# Table of Contents

Solution overview ................................................................................................................................. 1  
Cost ......................................................................................................................................................... 2  
  Cost table for a 60-minute source video ............................................................................................... 2  
Architecture overview ............................................................................................................................. 4  
Solution components ............................................................................................................................... 5  
  Ingest ..................................................................................................................................................... 5  
  Encoding ............................................................................................................................................... 5  
  Processing ............................................................................................................................................ 6  
Security .................................................................................................................................................... 7  
  Amazon S3 bucket policy ....................................................................................................................... 7  
  IAM roles ............................................................................................................................................. 7  
Considerations .......................................................................................................................................... 8  
  Customization ...................................................................................................................................... 8  
  Regional deployments ............................................................................................................................ 8  
Template .................................................................................................................................................. 9  
Automated deployment ............................................................................................................................. 10  
  Launch the stack .................................................................................................................................. 10  
Monitoring the solution with AppRegistry ............................................................................................ 12  
  Activate CloudWatch Application Insights ............................................................................................ 12  
  Activate AWS Cost Explorer .................................................................................................................. 13  
  Activate cost allocation tags associated with the solution ..................................................................... 14  
Changing the job settings file .................................................................................................................. 15  
Working with multiple job settings files .................................................................................................. 16  
Troubleshooting ....................................................................................................................................... 17  
Operational metrics .................................................................................................................................... 18  
Additional resources ............................................................................................................................... 19  
Source code ............................................................................................................................................. 20  
Revisions .................................................................................................................................................. 21  
Contributors ........................................................................................................................................... 22  
Notices ..................................................................................................................................................... 23
Deploy a customizable architecture to build a video-on-demand workflow

Publication date: December 2020 (last update (p. 21): May 2023)

Video on Demand on AWS Foundation is a reference implementation that automatically provisions the Amazon Web Services (AWS) services necessary to build a scalable, distributed video-on-demand workflow. To build highly available, resilient architectures that ingest, store, process, and deliver video content on demand, this solution uses the following AWS services:

- AWS Elemental MediaConvert to transcode media files from their source format into versions that play back on devices like smartphones, tablets, and PCs, and other devices.
- Amazon CloudFront for global distribution.
- Amazon Simple Storage Service (Amazon S3) for object storage.
- AWS Lambda to run code without provisioning or managing servers.
- Amazon CloudWatch to track encoding jobs in MediaConvert.
- Amazon Simple Notification Service (Amazon SNS) to send notifications for completed jobs.

This solution is designed to help you 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. 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 Customization (p. 8).

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.

This implementation guide discusses architectural considerations and configuration steps for deploying the Video on Demand on AWS Foundation solution 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. 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 Amazon CloudFront. After encoding all of your videos with this solution, the monthly cost will be for Amazon S3 storage, and any CloudFront costs from streaming your new video content to users.

As of the most recent revision, the estimated cost of running this solution for a 60-minute video with the dimensions listed in the following table, in the US East (N. Virginia) Region is approximately **$232.86 per month, per job**.

#### Cost table for a 60-minute source video

<table>
<thead>
<tr>
<th>AWS service</th>
<th>Dimensions</th>
<th>Cost per month [USD]</th>
</tr>
</thead>
<tbody>
<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>Amazon S3</td>
<td>A 60-minute video will at most use 9 GB of storage on S3 depending on the complexity of the video content. $0.023 per GB * 9 GB = $0.207. <strong>Note</strong> Source videos uploaded to Amazon S3 will add to this cost. After MediaConvert processing, delete source content from S3 to save storage costs.</td>
<td>$ 0.207</td>
</tr>
<tr>
<td>AWS Lambda</td>
<td>4 requests per file using $0.20 per million requests.</td>
<td>$ 0.0000024</td>
</tr>
<tr>
<td>Amazon CloudWatch</td>
<td>Free tier. Check the CloudWatch pricing page for more information.</td>
<td>$ 0.00</td>
</tr>
</tbody>
</table>
### AWS service

<table>
<thead>
<tr>
<th>AWS service</th>
<th>Dimensions</th>
<th>Cost per month [USD]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon Simple Notification Service (Amazon SNS)</td>
<td>Free tier. Check the Amazon SNS pricing page for more information.</td>
<td>$ 0.00</td>
</tr>
</tbody>
</table>
| AWS Elemental MediaConvert                       | HLS output profile:  
- 3 SD resolution at 30 fps or less  
- 2 HD resolution at 30 fps or less.                                                                               | $ 3.15               |
| **Total:**                                       |                                                                                                                      | **$232.86**           |

A significant cost of running this solution comes from MediaConvert. This section breaks down the MediaConvert costs for the sample 60-minute source video.

