bitrate mode

This solution leverages AWS Elemental MediaConvert Quality Variable Bit Rate (QVBR) encoding mode which ensures consistent, high-quality video transcoding with the smallest file size for any type of source video content. With QVBR, the encoder determines the right number of bits to use for each part of the video to maintain the video quality that you specify. If enabled, the solution configures the encoding templates to activate QVBR mode with the recommended settings for each output. For more information, refer to MediaConvert templates (p. 24).

Accelerated transcoding

This solution includes an option to activate Accelerated Transcoding in AWS Elemental MediaConvert, which increases the processing speed of file-based video encoding jobs by up to 25 times. For more information, refer to the MediaConvert documentation.

Important

There are 2 options to turn on acceleration. The Preferred option turns on acceleration, but falls back to standard encoding if the source file isn't supported. The Enabled option applies acceleration to every encoding job, and the job will fail if the source file is not supported.

Frame capture

This solution can also create a set of thumbnails from your source videos. If this feature is turned on, the solution will create a set of thumbnails for each selected output. The thumbnails are stored in the Amazon S3 bucket with your video output.

MediaPackage

This solution includes the option to use AWS Elemental MediaPackage as part of the workflow. When activated, the solution creates a separate set of MediaConvert custom templates that include H.265 MP4 and HLS. The solution also creates a packaging group in MediaPackage that is configured to ingest the MediaConvert HLS output stored in Amazon S3. MediaPackage packages the content, formatting it in response to playback requests from downstream devices. By default, this solution creates packaging configurations for HLS, DASH, MSS, and CMAF.
Important
Customers who ingest large quantities of files may exceed MediaPackage limits for video-on-demand content. For more information and instructions on how to request a limit increase, refer to VOD Content Limits in the AWS Elemental MediaPackage User Guide.

Error handling

The ingest, processing, and publishing workflow AWS Lambda functions, and Amazon CloudWatch Events are configured to invoke an error handler Lambda function that updates the Amazon DynamoDB table with error message details, and sends an Amazon Simple Notification Service (Amazon SNS) notification to a subscribed email address.

Ingest step functions

Video-only workflow

When a new MP4, MPG, M4V, M2TS, or MOV video is added to the source Amazon S3 bucket, a Lambda function invokes the ingest workflow. During ingestion, source video details are added to Amazon DynamoDB, the content is validated using MediaInfo, open-source software that displays technical information about media files, and details are stored in DynamoDB.

Important
Source video file extensions (.mp4, .mpg, .m4v, .m2ts or .mov) must be lowercase and file names cannot contain spaces.

Metadata and video workflow

When a new metadata file is added to the source Amazon S3 bucket, a Lambda function invokes the ingest workflow. During ingestion, the metadata file, source video, and encoding configuration details are added to Amazon DynamoDB, the source video is validated using MediaInfo and details are stored in DynamoDB.

Important
You must upload the source video file to the Amazon S3 bucket before you upload the metadata file. Note that the upload must complete before you upload the metadata file.
Processing step functions

The solution uses the height and width of the source video to determine which job template to use to submit encoding jobs to AWS Elemental MediaConvert. If you allow frame capture, the frame capture parameters are added to the job template. Then, the encoding job is created in MediaConvert and the details are stored in DynamoDB.

Publishing step functions

After the video is encoded, AWS Elemental MediaConvert sends a notification to Amazon CloudWatch. An Amazon CloudWatch Events rule invokes the publishing AWS Step Functions step function, which validates the outputs, and updates the DynamoDB table with the new content details.

When the workflow is finished, Amazon SNS and/or Amazon SQS sends a publish notification based on the configuration you choose. If you choose to archive your source content, the source files are tagged to allow the Amazon S3 lifecycle policy to move files to Amazon S3 Glacier or Amazon Deep Archive.

