Implementation Guide

Media2Cloud on AWS

Media2Cloud on AWS: Implementation Guide

Table of Contents

Extract key details from your media files in your AWS accounts

Publication date: *January 2019 (last update: November 2023*)

Migrating your digital asset management to the cloud allows you to take advantage of the latest innovations in asset management and supply chain applications. However, transferring your existing video archives to the cloud can be a challenging and slow process.

The Media2Cloud on AWS solution helps streamline and automate the content ingestion process. It sets up serverless end-to-end ingestion and analysis workflows to move your video assets and associated metadata to the Amazon Web Services (AWS) Cloud. During the migration, this solution analyzes and extracts machine learning metadata from your video and images using Amazon Rekognition, Amazon Transcribe, and Amazon Comprehend. It extracts tabular information from scanned documents using Amazon Textract. This solution also includes a web interface to help you to immediately start ingesting and analyzing your content.

Media2Cloud on AWS is designed to provide a serverless framework for accelerating the setup and configuration of a content ingestion and analysis process. We recommend that you use this solution as a baseline and customize it to meet your specific needs.

This implementation guide provides an overview of the Media2Cloud solution, its reference architecture and components, considerations for planning the deployment, configuration steps for deploying the solution to the AWS Cloud.

The guide is intended for IT infrastructure architects and developers who have practical experience working with video workflows and architecting in the AWS Cloud.

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 on 100 hours of videos totaling one terabyte with the default settings in the US East (N. Virginia) Region is **$2,149.95** (one time	Cost

If you want to . . .	Read . . .
processing) with **$104.60**/month (recurrin g) for Amazon S3 data storage and Amazon OpenSearch Service search engine.	
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:

Save development time

Leverage the Media2Cloud on AWS solution out-of-the-box or as a reference implementation for building a serverless framework to accelerate setting up and configuring the content ingestion process.

Simple user interface

The solution creates a web interface that makes it easy to upload, browse, and search your video and image files and extracted metadata.

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

This solution includes a Service Catalog AppRegistry resource to register the solution's CloudFormation template and its underlying resources as an application in both Service Catalog 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

Archive migration

Legacy content archives are typically stored on linear tape-open (LTO), digital video, audio tapes, or on-premises hard drives, which are cold storage options with long retrieval times. By digitizing your files, Media2Cloud on AWS accelerates the migration of on-premises archives to the cloud. In addition, moving your archives to the cloud and indexing the content with artificial intelligence (AI) and machine learning (ML) enables you to better manage and monetize legacy content.

Media intelligence

Media and entertainment companies can reduce the cost of media storage while improving the search functionality and monetization of media asset libraries by migrating them to the cloud. You can use Media2Cloud on AWS to set up workflows to migrate media assets to AWS, offering options to analyze video, images, audio, and documents to generate valuable metadata upon migration. This can help you to streamline the media migration process and improve the search, discovery, and monetizing content.

Educational content delivery

Professional development and educational initiatives create incentives for nonprofit members, and can be important revenue generators for nonprofit organizations. Media2Cloud on AWS 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 by automating content digitization and analysis.

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.

workflow

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

proxies

Files created at a lower resolution using compression technology for video, audio, images, and documents.

LTO

Linear Tape-Open is a magnetic tape data storage technology used for backup and archiving purposes.

