Implementation Guide

Video on Demand on AWS Foundation

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Video on Demand on AWS Foundation: Implementation Guide

Copyright © 2024 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

Deploy a customizable architecture to build a video-on-demand workflow

Publication date: *December 2020 (last update: November 2024)*

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 smartphones, tablets, 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 EventBridge to invoke the Lambda job complete function.
- Amazon CloudWatch to log metrics about encoding jobs in MediaConvert.
- Amazon Simple Notification Service (Amazon SNS) to send notifications for completed jobs.

We designed this solution to help you begin encoding video files with MediaConvert. Out of the box, this solution provides a sample MediaConvert `job-settings.json` file, which you can use 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 MediaConvert. For more information, refer to Customization.

If you want 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 provides an overview of the Video on Demand on AWS Foundation solution, its reference architecture and components, considerations for planning the deployment, configuration steps for deploying the solution to the AWS Cloud.

The intended audience for using this solution's features and capabilities in their environment includes IT infrastructure architects, solution architects, administrators, business decision makers, DevOps engineers, data scientists, and cloud professionals.

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

If you want to . . .	Read . . .
Know the cost for running this solution. The estimated cost for running this solution in the US East (N. Virginia) Region is approxima tely $232.86 per month, per job for AWS resources.	Cost
Understand the security considerations for this solution.	Security
Know how to plan for quotas for this solution.	Quotas
Know which AWS Regions support this solution.	Supported AWS Regions
View or download the AWS CloudForm ation template included in this solution to automatically deploy the infrastructure resources (the "stack") for this solution.	AWS CloudFormation template
Access the source code and optionally use the AWS Cloud Development Kit (AWS CDK) to deploy the solution.	GitHub repository

Features and benefits

The solution provides the following features:

Reference implementation

Leverage this solution as a reference implementation to automatically provision the AWS services necessary to build a scalable, distributed video-on-demand workflow.

Customization

You can customize this solution and then use it as the starting point to create a more complex workflow.

Integration with Service Catalog AppRegistry and Application Manager, a capability of AWS Systems Manager

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

Use cases

Streaming media

As consumer demand for video streaming increases, media and entertainment companies are looking for secure and reliable web-based video streaming alternatives to traditional television. This solution automatically provisions the services necessary to build a scalable, distributed architecture that ingests, stores, processes, and delivers video content. Using this solution, you can avoid inefficient trial-and-error approaches, and save on time and costs for your streaming media projects.

Educational content delivery

Professional development and educational initiatives create incentives and can be important revenue generators for nonprofit organizations. This solution can help you create modern, scalable content delivery and learning management systems (LMS) to support your membership and programming offerings. The solution streamlines the processes for delivering online training and learning content.

Concepts and definitions

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

application

A logical group of AWS resources that you want to operate as a unit.

Dynamic Adaptive Streaming over HTTP (DASH)

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

HTTP Live Streaming (HLS)

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

Quality-Defined Variable Bitrate (QVBR)

A video encoding technology that uses fewer bits in low-complexitiy periods of content, and more bits during high-completely periods (up to the maximum bitrate), delivering consistently high video quality. For more information, see What is Quality-Defined Variable Bitrate (QVBR)?

workflow

Generated state machines that run a number of operations in sequence.

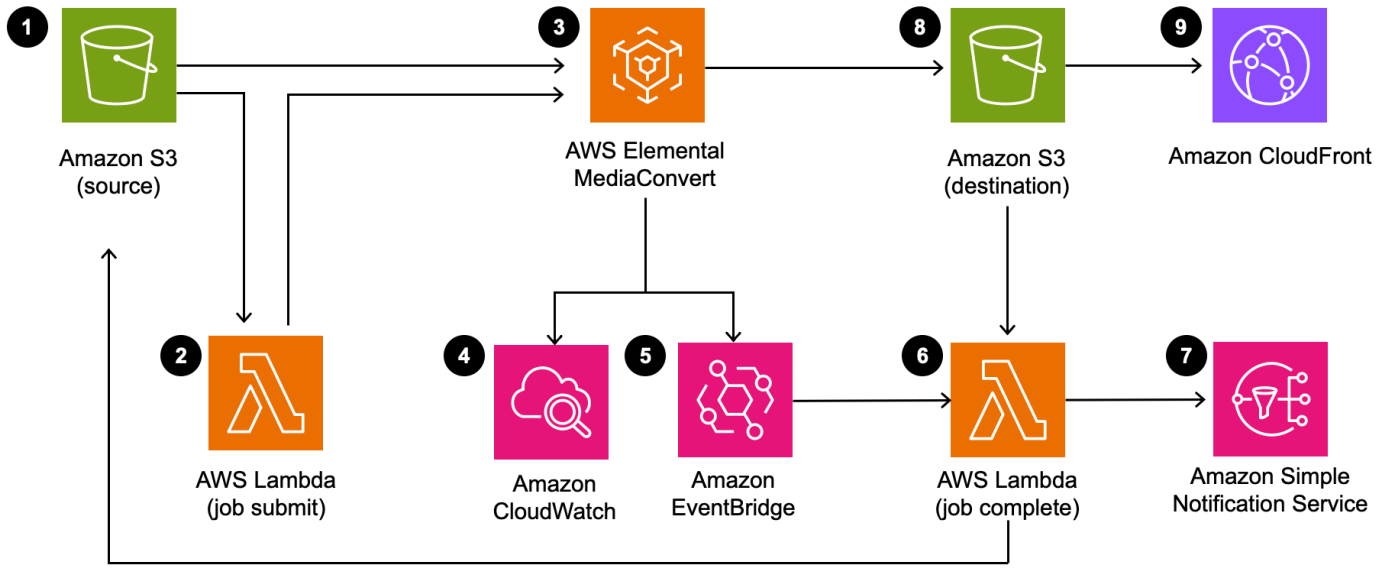
> **Note**
>
> For a general reference of AWS terms, see the AWS Glossary.