Integration with AWS Service Catalog AppRegistry and AWS Systems Manager Application Manager

This integration groups all the AWS resources in this solution as an Application using AWS Service Catalog AppRegistry, as well as governing the security policies in the context of an Application using AWS Systems Manager Application Manager integration. An application context includes a group of resources related to computing, networking, and storage resources. Moreover, customers can also define metadata specific to the application, including operational information, cost, and other dimensions.
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 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. 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 allow customers to assign granular access policies and permissions to services and users on the AWS Cloud. Video on Demand on AWS creates several IAM roles, including a role that grants AWS Elemental MediaConvert access to Amazon API Gateway and Amazon Simple Storage Service. This role is necessary to allow the services to operate in your account.
Design considerations

Customization

This solution leverages AWS Step Functions, which breaks the workflow into individual steps, making it easier to customize or extend the architecture for your specific video-on-demand needs. For example, you can modify or replace the encoding steps to produce different content sets. You can also add steps to extend support for more complex workflows, including image processing for poster artwork or additional custom data to the metadata file that will then be stored in Amazon DynamoDB. The solution originates AWS Elemental MediaConvert output content directly from Amazon Simple Storage Service (Amazon S3) through Amazon CloudFront. You can, however, customize the solution to leverage a dedicated origin server such as AWS Elemental MediaPackage.

Each time the workflow is initiated, the solution creates a unique identifier. The unique identifier is used as the primary key in Amazon DynamoDB and the run ID in AWS Step Functions. The unique identifier is passed to each step in the workflow, allowing information to be stored and retrieved in DynamoDB. This makes it easier to add and remove steps from the workflow.

Solution updates

To continue using this solution with the latest features and improvements, you must deploy the latest version of this stack. For information about updating your stack, refer to Update the stack (p. 18).

Installing version 5.3.0 creates three new MediaConvert job templates that only output HLS renditions to reduce cost, without the use of presets. For more information, refer to MediaConvert templates (p. 24). Updating an existing solution deployment to version 5.3.0 creates these new templates without deleting the presets or templates created by the older versions. To use an older template with the latest version of the solution, specify the template using the JobTemplate field in your metadata file. For more information, refer to Metadata file (p. 23). Or, you can replace the default templates in the Input Validate AWS Lambda function by modifying the MediaConvert_Template_<resolution> environment variables.

Regional deployments

This solution uses AWS Elemental MediaConvert and AWS Elemental MediaPackage, which are available in specific AWS Regions only. Therefore, you must deploy this solution in a Region that supports these services. For the most current service availability by Region, refer to the AWS Regional Services List.
This solution uses AWS CloudFormation to automate the deployment of the Video on Demand on AWS solution in the AWS Cloud. It includes the following AWS CloudFormation template, which you can download before deployment:

**video-on-demand-on-aws.template**: Use this template to launch the Video on Demand on AWS solution and all associated components. The default configuration deploys AWS Lambda functions, Amazon Simple Storage Service (Amazon S3) buckets, AWS Step Functions, AWS Elemental MediaConvert, AWS Elemental MediaPackage (optional), an Amazon DynamoDB table, 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 solution into your account.

To update your existing stack to the latest version, refer to Update the stack (p. 18). For additional options, including using an older version of this solution, refer to Solution updates (p. 9).

Time to deploy: Approximately 20 minutes

Launch the stack

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 through this survey. Data collection is subject to the AWS Privacy Notice.

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 Collection of operational metrics (p. 27) section of this guide.

Use this AWS CloudFormation template to deploy the latest version of this solution.

Note
You are responsible for the cost of the AWS services used while running this solution. Refer to the Cost (p. 2) section for more details. For full details, refer to 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 below to launch the video-on-demand-on-aws AWS CloudFormation template.

   ![Launch solution](image)

   You can also download the template as a starting point for your own implementation.

2. The template is launched 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 and AWS Elemental MediaPackage, which are available in specific AWS Regions only. Therefore, you must deploy this solution in a Region that supports these services. For the most current service availability by Region, refer to the AWS Regional Services List.