The video outputs in this example use the following basic tier MediaConvert settings: AVC codec, 1 pass quality, and 30 fps. The CloudFormation template creates a destination S3 bucket where the processed video are stored. Each processed result will be stored in a folder with the same name as the process execution id and a previously processed video can be found here. Pricing increases when using higher frame rates than 30 FPS. 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, see the pricing webpage for each AWS service used in this solution.
Deploying this solution builds the following environment in the AWS Cloud:

1. An **Amazon Simple Storage Service** (Amazon S3) bucket to store source video files. A sample job settings file, used to define the encoding settings for MediaConvert, is uploaded to the source S3 bucket.
2. An **AWS Lambda** job submit function to create the encoding jobs in **AWS Elemental MediaConvert**.
3. **MediaConvert** transcodes the video into HLS Adaptive Bitrate files.
4. **Amazon CloudWatch** tracks encoding jobs in MediaConvert and invokes the Lambda job complete function.
5. A Lambda job complete function to process the outputs.
6. An **Amazon Simple Notification Service** (Amazon SNS) topic sends notifications of completed jobs.
7. A destination S3 bucket to store the outputs from MediaConvert.
8. **Amazon CloudFront** is configured with the destination S3 bucket as the origin for global distribution of the transcoded video content.
Solution components

Ingest

To invoke 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 clientAWS 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.

**Note**
Uploading a video to the root of the source S3 bucket causes the job_submit Lambda function to fail. You must upload videos to a folder within the root, and the folder must contain a job-settings.json 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 solution supports 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.

By default, the Video on Demand on AWS Foundation solution includes a sample encoding job settings file that encodes your source videos into MP4, SD and HD formats of HTTP Live Streaming (HLS), and SD and HD formats of Dynamic Adaptive Streaming over HTTP (DASH). You can overwrite the provided encoding job settings with your own to output whatever format you need that's supported by AWS Elemental MediaConvert.

The sample job settings file created as part of the AWS CloudFormation deployment has the QVBR rate control activated with accelerated transcoding set to PREFERRED and generates the following output:

- HLS 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.
A CloudWatch Events rule is configured to invoke 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.

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

Video on Demand on AWS Foundation jobs manifest file

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.
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 allow 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.
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. For details, refer to Working with multiple job settings files (p. 16).

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 the AWS Regional Services Link.
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:

**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

**Important**
This solution includes an option to send anonymized operational metrics to AWS. We use this data to better understand how customers use this solution and related services and products. AWS owns the data gathered though this survey. Data collection is subject to the [AWS Privacy 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 [Operational metrics](#) section of this guide.

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. For detailed, refer to the [Cost](#) section of this guide, and refer to the pricing webpage for each AWS service used in this solution.

1. Sign in to the AWS Management Console and select the button to launch the `video-on-demand-on-aws-foundation.template` AWS CloudFormation template.

   ![Launch solution](#)

   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 the [AWS Regional Services List](#).

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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notification Email Address</td>
<td><code>&lt;Requires input&gt;</code></td>
<td>A valid email address to receive Amazon SNS notifications.</td>
</tr>
</tbody>
</table>
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 allow encoding, publishing, and error notifications.

10. In the subscription notification emails, select each link to allow 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.
Monitoring the solution with AppRegistry

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

Video on Demand on AWS Foundation stack in Application Manager

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

Activate CloudWatch Application Insights

1. Sign in to the Systems Manager console.
2. In the navigation pane, choose Application Manager.
3. In Applications, choose AppRegistry applications.
4. In AppRegistry applications, search for the application name for this solution and select it.
The next time you open Application Manager, you can find the new application for your solution in the **AppRegistry application** category.