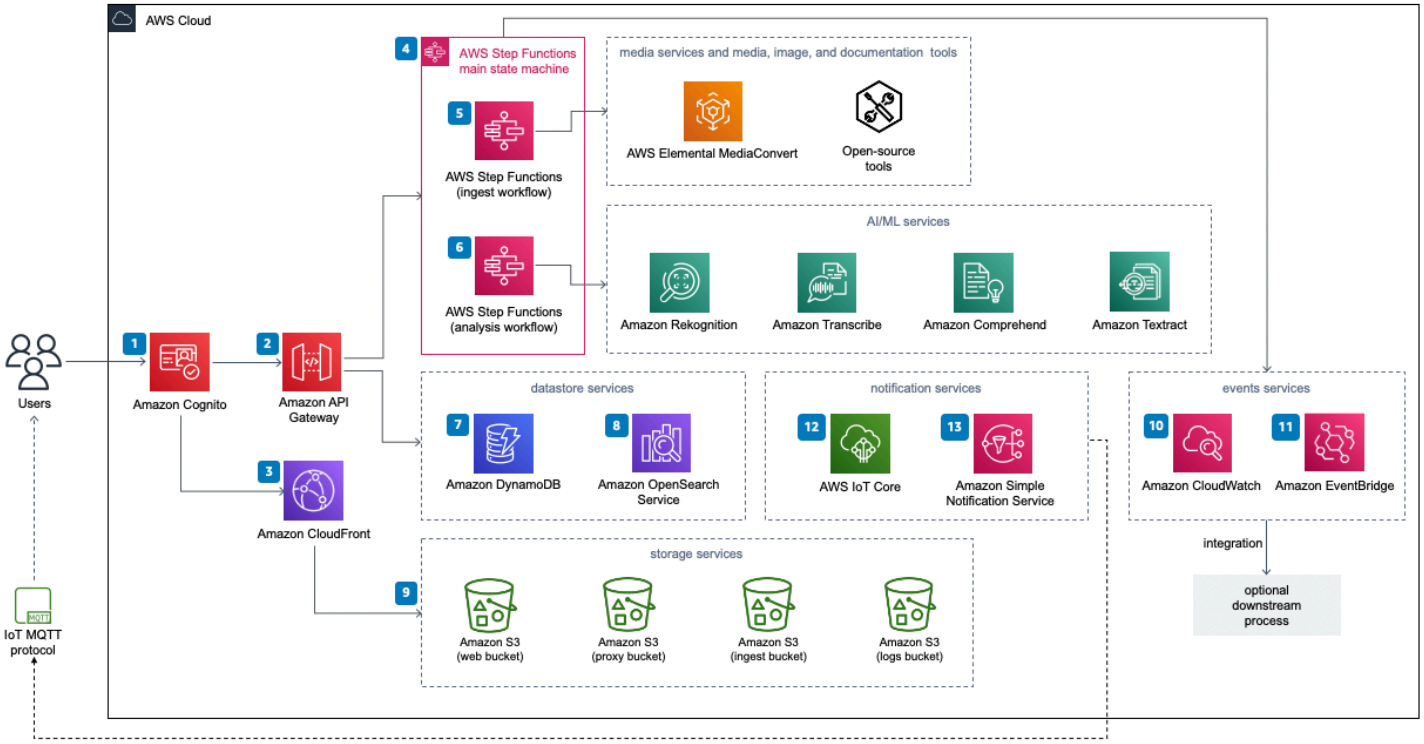
MAM

Media Asset Management is a software tool that centrally manages, organizes, and catalogs media assets. MAM provides the option to associate media assets to one or several files with customized metadata to ease the content searching, retrieving, and distribution tasks.

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

Architecture overview

Deploying the Media2Cloud on AWS solution builds the following environment in the AWS Cloud.



Media2Cloud on AWS architecture on AWS

The AWS CloudFormation template deploys the following infrastructure:

1. An Amazon Cognito user pool to provide a user directory.
2. An Amazon API Gateway RESTful API endpoint, which is configured to use AWS IAM authentication.
3. An Amazon CloudFront distribution that hosts the web application artifacts such as minimized JavaScript files and graphics stored in the web bucket.
4. An AWS Step Functions main state machine which serves as the entry point to the solution's backend ingestion and analysis workflows.
5. An AWS Step Functions ingestion sub-state machine that orchestrates the ingestion process by media file type and generates proxies for ingested media. It uses AWS Elemental MediaConvert for video and audio files and open-source tools for image files and documents.

6. A Step Functions analysis sub-state machine that is responsible for the analysis process. It consists of AWS Step Functions that run analysis jobs with Amazon Rekognition, Amazon Transcribe, Amazon Comprehend, and Amazon Textract.
7. Amazon DynamoDB tables to store artifacts generated during the ingestion and analysis processes, such as overall status, pointers to where intermediate files are stored, and state machine run tokens.
8. An Amazon OpenSearch Service cluster, which stores ingestion attributes and machine learning metadata, and facilitates customers' search and discovery needs.
9. Four Amazon Simple Storage Service (Amazon S3) buckets to store uploaded content, file proxies that the solution generates during ingestion, static web application artifacts, and access logs for services used.
10. Amazon CloudWatch event rules that are logged when specific tasks undergo state changes.
11. Amazon EventBridge used by an internal queue management system where the backlog system notifies workflows (state machines) when a queued AI/ML request has been processed.
12. An AWS IoT Core topic that allows the ingestion and analysis workflows to communicate with the front-end web application asynchronously through publish/subscribe MQTT messaging.
13. Amazon Simple Notification Service (Amazon SNS) topics to allow Amazon Rekognition to publish job status in the video analysis workflow, and to support custom integration with customers' system, by allowing the solution to publish `ingest_completed` and `analysis_completed` events.