3. On the Select Template page, verify that you selected the correct template and choose Next.

4. Under Parameters, review the parameters for the template, and modify them as necessary.
This solution uses the following default values.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notification Email Address</td>
<td>&lt;Requires input&gt;</td>
<td>A valid email address to receive Amazon SNS notifications.</td>
</tr>
<tr>
<td>Workflow Trigger</td>
<td>&lt;Requires input&gt;</td>
<td>Choose VideoFile to ingest source videos only; choose MetadataFile to ingest metadata files and source videos.</td>
</tr>
<tr>
<td>Archive Source Content</td>
<td>false</td>
<td>Choose GLACIER to activate an Amazon S3 lifecycle policy on the source bucket to move applicable files to Amazon S3 Glacier after seven days, DEEP_ARCHIVE to move to Glacier Deep Archive.</td>
</tr>
<tr>
<td>Enable SNS</td>
<td>true</td>
<td>Choose true receive SNS notifications for the ingest and pushlish workflows. Choose false to only receive error messages.</td>
</tr>
<tr>
<td>Enable SQS</td>
<td>true</td>
<td>Choose true to deploy an SQS queue for publishing messages.</td>
</tr>
<tr>
<td>Enable Frame Capture</td>
<td>false</td>
<td>Choose true to create thumbnails for each AWS Elemental MediaConvert output.</td>
</tr>
<tr>
<td>Accelerated Transcoding</td>
<td>PREFERRED</td>
<td>Choose PREFERRED to activate Accelerated Transcoding for supported file types (recommended), ENABLE to apply to all encoding jobs.</td>
</tr>
<tr>
<td>Enable MediaPackage</td>
<td>false</td>
<td>Choose true to activate AWS Elemental MediaPackage as part of the workflow.</td>
</tr>
</tbody>
</table>

5. Choose Next.
6. On the Options page, choose Next.
7. On the Review page, review and confirm the settings. Be sure to check the box acknowledging that the template will create AWS Identity and Access Management (IAM) resources.
8. 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.

   After the stack is created, Amazon SNS sends three subscription notifications to the admin email address with links to activate encoding, publishing, and error notification.
9. In the subscription notification emails, select each link to activate SNS notifications.
Note
In addition to the AWS Lambda functions that create solution resources and trigger the ingest and publishing processes, 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 as it is necessary to manage associated resources.
Monitoring this solution with AppRegistry

The Video on Demand 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 Video on Demand on AWS stack in Application Manager.

![Video on Demand on AWS stack in Application Manager](image)

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.

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.
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:

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.
## Additional resources

### AWS services

<table>
<thead>
<tr>
<th>Service</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS Elemental MediaConvert</td>
<td>Amazon DynamoDB</td>
</tr>
<tr>
<td>AWS Elemental MediaPackage</td>
<td>Amazon CloudWatch</td>
</tr>
<tr>
<td>Amazon CloudFront</td>
<td>Amazon SQS</td>
</tr>
<tr>
<td>AWS Step Functions</td>
<td>Amazon SNS</td>
</tr>
<tr>
<td>AWS Lambda</td>
<td>AWS CloudFormation</td>
</tr>
<tr>
<td>Amazon S3</td>
<td>AWS Identity and Access Management (IAM)</td>
</tr>
<tr>
<td><strong>AWS Cloud Development Kit (AWS CDK)</strong></td>
<td><strong>AWS Systems Manager</strong></td>
</tr>
</tbody>
</table>

### Other resources

- MediaInfo
Update the stack

Complete the following steps to update your AWS CloudFormation stack to the current version.

1. From the main account where the Video on Demand on AWS template is deployed, sign in to the AWS CloudFormation console.
2. From the Stacks page, select this solution’s stack and choose Update.
3. On the Update stack page, verify that Replace current template is selected.
   a. In the Specify template section, select Amazon S3 URL.
   b. Copy the link of the latest template.
   c. Paste the link in the Amazon S3 URL box.
   d. Verify that the correct template URL shows in the Amazon S3 URL text box and choose Next.
4. On the Specify stack details page, under Parameters, review the parameters for the template and modify them as necessary. Refer to Launch the stack (p. 11) for details about the parameters.
5. Choose Next.
6. On the Configure stack options page, choose Next.
7. 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.
8. Choose View change set and verify the changes.
9. Choose Update stack to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the Status column. You should receive an UPDATE_COMPLETE status in approximately 20 minutes.
Ingest workflow