Architecture overview

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

Architecture diagram

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



Video on Demand on AWS Foundation architecture

The AWS CloudFormation template deploys the following infrastructure:

1. An Amazon S3 bucket stores source video files. The solution uploads a sample job settings file to the source S3 bucket. This file defines the encoding settings for AWS Elemental MediaConvert.
2. An AWS Lambda job submit function creates the encoding jobs in MediaConvert.
3. MediaConvert transcodes the video into HLS adaptive bitrate (ABR) files.
4. Amazon CloudWatch logs metrics about encoding jobs in MediaConvert.
5. Amazon EventBridge invokes the Lambda job complete function.
6. A Lambda job complete function processes the outputs.

7. An Amazon SNS topic sends notifications of completed jobs.
8. A destination S3 bucket stores the MediaConvert outputs.
9. Amazon CloudFront is configured with the destination S3 bucket as the origin for global distribution of the transcoded video content.

AWS Well-Architected design considerations

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

This section describes how the design principles and best practices of the Well-Architected Framework benefit this solution.

Operational excellence

This section describes how we architected this solution using the principles and best practices of the operational excellence pillar.

This solution pushes metrics to CloudWatch at various stages to provide observability into the infrastructure, Lambda functions, MediaConvert, S3 buckets, and the rest of the solution components.

Security

This section describes how we architected this solution using the principles and best practices of the security pillar.

This solution uses AWS Identity and Access Management (IAM) roles to allow customers to assign granular access policies and permissions to services and users in the AWS Cloud.

Reliability

This section describes how we architected this solution using the principles and best practices of the reliability pillar.

This solution uses AWS serverless services wherever possible (for example, Lambda and Amazon S3) to ensure high availability and quick recovery from service failure.

Performance efficiency

This section describes how we architected this solution using the principles and best practices of the performance efficiency pillar.

This solution uses serverless architecture. It can be launched in any AWS Region that supports the AWS services used in the solution, such as Lambda, Amazon S3, and MediaConvert.

This solution is automatically tested and reviewed by solutions architects and subject matter experts for areas to experiment and improve.

Cost optimization

This section describes how we architected this solution using the principles and best practices of the cost optimization pillar.

You can measure the efficiency of the workloads, and the costs associated with delivery, by using Application Manager.

Sustainability

This section describes how we architected this solution using the principles and best practices of the sustainability pillar.

This solution uses managed and serverless services to minimize the environmental impact of the backend services. If desired, you can run this solution only during specific events and then delete the stack after the program ends. This approach helps reduce the carbon footprint compared to the footprint of continually operating on-premises servers.

Architecture details

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

Ingest

To invoke the video processing workflow, you must upload the source video assets to the source S3 bucket through standard tools. For example, you can use the AWS Management Console, AWS Command Line Interface (AWS CLI), or third-party tools that interface with 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:

1. An Amazon S3 event notification invokes the `job-submit` Lambda function.
2. The `job-submit` Lambda function:
   a. Receives the details for the source video from the event.
   b. Applies the settings contained in the job settings file in the same top-level folder as the uploaded video in Amazon S3.
   c. Submits a job to MediaConvert using the processed job settings file.

To help you track the job in MediaConvert, the solution includes the following in the submitted job:

- Name of the workflow defined at deployment
- A globally unique identifier (GUID) created by the `job-submit` Lambda function

Encoding

The solution supports MediaConvert QVBR encoding mode, which ensures consistent, high-quality video transcoding with the smallest file size for any type of source video content.