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.

The Media2Cloud on AWS solution pushes metrics to Amazon CloudWatch at various stages to provide observability into the infrastructure, such as Lambda functions, AI services, S3 buckets, and the rest of the AWS services in this solution .

Security

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

We highly recommends that customers encrypt sensitive data in transit and at rest. This solution automatically encrypts media files and metadata at rest with Amazon S3 server-side encryption (SSE). The solution's Amazon SNS topics and DynamoDB tables are also encrypted at rest using SSE.

Documents indexed to the OpenSearch Service cluster are encrypted at rest. Node-to-node communication within the cluster is also encrypted.

Media2Cloud on AWS deploys a static website hosted in an S3 bucket. To help reduce latency and improve security, this solution includes a CloudFront distribution with an origin access identity, which is a special CloudFront user that helps restrict access to the solution's website bucket contents. For more information, refer to Restricting access to an Amazon S3 origin.

Reliability

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

Media2Cloud on AWS uses AWS serverless services wherever possible (for example, Lambda, API Gateway, S3, and DynamoDB) 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.

Media2Cloud on AWS uses serverless architecture throughout the solution. You can launch in any AWS Region that supports the AWS services used in the solution.

This solution is automatically tested and reviewed by solution 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.

Media2Cloud on AWS uses a serverless and event-driven architecture; therefore, customers are only charged for what they use. The solution's design allows users to configure and tailor their own media workflows, and use only the AWS services that they need.

Sustainability

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

Media2Cloud on AWS uses managed and serverless services to minimize the environmental impact of the backend services. A critical component for sustainability provided by the solution is maximizing the usage of the AWS AI services. The serverless design of Media2Cloud on AWS (using Lambda, API Gateway, Amazon S3, and DynamoDB) aims to reduce the carbon footprint compared to the footprint of continually operating operating on-premises servers.

Architecture details

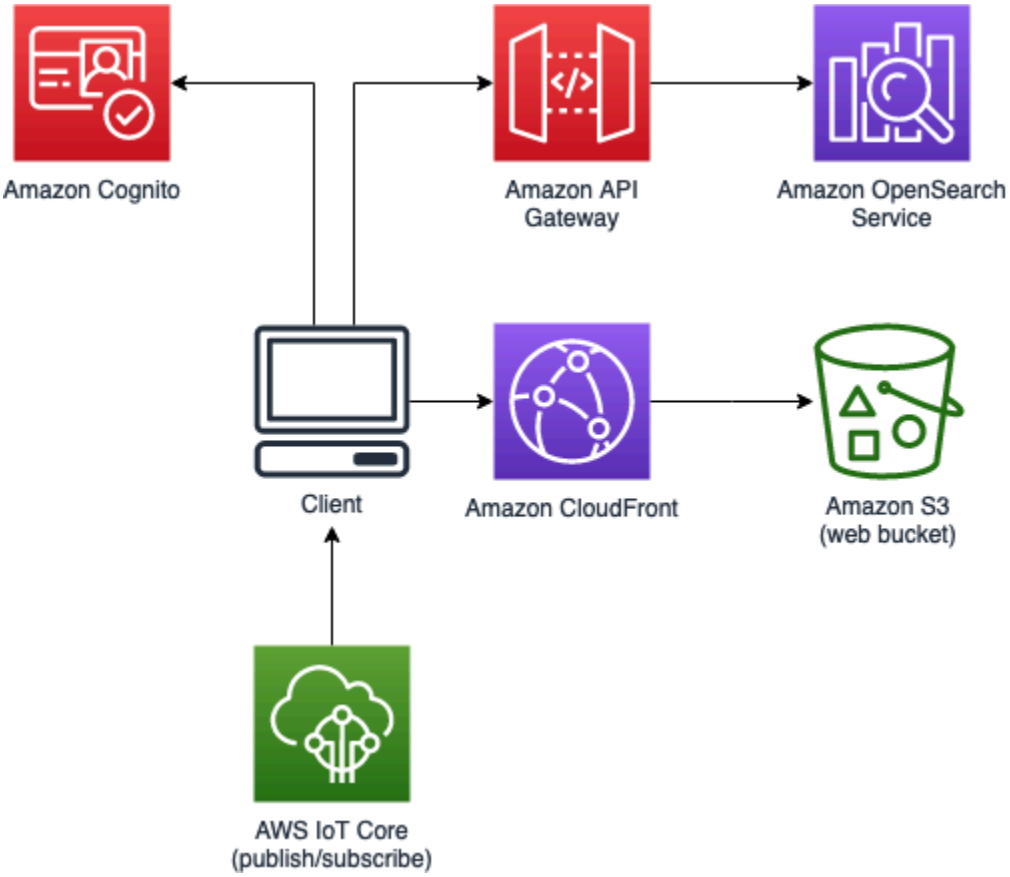
The AWS CloudFormation template deploys three logical components: a front-end web application, orchestration workflows (ingestion and analysis), and data storage. The web application provides an interface for customers to upload media content, and view and manage their archive collection. The ingestion and analysis workflows are initiated when a customer uploads content to the application. The ingestion workflow orchestrates tasks to ingest source videos, images, audio, and documents in a serverless manner. The analysis workflow analyzes and extracts machine learning metadata from content. When you upload a media asset to the Amazon S3 ingestion bucket, the ingestion workflow creates a standardized proxy file and thumbnails for analysis. The analysis workflow analyzes the content and extracts metadata using AWS AI services.