When a new video is added to the source Amazon Simple Storage Service (Amazon S3) bucket, an AWS Lambda function starts the ingest step function. The ingest step function includes:

- **Input Validate** - Parses the input to the workflow, checks for the source video file, and defines the workflow configuration using the AWS Lambda function environment variables. If turned on, this step downloads the metadata file and overwrites the default environment variables with the variable definitions in the metadata file (metadata and video version only). For more information, refer to Metadata file (p. 23).
- **MediaInfo** - Generates a signed Amazon S3 URL for the source video and runs MediaInfo to extract metadata about the video.
- **DynamoDB Update** - Takes accumulated data from each step and stores it in Amazon DynamoDB.
- **SNS Notification** - Sends an Amazon Simple Notification Service (Amazon SNS) notification with a summary of the ingest process.
- **Process Execute** - Starts the processing workflow.
Processing workflow

When the ingest workflow is complete, it starts the processing workflow. The processing workflow includes:

- **Profiler** – Gets the source video’s height and width from the metadata file, defines the settings for frame capture (if turned on), and chooses which template to use for encoding based on the source video’s height. For example, if the source video is greater than or equal to 1080p, the 1080p job template will be used.

- **Encoding Profile Check, Accelerated Transcoding Check, and Frame Capture Check** – Helps visualize which settings the profiler step applied.

- **Encode Job Submit** – Submits the encoding job with the template defined by the profiler to AWS Elemental MediaConvert.

- **DynamoDB Update** – Takes accumulated data from each step and stores it in Amazon DynamoDB.
Publishing workflow

When encoding is complete, Amazon Simple Notification Service (Amazon SNS) sends a notification that invokes an AWS Lambda function that starts the publishing process. The publishing process includes:

- **Output Validate** - Checks the event data for the completed encoding job, gets the GUID from the AWS Elemental MediaConvert notification, gets the asset details from Amazon DynamoDB, and generates the Amazon Simple Storage Service (Amazon S3) and Amazon CloudFront URLs for the MediaConvert outputs.
- **Archive Choice** - If Glacier or Glacier Deep Archive was activated, this step tags the source video with a unique identifier and the archive to invoke the Amazon S3 Glacier lifecycle policy.
- **MediaPackage Choice** - If you configure the solution to use AWS Elemental MediaPackage, this step takes the output from MediaConvert and uses it as a source for a MediaPackage asset, which contains all the information MediaPackage requires to ingest file-based video content.
- **DynamoDB Update** - Updates Amazon DynamoDB table with the event data.
- **SQS Choice** – If activated, this step sends all workflow outputs to an SQS queue that is ingested into upstream workflows or processes.
- **SNS Choice** – If activated, this step sends an Amazon SNS notification with a summary of the workflow and the Amazon CloudFront URLs.
Workflow configuration

The Input Validate AWS Lambda function contains the following environment variables that define the workflow configuration.

<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Archive Source</td>
<td>Choose whether to archive source videos in Amazon Glacier or Glacier Deep Archive.</td>
</tr>
<tr>
<td>CloudFront</td>
<td>The Amazon CloudFront domain name. This is used to generate the playback URLs for the AWS Elemental MediaConvert outputs.</td>
</tr>
<tr>
<td>Destination</td>
<td>The name of the destination Amazon S3 bucket for all MediaConvert outputs.</td>
</tr>
<tr>
<td>FrameCapture</td>
<td>Choose whether to create thumbnails for each MediaConvert output.</td>
</tr>
<tr>
<td>MediaConvert_Template_2160p</td>
<td>The name of the UHD template for MediaConvert.</td>
</tr>
<tr>
<td>MediaConvert_Template_1080p</td>
<td>The name of the HD template for MediaConvert.</td>
</tr>
<tr>
<td>MediaConvert_Template_720p</td>
<td>The name of the SD template for MediaConvert.</td>
</tr>
<tr>
<td>Source</td>
<td>The name of the source Amazon S3 bucket.</td>
</tr>
<tr>
<td>WorkflowName</td>
<td>Used to tag MediaConvert encoding jobs. This is defined by the AWS CloudFormation stack name.</td>
</tr>
<tr>
<td>InputRotate</td>
<td>Specify how MediaConvert should rotate the source video.</td>
</tr>
<tr>
<td>AcceleratedTranscoding</td>
<td>The option to activate Accelerated Transcoding in MediaConvert.</td>
</tr>
<tr>
<td>EnableSQS</td>
<td>The option to activate SQS.</td>
</tr>
<tr>
<td>EnableSNS</td>
<td>The option to activate SNS.</td>
</tr>
</tbody>
</table>