By default, this solution includes a sample encoding job settings file that encodes your source videos into MP4, SD and HD formats of HLS, and SD and HD formats of DASH. You can overwrite the provided encoding job settings to use output formats that are supported by MediaConvert.

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

- HLS ABR with 5 renditions @ 1920x1080, 1280x720, 960x540, 640x360, 480x270

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*. To ensure validity of the job settings file, only export jobs that have successfully run and completed.

Processing

This solution includes an EventBridge rule 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 solution then adds details for the input file, job settings, and outputs to a `jobs-manifest.json` file stored at the root of the source S3 bucket. The following is an example `jobs-manifest.json` file.

```
{
  "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/..."
              }
        }
    }
    ...
```

```
}
```

The `job-complete` Lambda function also sends a summary of the job and the outputs to the Amazon SNS topic created at deployment. Errors from the encoding process are also captured by the Lambda function and sent to the SNS topic.

AWS services in this solution

AWS service	Description
AWS CloudFormation	**Core.** This solution uses AWS CloudFormation templates and stacks to automate its deployment.
Amazon CloudFront	**Core.** Accelerates delivery of the video content from the destination S3 bucket to end users.
AWS Elemental MediaConvert	**Core.** Transcodes media files from their source format into HLS ABR files.
AWS Lambda	**Core.** Provides job submit functions without provisioning or managing servers.
Amazon SNS	**Core.** Sends notifications of completed jobs.
Amazon S3	**Core.** Provides the storage for source video files and the outputs from MediaConvert.
Amazon CloudWatch	**Supporting.** Logs metrics about encoding jobs in MediaConvert.
Amazon EventBridge	**Supporting.** Invokes the Lambda job complete function.
AWS Systems Manager	**Supporting.** Provides application-level resource monitoring and visualization of resource operations and cost data.

Plan your deployment

This section describes the cost, security, Regions, and other considerations prior to deploying the 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. 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 this 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**.

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.

Cost table for a 60-minute source video

The following table provides a sample cost breakdown for processing a 60-minute source video using the solution deployed with the default parameters in the US East (N. Virginia) Region.

AWS service	Dimensions	Cost [USD]
Amazon CloudFront	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*	$229.50

AWS service	Dimensions	Cost [USD]
	2700 GB/hour × $0.085 per GB = $229.50 an hour	
Amazon S3	A 60-minute video uses a maximum of 9 GB of storage on Amazon S3, depending on the complexity of the video content. *$0.023 per GB × 9 GB = $0.207* **Note** Source videos uploaded to Amazon S3 add to this cost. After MediaConvert processing, delete source content from Amazon S3 to reduce storage costs.	$0.207
AWS Lambda	4 Lambda requests per file using $0.20 per million requests *$0.0000002 × 4 requests = $0.0000008*	$0.0000008
Amazon CloudWatch	AWS Free Tier. See Amazon CloudWatch Pricing for more information.	$0.00

AWS service	Dimensions	Cost [USD]
Amazon EventBridge	AWS Free Tier. See Amazon EventBridge Pricing for more information.	$0.00
Amazon SNS	AWS Free Tier. See Amazon SNS Pricing for more information.	$0.00
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**

MediaConvert cost

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 the Cost table for a 60-minute source video use the following Basic tier MediaConvert settings:

- AVC codec
- 1 pass quality
- 30 fps

This solution's CloudFormation template creates a destination S3 bucket where the processed videos are stored. The solution stores each processed result in a folder with the same name as the process ID. Pricing increases when using higher frame rates than 30 fps. For more information about MediaConvert pricing, refer to AWS Elemental MediaConvert Pricing.

Security

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

Amazon S3 bucket policy

The S3 buckets for MediaConvert output include a policy that allows access from CloudFront. Because the CloudFront endpoints are publicly accessible, the MediaConvert output bucket is also publicly accessible when accessed with CloudFront. For information on how to secure Amazon CloudFront, refer to Serving private content with signed URLs and signed cookies in the *Amazon CloudFront Developer Guide*.

IAM roles

IAM roles allow you to assign granular access policies and permissions to services and users on the AWS Cloud. This solution creates several IAM roles, including a role that grants MediaConvert access to Amazon S3. This role is necessary to allow the services to operate in your account.

Supported AWS Regions

This solution uses the MediaConvert service, which is not currently available in all AWS Regions. For the most current availability of AWS services by Region, see the AWS Regional Services List.

This solution is available in the following AWS Regions:

Region name	
US East (Ohio)	Asia Pacific (Sydney)
US East (N. Virginia)	Asia Pacific (Tokyo)
US West (N. California)	Canada (Central)
US West (Oregon)	Europe (Frankfurt)

Region name	
Africa (Cape Town)	Europe (Ireland)
Asia Pacific (Melbourne)	Europe (London)
Asia Pacific (Mumbai)	Europe (Paris)
Asia Pacific (Osaka)	Europe (Stockholm)
Asia Pacific (Seoul)	South America (São Paulo)
Asia Pacific (Singapore)	AWS GovCloud (US-West)

Customization

This solution uses a serverless architecture that you can update and extend for your specific video processing needs. For example, you can add or replace 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 mulitple job settings files.