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

### 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:

2. In the navigation pane, select **Cost Explorer**.
3. On the **Welcome to Cost Explorer** page, choose **Launch Cost Explorer**.
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.
Changing the job settings file

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 invokes a workflow to apply job settings to the job created in AWS Elemental MediaConvert.

This job-settings.json file can be customized, or replaced with a new MediaConvert job template. To make a new job template or to customize the existing job-settings.json job template, refer to Working with AWS Elemental MediaConvert job templates in the AWS MediaConvert User Guide.

When your updated job template json file is ready, name the file job-settings.json and upload it to replace the one that is already in the assets01 folder of the Amazon S3 source bucket.
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 invokes a workflow to apply job settings to the job created in AWS Elemental MediaConvert.

To support different job settings, create additional folders at the root of the source S3 bucket and include different job setting 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 launches 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.
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. Incorrect encoding settings will result in the job_submit Lambda function to fail.
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 activated, 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 Notice](#). To opt out of this feature, modify the AWS CloudFormation template mapping section from:

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

to:

```yaml
AnonymizedData:
  SendAnonymizedData:
    Data: No
```
Additional resources

**AWS services**

<table>
<thead>
<tr>
<th>AWS service</th>
<th>AWS service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon CloudFront</td>
<td>AWS CloudFormation</td>
</tr>
<tr>
<td>Amazon CloudWatch</td>
<td>AWS Elemental MediaConvert</td>
</tr>
<tr>
<td>Amazon Simple Notification Service</td>
<td>AWS Lambda</td>
</tr>
<tr>
<td>Amazon Simple Storage Service</td>
<td>AWS Systems Manager</td>
</tr>
</tbody>
</table>
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>November 2020</td>
<td>Initial release</td>
</tr>
<tr>
<td>July 2021</td>
<td>Release version 1.1.0</td>
</tr>
<tr>
<td></td>
<td>• Added new input file extensions .wmv, .mxf, .mkv, .m3u8, .mpeg, .webm, and .h264. All file extensions now work in uppercase or lowercase format. For example, both .WMV and .wmv file extensions now trigger jobs through S3.</td>
</tr>
<tr>
<td></td>
<td>• Updates to job-settings.json from Professional tier to Basic tier to reduce cost by 37%.</td>
</tr>
<tr>
<td></td>
<td>• Default job-settings.json frames per second set to follow source instead of setting a strict 30 fps.</td>
</tr>
<tr>
<td></td>
<td>For more information, refer to the <a href="#">CHANGELOG.MD</a> file in the GitHub repository.</td>
</tr>
<tr>
<td>October 2022</td>
<td>Release version 1.2.0: Added a Service Catalog AppRegistry resource to register the CloudFormation template and underlying resources as an application in both 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.</td>
</tr>
<tr>
<td></td>
<td>For more information, refer to the <a href="#">CHANGELOG.MD</a> file in the GitHub repository.</td>
</tr>
<tr>
<td>October 2022</td>
<td>Minor changes to the AppRegistry section.</td>
</tr>
<tr>
<td></td>
<td>For more information, refer to the <a href="#">CHANGELOG.MD</a> file in the GitHub repository.</td>
</tr>
<tr>
<td>April 2023</td>
<td>Release version 1.2.1: Mitigated impact caused by new default settings for S3 Object Ownership (ACLs disabled) for all new S3 buckets.</td>
</tr>
<tr>
<td></td>
<td>For more information, refer to the <a href="#">CHANGELOG.MD</a> file in the GitHub repository.</td>
</tr>
<tr>
<td>May 2023</td>
<td>Release version 1.3.0: Added package_lock.json files to packages. Upgraded to CDK V2, and updated Node.js to 16. Updated parameter names for consistency.</td>
</tr>
<tr>
<td></td>
<td>For more information, refer to the <a href="#">CHANGELOG.MD</a> file in the GitHub repository.</td>
</tr>
</tbody>
</table>
Contributors

The following individuals contributed to this document:

- Tom Nightingale
- Joan Morgan
- Eddie Goynes
- San Dim Ciin
- David Chung
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.