These variables are set when you deploy the AWS CloudFormation template and apply to all source videos uploaded to the solution’s Amazon Simple Storage Service (Amazon S3) bucket.

If you set the solution to ingest source videos and metadata files, you can overwrite these files using a metadata file. For more information, refer to MediaConvert templates (p. 24).
Metadata file

When you set the solution to ingest source videos and metadata files, the source Amazon Simple Storage Service (Amazon S3) bucket is configured with an event notification that invokes the workflow when you upload a JSON file.

**Note**
The JSON file does not need a specific name, but it must have the JSON file extension. We recommend naming the JSON file the same name as the video file for consistency and ease of reference.

To invoke the workflow, you must upload a JSON metadata file. If you only upload a source video file, the workflow will not start.

**Important**
You must upload the source video file to the Amazon S3 bucket before you upload the metadata file. Note that the upload must complete before you upload the metadata file.

The definitions in the metadata file overwrite the default settings you specified when you deployed the solution. This allows you to define different workflow configurations for each source video. If you do not specify a definition in the metadata file, the solution will use the default value you set during deployment. Note that the metadata file must include a definition for `srcVideo`.

The following example metadata files shows all available variable definitions.

```json
{
  "srcVideo": "string",
  "archiveSource": "DISABLED|GLACIER|DEEP_ARCHIVE",
  "frameCapture": boolean,
  "srcBucket": "string",
  "destBucket": "string",
  "cloudFront": "string",
  "jobTemplate_2160p": "string",
  "jobTemplate_1080p": "string",
  "jobTemplate_720p": "string",
  "acceleratedTranscoding": "DISABLED|PREFERRED|ENABLED",
  "enableSqs": boolean,
  "enableSns": boolean,
  "jobTemplate": "custom-job-template",
  "InputRotate": "DEGREE_0|DEGREES_90|DEGREES_180|DEGREES_270|AUTO"
}
```

The following sample JSON metadata file will overwrite the default settings for the **Archive Source Content** and **Enable Frame Capture** AWS CloudFormation template parameters for the `example.mpg` file. The file will also set the job template for AWS Elemental MediaConvert to `custom-job-template`.

```json
{
  "srcVideo": "example.mpg",
  "archiveSource": "GLACIER",
  "frameCapture": false,
  "jobTemplate": "custom-job-template"
}
```

The Video on Demand on AWS solution also supports adding additional metadata, such as title, genre, or any other information, you want to store in Amazon DynamoDB.
MediaConvert templates

The Video on Demand on AWS solution outputs 4K, 1080p, and 720p MP4, and any combination of 1080p, 720p, 540p, 360p, and 270p HLS and DASH. By default, the solution selects the job template for AWS Elemental MediaConvert based on the source video height. The solution includes three default job templates:

- MediaConvert_Template_2160p: 5 HLS outputs AVC 2160p through 270p
- MediaConvert_Template_1080p: 5 HLS outputs AVC 1080p through 270p
- MediaConvert_Template_720p: 4 HLS outputs AVC 720p through 270p

By default, the solution is configured to leverage Quality-Defined Variable Bitrate (QVBR) mode in MediaConvert. The QVBR settings are configured to the recommended values for each output, as shown in the following table.