Quotas

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

Quotas for AWS services in this solution

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

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

AWS CloudFormation quotas

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

AWS Elemental MediaConvert quotas

All MediaConvert jobs run in a queue. If you don't specify a queue when you create your job, MediaConvert sends it to the default on-demand queue. For information about how many queues you can create and how many jobs those queues can run, refer to How queues work in AWS Elemental MediaConvert in the *MediaConvert User Guide* and see Service quotas in the *AWS General Reference Guide*.

Deploy the solution

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

Before you launch the solution, review the cost, architecture, network security, and other considerations discussed earlier in this guide.

Time to deploy: Approximately 10 minutes

> **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 updated template and deploy the solution. For more information, see the Anonymized data collection section of this guide.

AWS CloudFormation template

You can download the CloudFormation template for this solution before deploying it.

View template

video-on-demand-on-aws-foundation.template - Use this template to launch the solution and all associated components. The default configuration deploys the core and supporting services found in the AWS services in this solution section, but you can customize the template to meet your specific needs.

Launch the stack

Follow the step-by-step instructions in this section to configure and deploy the solution into your account.

Time to deploy: Approximately 10 minutes

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

   Launch solution

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

   > **Note**
   >
   > This solution uses 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 **Create stack** page, verify that the correct template URL is in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack. For information about naming character limitations, see IAM and AWS STS quotas, name requirements, and character limits in the *AWS Identity and Access Management User Guide*.
5. Under **Parameters**, review the parameters for this solution template and modify them as necessary. This solution uses the following default values.

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

6. Select **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review and create** page, review and confirm the settings. Select the box acknowledging that the template creates IAM resources.
9. Choose **Submit** 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 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 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` function is inactive. However, do not delete the function, since it is necessary to manage associated resources.

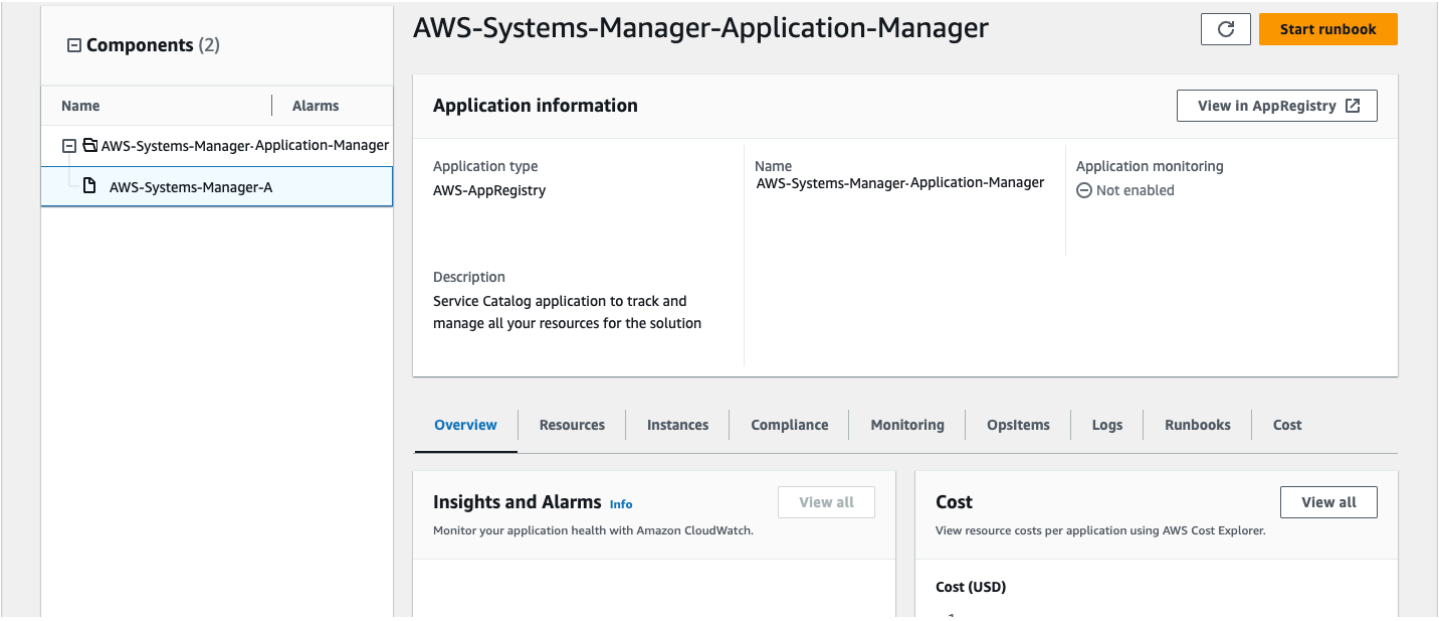
Monitor the solution with Service Catalog AppRegistry

The solution includes 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.