The Amazon S3 ingestion bucket has an Amazon S3 lifecycle policy that allows the solution to move uploaded videos and images to Amazon Simple Storage Service Glacier (Amazon Glacier) for archiving. Additionally, the AWS CloudFormation template deploys multiple Amazon DynamoDB tables to store metadata about each processed content, such as *pointers* to where its proxy files are stored in Amazon S3 and the types of AI/ML analysis performed on it. The solution also deploys an Amazon OpenSearch Service cluster that allows customers to search and discover technical media metadata or AI/ML generated metadata.

Web interface

This solution deploys a web interface that allows you to upload, browse, search video and image files, extract metadata, and search and discover content metadata. The web interface leverages Amazon Cognito for user authentication and is powered by web assets hosted in the web Amazon S3 bucket. Amazon CloudFront provides public access to the solution's website bucket contents. An Amazon API Gateway RESTful API is used for searching results stored in the Amazon OpenSearch Service cluster, and AWS IoT Core is used as a publish/subscribe message broker to periodically update workflow progress to connected web clients.

The web interface also provides a human-in-the-loop feature that allows customers to remediate cases where Amazon Rekognition cannot detect individuals in a video or cannot provide logical groupings of related individuals based on the customer's specific use case, for example, Olympic athletes for a specific year. Using the web interface, customers can create face collections and index people in the video to that face collection.