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Maximum Bitrate</th>
<th>QVBR Quality Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>2160p</td>
<td>15,000 Kbps</td>
<td>9</td>
</tr>
<tr>
<td>1080p</td>
<td>8,500 Kbps</td>
<td>8</td>
</tr>
<tr>
<td>720p</td>
<td>6,000 Kbps</td>
<td>8</td>
</tr>
<tr>
<td>540p</td>
<td>3,500 Kbps</td>
<td>7</td>
</tr>
<tr>
<td>360p</td>
<td>1,500 Kbps</td>
<td>7</td>
</tr>
<tr>
<td>270p</td>
<td>400 Kbps</td>
<td>7</td>
</tr>
</tbody>
</table>

You can also modify the solution to use different QVBR settings, other system job templates, or your own custom job templates. For more information about working with job templates for MediaConvert, refer to Working with MediaConvert Job Templates. For more information about QVBR Mode, refer to Using the QVBR Rate Control Mode.

If you set the solution to ingest source videos and metadata files, you can specify the template using the JobTemplate field in your metadata file. For more information, refer to Metadata file (p. 23). Or, you can replace the default templates in the Input Validate AWS Lambda function by modifying the MediaConvert_Template_<resolution> environment variables.
Uninstall the solution

You can uninstall the Video on Demand on AWS solution from the AWS Management Console, or by using the AWS Command Line Interface (AWS CLI). You must manually delete the Amazon Simple Storage Service (Amazon S3) buckets, an Amazon DynamoDB table, and CloudWatch Logs created by this solution. AWS Solutions Implementations do not automatically delete these resources in case you have stored data that you wish to retain.

Note
AWS CloudFormation StackSets are automatically deleted when you uninstall the solution's stack.

Using the AWS Management Console

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

Using AWS Command Line Interface

Verify that 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 the AWS CLI is available, run the following command.

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

Replace `<your-stack-name>` with the name of your CloudFormation stack.

Deleting the Amazon S3 buckets

This solution is configured to retain the Amazon S3 buckets if you decide to delete the AWS CloudFormation stack to prevent against 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 one of the S3 buckets and choose Delete.

Repeat the steps until you have deleted all the `<stack-name>` S3 buckets.

To delete the S3 buckets using AWS CLI, run the following command:

```
$ aws s3 rb s3://<bucket-name> --force
```
Alternatively, you can configure the AWS CloudFormation template to delete the Amazon S3 buckets automatically. Prior to deleting the stack, change the deletion behavior in the AWS CloudFormation DeletionPolicy attribute.

Deleting the Amazon DynamoDB tables

This solution is configured to retain the DynamoDB tables if you decide to delete the AWS CloudFormation stack to prevent accidental data loss. After uninstalling the solution, you can manually delete the DynamoDB tables if you do not need to retain the data. Follow these steps:

1. Sign in to the Amazon DynamoDB console.
2. Choose Tables from the left navigation pane.
3. Select the `<stack-name>` table and choose Delete.

To delete the DynamoDB tables using AWS CLI, run the following command:

```bash
$ aws dynamodb delete-table <table-name>
```

Deleting the CloudWatch Logs

This solution retains the CloudWatch Logs if you decide to delete the AWS CloudFormation stack to prevent against accidental data loss. After uninstalling the solution, you can manually delete the logs if you do not need to retain the data. Follow these steps to delete the CloudWatch Logs.

1. Sign in to the Amazon CloudWatch console.
2. Choose Log Groups from the left navigation pane.
3. Locate the log groups created by the solution.
4. Select one of the log groups.
5. Choose Actions and then choose Delete.