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. For example, deployment status, CloudWatch alarms, resource configurations, and operational issues.

The following figure depicts an example of the application view for the solution stack in Application Manager.



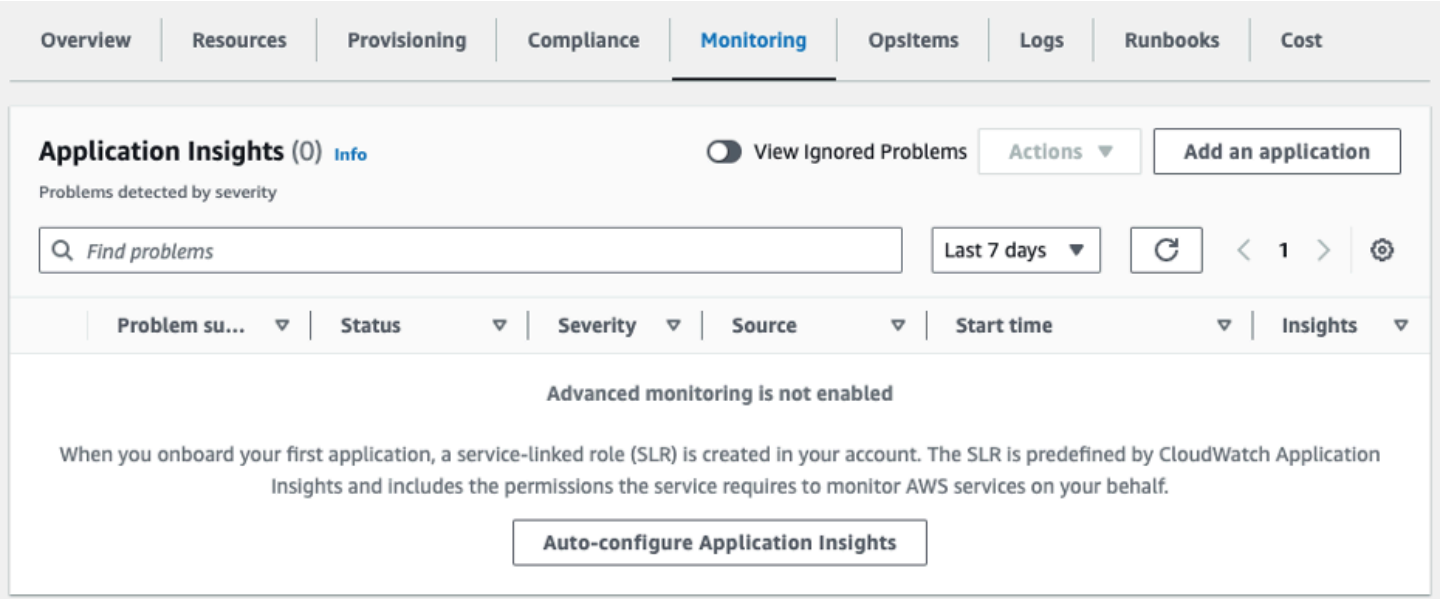
Solution stack in Application Manager

Activate CloudWatch Application Insights

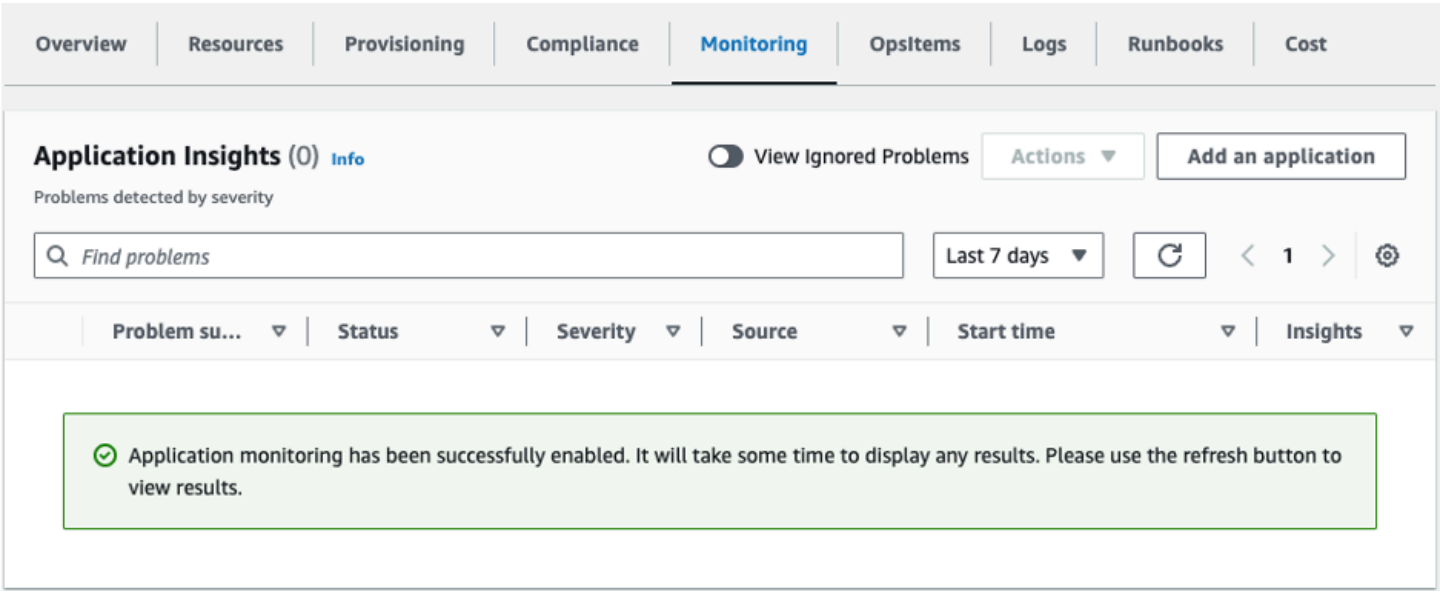
1. Sign in to the Systems Manager console.

2. In the navigation pane, choose **Application Manager**.
3. In **Applications**, search for the application name for this solution and select it.

   The application name will have **App Registry** in the **Application Source** column, and will have a combination of the solution name, Region, account ID, or stack name.
4. In the **Components** tree, choose the application stack you want to activate.
5. In the **Monitoring** tab, in **Application Insights**, select **Auto-configure Application Insights**.