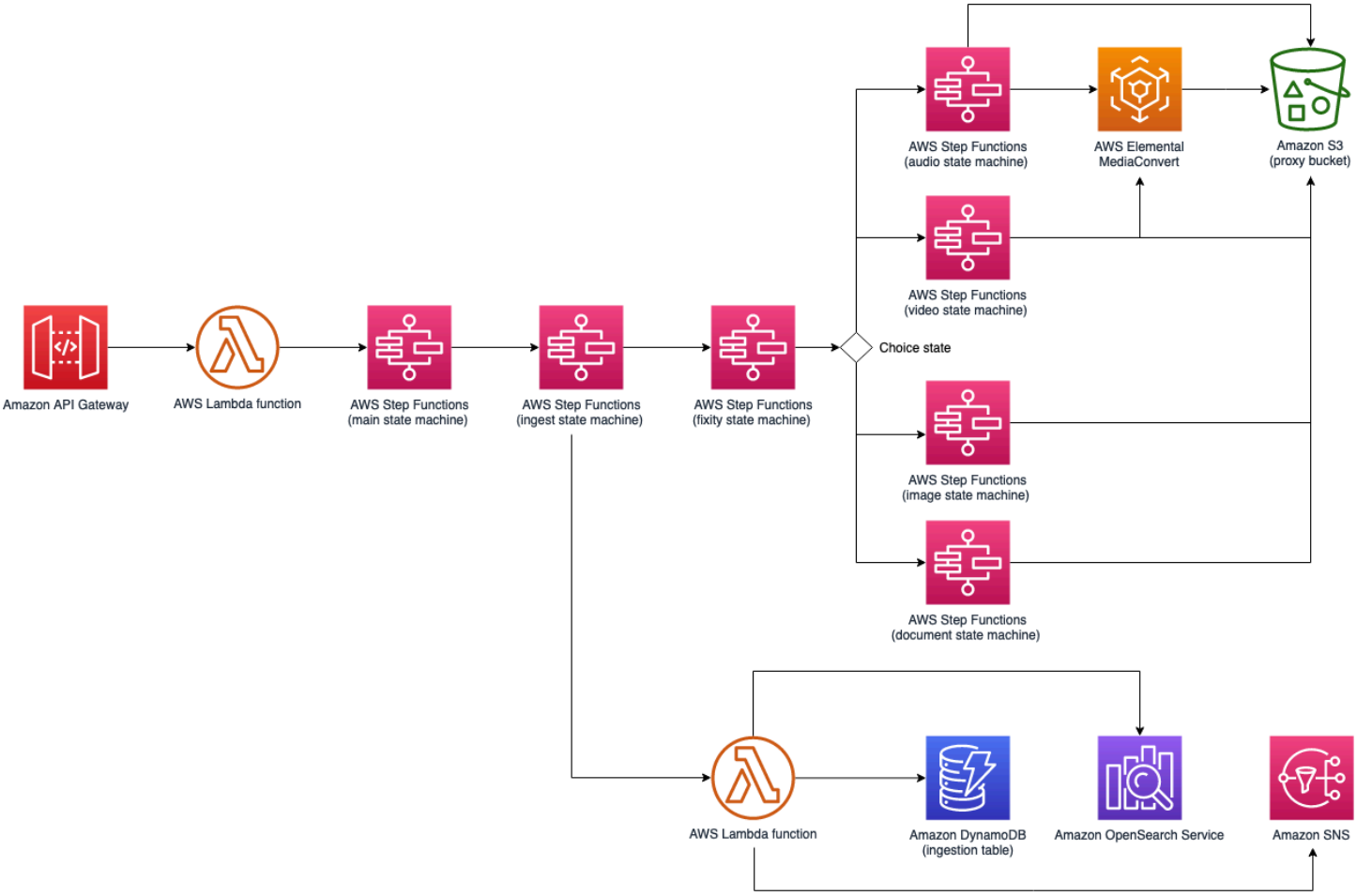
Media2Cloud on AWS web interface

Ingestion workflow

The ingestion workflow includes AWS Step Functions and AWS Lambda, which orchestrate the specific ingestion workflow for a video, image, audio file, or document. When a customer uploads a new media file to the Amazon S3 ingestion bucket through the Media2Cloud on AWS web interface, the ingestion process starts. The workflow generates an asset unique identifier, computes and validates an MD5 checksum, and extracts media information such as bitrate, formats, audio channels container format for video, or EXIF information such as GPS location, model, and make for image.

For video and audio files, the ingestion workflow initiates AWS Elemental MediaConvert to create standardized proxy files and thumbnails of the media for analysis. For image files, the ingestion workflow uses an open-source tool, EXIFTool to extract technical metadata and to create proxy images. Similarly for documents, the ingestion workflow generates image proxies for each page in a document. If the media content is in Amazon Glacier or S3 Glacier Deep Archive storage, the

workflow temporarily restores the media content from archive storage to Amazon S3 storage. Proxy files are created and stored in a Amazon S3 proxy bucket, while the technical metadata extracted from media content are indexed in an Amazon OpenSearch Service cluster. When video ingestion process completes, Amazon SNS sends notifications to subscribed users who might use the notification to start other workflows. For example, third party partner solutions, such as Media Asset Manager (MAM) and Archive System, can subscribe to the Amazon SNS topic and then integrate the derived information into their workflows. When an Amazon SNS ingestion notification is received, the automated system can import the files into its system. For more information, refer to Amazon SNS notifications.