Repeat the steps until you have deleted all the solution log groups.
Collection of operational metrics

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 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
- **Use Glacier** - Whether Amazon Glacier is used
- **Workflow Trigger** - The workflow trigger selected
- **Frame Capture** - Whether thumbnails are created for AWS Elemental MediaConvert output
- **Enable MediaPackage** - Whether AWS Elemental MediaPackage is enabled

AWS owns the data gathered through 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 CloudFormation template with a text editor.
3. Modify the 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, select Choose file and select the edited template from your local drive.
8. Choose Next and follow the steps in the section called “Launch the stack” (p. 11) in the Automated Deployment section of this guide.
Source code

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

<table>
<thead>
<tr>
<th>Date</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>April 2017</td>
<td>Initial release</td>
</tr>
<tr>
<td>June 2017</td>
<td>Added information on new watermark functionality and processing workflow; updated information on the ingest and publishing workflows.</td>
</tr>
<tr>
<td>October 2017</td>
<td>Added information on new metadata and video and video-only AWS CloudFormation templates and functionality; updated information on the ingest process and error handling.</td>
</tr>
<tr>
<td>March 2018</td>
<td>Added AWS MediaConvert functionality.</td>
</tr>
<tr>
<td>November 2018</td>
<td>Added information on new AWS Elemental MediaConvert job templates and functionality, and environment variables; removed Amazon Elastic Transcoder functionality.</td>
</tr>
<tr>
<td>April 2019</td>
<td>Added information on support for AWS Elemental MediaConvert Quality-Defined Variable Bitrate (QVBR) Mode.</td>
</tr>
<tr>
<td>November 2019</td>
<td>Added information on support for AWS Elemental MediaPackage, and Node.js and MediaInfo updates.</td>
</tr>
<tr>
<td>April 2020</td>
<td>Added information about Accelerated Transcoding and notification options.</td>
</tr>
<tr>
<td>December 2020</td>
<td>Release version 5.2.0: Added information about the new AWS Elemental MediaConvert templates, installation and upgrade behavior using latest solution version. For more information, refer to the CHANGELOG.md file in the GitHub repository.</td>
</tr>
<tr>
<td>September 2021</td>
<td>Document enhancements, including updates to diagrams and architecture overview section.</td>
</tr>
<tr>
<td>November 2021</td>
<td>Release version 5.3.0: Added new input file formats, reduced cost by removing Dash and MP4 outputs from AWS Elemental MediaConvert templates, changed frame rate to follow source, deinterlacer not activated by default allowing for basic tier pricing. For more information, refer to the CHANGELOG.md file in the GitHub repository.</td>
</tr>
<tr>
<td>December 2021</td>
<td>Release version 5.3.1: For more information, refer to the CHANGELOG.md file in the GitHub repository.</td>
</tr>
<tr>
<td>Date</td>
<td>Change</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>November 2022</td>
<td>Release version 6.0.0: Added 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. You can now manage costs, view logs, implement patching, and run automation runbooks for this solution from a central location. For more information, refer to the CHANGELOG.md file in the GitHub repository.</td>
</tr>
<tr>
<td>February 2023</td>
<td>Release version 6.1.0: For more information, refer to the CHANGELOG.md file in the Github repository.</td>
</tr>
<tr>
<td>April 2023</td>
<td>Release version 6.1.1: 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.</td>
</tr>
<tr>
<td>May 2023</td>
<td>Release version 6.1.2: Updated cache policy name to be unique for blueprint, updated stack name and logical ID for AppRegistry, added package-lock.json files to packages, and updated Lambda nodes to Node.js 16. For more information, refer to the CHANGELOG.md file in the Github repository.</td>
</tr>
<tr>
<td>July 2023</td>
<td>Release version 6.1.3: Upgraded to Node.js to 18 and JS SDK to v3. Updated S3 client to generate presigned URLs using v4 signature by default. Updated parameter names for consistency. Bug fixes. For more information, refer to the CHANGELOG.md file in the GitHub repository.</td>
</tr>
</tbody>
</table>
Contributors

The following individuals contributed to this document:

- Eddie Goynes
- Tom Nightingale
- Joan Morgan
- San Dim Ciin
- Eric Thoman
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.

Video on Demand on AWS is licensed under the terms of the Apache License Version 2.0 available at The Apache Software Foundation.
AWS glossary

For the latest AWS terminology, see the AWS glossary in the AWS Glossary Reference.