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

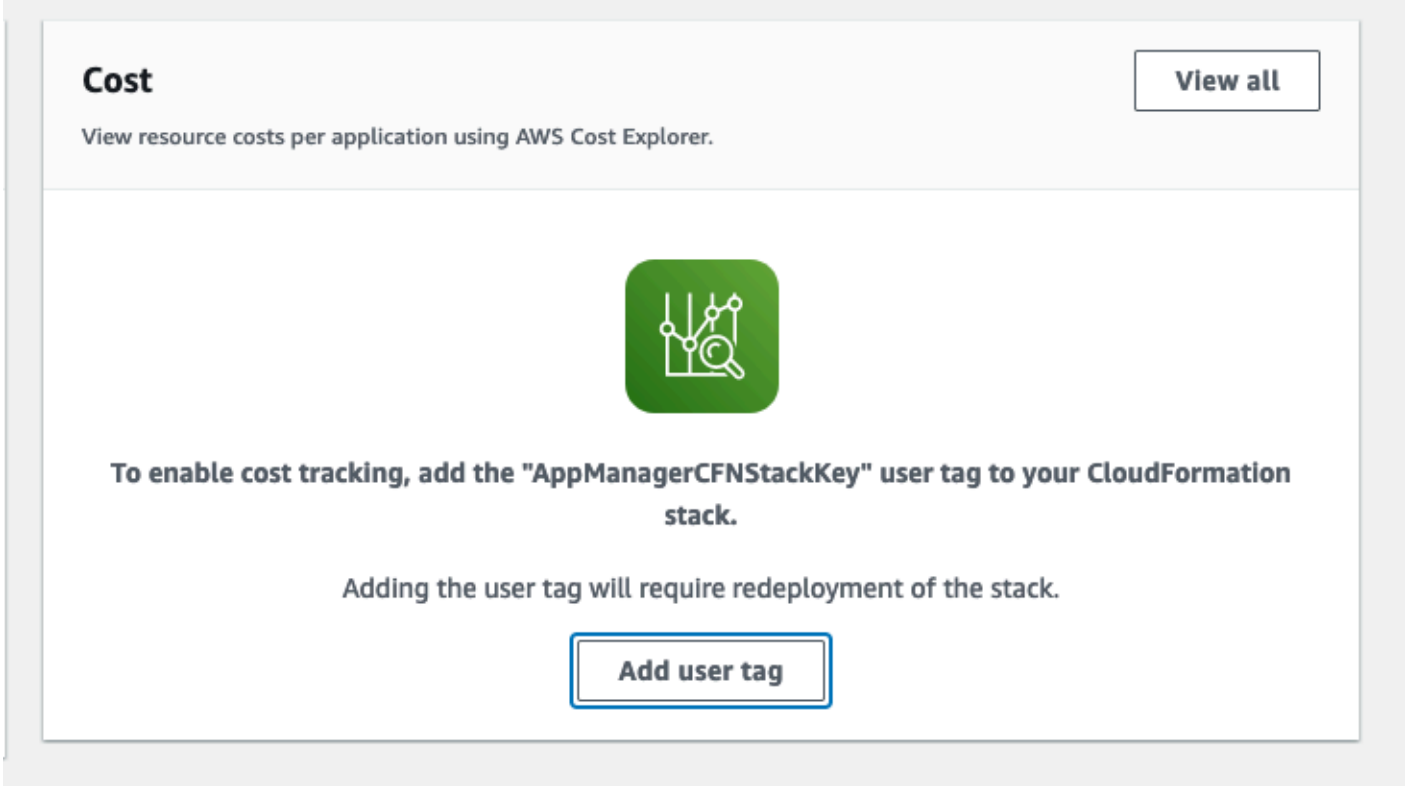


Confirm cost tags associated with the solution

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

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

   The application name will have **App Registry** in the **Application Source** column, and will have a combination of the solution name, Region, account ID, or stack name.
4. In the **Overview** tab, in **Cost**, select **Add user tag**.



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

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

Activate cost allocation tags associated with the solution

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

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

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** to view the solution's costs and usage over time.

Update the solution

If you have previously deployed the solution, follow this procedure to update the solution's CloudFormation stack to get the latest version of the solution's framework.

1. From the main account where the solution is deployed, sign in to the AWS CloudFormation console.
2. Select your existing Video on Demand on AWS Foundation CloudFormation stack, and select **Update**.
3. Select **Replace current template**.
4. Under **Specify template**:
   a. Select **Amazon S3 URL**.
   b. Copy the link of the `video-on-demand-on-aws-foundation.template` the section called "AWS CloudFormation 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**. Choose **Next** again.
5. Under **Parameters**, review the parameters for the template and modify them as necessary. For details about the parameters, see Launch the stack.
6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Select the box acknowledging that the template creates IAM resources.
9. Choose **View change set** and verify the changes.
10. 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 a UPDATE_COMPLETE status in approximately 10 minutes.

Troubleshooting

If these instructions don't address your issue, Contact AWS Support provides instructions for opening an AWS Support case for this solution.

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

If you're notified about a MediaConvert job failure, complete the following steps.

1. From the main account where the solution is deployed, sign in to the AWS Elemental MediaConvert console.
2. In the navigation pane, select **Jobs**.
3. Select the job ID of the job that failed.