Media2Cloud on AWS ingestion workflow

Analysis workflow

The analysis workflow includes AWS Step Functions and AWS Lambda which leverage Amazon Rekognition, Amazon Transcribe, Amazon Comprehend, and Amazon Textract to analyze and extract machine learning metadata from the proxy files generated in the ingestion workflow. The

Media2Cloud on AWS solution provides the following preset options for the analysis process when you deploy the template: **Default**, **All**, and **Audio and Text**.

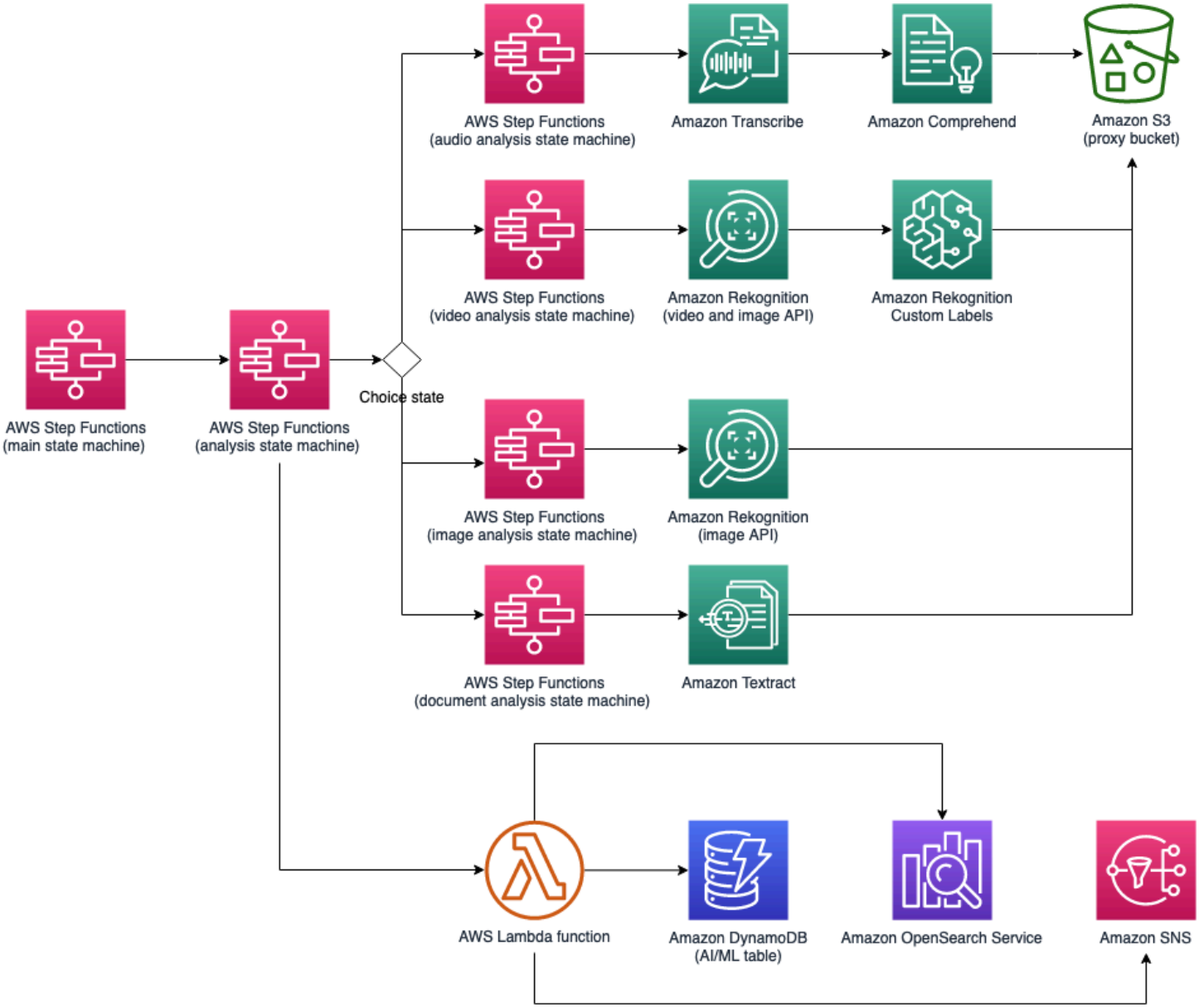
- **Default** - Activates *celebrity recognition*, *labels*, *transcription*, *key phrases*, *entities*, and *text* processes.
- **All** - Activates all detections including *celebrity recognition*, *labels*, *transcription*, *key phrases*, *entities*, *text*, *faces*, *face matches*, *person*, *moderation*, *sentiment*, and *topic* processes.
- **Audio and Text** - Activates *transcription*, *key phrases*, *entities*, and *text* processes.

The web interface also allows the end user to refine the AI/ML settings during the upload process.

The analysis workflow includes four sub-state machines to process the analysis.

- The video analysis state machine analyzes and extracts AI/ML metadata from the video proxy using Amazon Rekognition video APIs.
- The audio analysis state machine analyzes and extracts AI/ML metadata from the audio stream of the proxy file using Amazon Transcribe and Amazon Comprehend.
- The image analysis state machine analyzes and extracts image metadata with Amazon Rekognition image APIs.
- The document analysis state machine extracts text, images, and data using Amazon Textract.

To start the analysis workflow, a Lambda function first checks an incoming analysis request and prepares the optimal AI/ML analysis option to run, based on the type of media in the request, and the availability of specific detections. For video and audio, it transforms the metadata results into WebVTT subtitle tracks, chapter markers, key phrases, labels, sentiments, entities, and locations. The analysis workflow can also provide customized analysis output if the customer uses Amazon Rekognition custom label models, Amazon Transcribe custom vocabularies or Amazon Comprehend custom entity recognition. The machine learning metadata results are stored in an Amazon S3 proxy bucket and indexed in an Amazon OpenSearch Service cluster. When the analysis is completed, Amazon SNS sends notifications to subscribed users. For more information, refer to Amazon SNS notifications.



Media2Cloud on AWS analysis workflow

Error handling

The Media2Cloud on AWS solution applies a catch and retry concept for error handling to the state machines to improve the resiliency of the solution by retrying the state run multiple times. When the state run exhausts the retries, it stops the run and generates an error.

The solution also uses Amazon CloudWatch Events to respond to run errors caused by the state machines (ingestion and analysis). The Lambda error handling function processes the error by

analyzing the run history of the failed state machine and sends an Amazon SNS notification to subscribers.

Proxy files

When a new video is uploaded to Amazon S3, the Media2Cloud on AWS solution automatically converts the video to MP4 format and creates a compressed version of the video known as a proxy file. For this solution, proxy files are used to allow users to upload videos of various sizing and formatting, without being subject to Amazon Rekognition and Amazon Transcribe quotas. Additionally, the proxy files can be used as reference proxies in a Media Asset Manager (MAM) for search, discovery, and proxy editing. The solution also generates compressed proxies for audio, images, and documents after extracting technical data from these file types during the ingestion process.

Amazon DynamoDB

The solution deploys the following Amazon DynamoDB tables which are configured to use on-demand capacity and encryption at rest using Server-Side Encryption (SSE).

- A table to store ingestion information
- A table to store machine learning metadata
- A table to temporarily store state machine run tokens that are used internally to communicate back to specific state machine runs
- A table to temporarily store service backlog requests, an internal queue management system to support large number of AI requests

Amazon OpenSearch Service

The solution configures an Amazon OpenSearch Service cluster to index ingestion technical metadata and analysis metadata. The solution creates indices per type of the machine learning categories such as `celeb`, `label`, `face`, `faceMatch`, `segment`, `moderation`, `person`, `textract`, `transcribe`, `keyphrase`, `entity`, and `ingest` to allow the end user to fine tune the search results. The indexed documents are encrypted at rest. Node-to-node encryption is also activated.

Amazon SNS

This solution deploys two Amazon SNS topics. One to receive ingestion, analysis, and error notifications from the workflows and the other topic is used internally by Amazon Rekognition to send job notifications to the state machine run.

Integration with Service Catalog AppRegistry and AWS Systems Manager Application Manager

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

AWS services in this solution

AWS service	Description
Amazon API Gateway	**Core.** Provides the RESTful API endpoint, which is configured to use IAM authentication.
Amazon CloudFront	**Core.** Hosts the web application artifacts such as minimized JavaScript files and graphics stored in the web bucket.
Amazon Cognito	**Core.** Provides user directory and access management.
Amazon DynamoDB	**Core.** Provides the storage for artifacts generated during the ingestion and analysis processes, such as overall status, pointers to where intermediate files are stored, and state machine run tokens.
AWS Lambda	**Core.** Supports orchestration of ingestion and analysis workflows.