4. On the **Job Summary** page, review the **Overview** section for an error message with more information on why the job failed. On this page, you can also find MediaConvert error codes 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's successfully completed. Incorrect encoding settings will result in the `job_submit` Lambda function to fail.

Contact AWS Support

If you have AWS Developer Support, AWS Business Support, or AWS Enterprise Support, you can use the Support Center to get expert assistance with this solution. The following sections provide instructions.

Create case

1. Sign in to Support Center.
2. Choose **Create case.**

How can we help?

1. Choose **Technical**
2. For **Service**, select **Solutions**.
3. For **Category**, select **Video on Demand on AWS Foundations**.
4. For **Severity**, select the option that best matches your use case.
5. When you enter the **Service**, **Category**, and **Severity**, the interface populates links to common troubleshooting questions. If you can't resolve your questions with these links, choose **Next step: Additional information**.

Additional information

1. For **Subject**, enter text summarizing your question or issue.
2. For **Description**, describe the issue in detail.
3. Choose **Attach files**.
4. Attach the information that AWS Support needs to process the request.

Help us resolve your case faster

1. Enter the requested information.
2. Choose **Next step: Solve now or contact us**.

Solve now or contact us

1. Review the **Solve now** solutions.
2. If you can’t resolve your issue with these solutions, choose **Contact us**, enter the requested information, and choose **Submit**.

Uninstall the solution

You can uninstall this solution from the AWS Management Console or by using the AWS CLI. You must manually delete the S3 buckets and CloudWatch logs created by this solution. AWS Solutions do not automatically delete these resources in case you have stored data to retain.

Using the AWS Management Console

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

Using AWS Command Line Interface

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

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

Deleting the Amazon S3 buckets

To prevent accidental data loss, this solution retains the solution-created Amazon S3 bucket (for deploying in an opt-in Region) if you decide to delete the CloudFormation stack. After uninstalling the solution, you can manually delete this S3 bucket if you do not need to retain the data. Follow these steps to delete the Amazon S3 bucket.

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

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

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

Alternatively, you can configure the CloudFormation template to delete the S3 buckets automatically. Before deleting the stack, change the deletion behavior in the AWS CloudFormation DeletionPolicy attribute.

Deleting the CloudWatch Logs

To prevent accidental data loss, this solution retains the CloudWatch logs if you decide to delete the CloudFormation stack. After uninstalling the solution, you can manually delete the logs if you don't 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.

Developer guide

This section provides the source code for the solution and additional customizations.

Source code

Visit our GitHub repository to download the source files for this solution and to share your customizations with others.

Job settings file

By default, this solution creates an `assets01` folder in the root of the S3 bucket with a `job-settings.json` file. Video uploads to this folder invoke a workflow to apply job settings to the job created in MediaConvert.

Changing the job settings file

You can customize the `job-settings.json` file or replace it with a new MediaConvert job template. To make a new job template or to customize the existing `job-setting.json` job template, refer to Working with AWS Elemental MediaConvert job templates in the *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 S3 source bucket.

Working with multiple job settings files

To support different job settings, create additional folders at the root of the source S3 bucket and include different job setting for each folder. The following shows an example folder structure. With this example 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:

```
assets01/
     job-settings.json
     video-01.mp4
assets02/
```

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

Important

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

Although you can export a completed job from MediaConvert to use as a job settings file, this solution doesn't 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 solution uses the output group name as part of the output destination path, and having distinct names makes it easier to locate where each output is being written.

Reference

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

Anonymized data collection

This solution includes an option to send anonymized operational metrics to AWS. We use this data to better understand how customers use this solution and related services and products. When 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 Video on Demand on AWS Foundation 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 through this survey. Data collection is subject to the AWS Privacy Notice. To opt out of this feature, complete the following steps before launching the AWS CloudFormation template.

1. Download the `video-on-demand-on-aws-foundation.template` the section called "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**, choose **Choose file** and select the edited template from your local drive.
8. Choose **Next** and follow the steps in Launch the stack in the Deploy the solution section of this guide.

Contributors

- Tom Nightingale
- Joan Morgan
- Eddie Goynes
- Raul Marquez

Revisions

Date	Change
November 2020	Initial release
July 2021	Release version 1.1.0: Cost reductions, job-settings.json file default settings change to 30 fps and support for additional file extension s, NPM CDK package updates, and Axios update. For more information, refer to the CHANGELOG.MD file in the GitHub repository.
October 2022	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. For more information, refer to the CHANGELOG.MD file in the GitHub repository.
December 2022	Minor changes to the AppRegistry section. For more information, refer to the CHANGELOG.MD file in the GitHub repository.
April 2023	Release version 1.2.1: Mitigated impact caused by new default settings for S3 Object Ownership (ACLs disabled) for all new S3 buckets.

Date	Change
	For more information, refer to the CHANGELOG.MD file in the GitHub repository.
May 2023	Release version 1.3.0: Added package_lock.json files to packages. Updated parameter names for consistency. For more information, refer to the CHANGELOG.MD file in the GitHub repository.
September 2023	Release version 1.3.1: Updated Lambda nodes to Node.js 18. Revised document for logical organization to improve readability and browsing experience. Added information about AWS Well-Architected design, features and benefits, use cases, and supported AWS Regions. Updated instructions to update and install the solution, providing additional clarity. For more information, refer to the CHANGELOG.MD file in the GitHub repository.
November 2023	Release version 1.3.2: Updated package versions to resolve security vulnerabilities. For more information, refer to the CHANGELOG.MD file in the GitHub repository.
November 2023	Documentation update: Added Confirm cost tags associated with the solution to the Monitoring the solution with AWS Service Catalog AppRegistry section.
February 2024	Documentation update: Added AWS Developer Support and merged Contact AWS Support into the Troubleshooting section.

Date	Change
July 2024	Release version 1.3.3: Upgraded vulnerable packages. For more information, refer to the CHANGELOG.MD file in the GitHub repository.
August 2024	Release version 1.3.4: Updated package versions to resolve security vulnerabilities. For more information, refer to the CHANGELOG.MD file in the GitHub repository.
August 2024	Release version 1.3.5: Updated package versions to resolve security vulnerabilities. For more information, refer to the CHANGELOG.MD file in the GitHub repository.
September 2024	Release version 1.3.6: Updated package versions to resolve security vulnerabilities. For more information, refer to the CHANGELOG.MD file in the GitHub repository.
November 2024	Release version 1.3.7: Updated package versions to resolve security vulnerabilities. For more information, refer to the CHANGELOG.MD file in the GitHub repository.

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents AWS current product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers, or licensors. AWS products or services are provided "as is" without warranties, representations, or conditions of any kind, whether express or implied. AWS 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 Foundation is licensed under the terms of the Apache License Version 2.0.