AWS service	Description
Amazon OpenSearch Service	**Core.** Stores ingestion attributes and machine learning metadata, and facilitates customers' search and discovery needs.
Amazon S3	**Core.** Provides storage for the uploaded content, file proxies that the solution generates during ingestion, static web application artifacts, and access logs for services used.
AWS Step Functions	**Core.** Provides main state machine which serves as the entry point to the solution's backend ingestion and analysis workflows.
AWS Elemental MediaConvert	**Supporting.** Can be integrated into workflows to transcode input video into mpeg4 format and generate proxies for ingested media.
Amazon EventBridge	**Supporting.** Used by an internal queue management system where the backlog system notifies workflows (state machines) when a queued AI/ML request has been processed.
AWS IoT Core	**Supporting.** Allows the ingestion and analysis workflows to communicate with the front-end web application asynchronously through publish/subscribe MQTT messaging.
Amazon SNS	**Supporting.** Allows Amazon Rekognition to publish job status in the video analysis workflow. In addition, Amazon SNS allows Amazon Rekognition to support custom integrations with customers' systems by allowing the solution to publish ingest_completed and analysis_completed events.

AWS service	Description
AWS Systems Manager	**Supporting.** Provides application-level resource monitoring and visualization of resource operations and cost data.
Amazon Comprehend	**Optional.** Can be integrated into workflows to find key phrases in text and references to real-world objects, dates, and quantities in text.
Amazon Rekognition	**Optional.** Can be integrated into workflows for celebrity recognition, content moderatio n, face detection, face search, label detection , person tracking, shot, text, and technical cue detection.
Amazon Textract	**Optional.** Can be integrated into workflows to extract tabular metdata from documents using Optical Character Recognition (OCR).
Amazon Transcribe	**Optional.** Can be integrated into workflows to create SRT or VTT captions files from video transcripts. It can also convert input audio to text.

How Media2Cloud works

This sections describes the image, video, and audio ingestion along with key features for analyses of data in this solution.

File paths in Amazon S3

When a customer deploys Media2Cloud on AWS, the solution creates four different Amazon Simple Storage Service (Amazon S3) buckets to store assets:

- A web bucket that stores the static HTML, CSS, and JavaScript files for the web interface.
- An ingestion bucket that stores your original source files.
- A proxy bucket that stores all the files and assets generated by the solution, including:
  - Video proxies and thumbnail images generated by AWS Elemental MediaConvert
  - MediaInfo XML output generated by MediaInfo
  - JSON documents generated by EXIFTool
  - Machine learning metadata generated by AWS AI services
  - Additional WebVTT tracks and analysis JSON documents created by the solution
- A logs bucket that stores all access logs for the web bucket, the ingestion bucket, the proxy bucket, and Amazon CloudFront standard logs.

Table 1: File types and Amazon S3 file paths

File type	File path
Web static assets	S3://*<web-bucket>* /
Uploaded file	S3://*<ingest-bucket>* /*<file-basename>* /*<filename>*
Technical metadata such as mediainfo and EXIF results	S3://*<proxy-bucket>* /*<uuid>*/*<filename>* /mediainfo/

File type	File path
	S3://<proxy-bucket> /<uuid>/<filename> /imageinfo/
Proxy files, thumbnails generated by MediaConvert	S3://<proxy-bucket> /<uuid>/<filename> /transcode/proxy S3://<proxy-bucket> /<uuid>/<filename> /transcode/aiml S3://<proxy-bucket> /<uuid>/<filename> /transcode/frameCapture
All AI/ML analysis results	S3://<proxy-bucket> /<uuid>/<filename> /raw/
Raw AI/ML analysis results from Amazon AI services	S3://<proxy-bucket> /<uuid>/<filename> /raw/<date-time> /comprehend/ S3://<proxy-bucket> /<uuid>/<filename> /raw/<date-time> /rekognition/ S3://<proxy-bucket> /<uuid>/<filename> /raw/<date-time> /transcribe/ S3://<proxy-bucket> /<uuid>/<filename> /raw/<date-time> /textract/

File type	File path
WebVTT tracks generated by analysis state machine	S3://*<proxy-bucket>* /*<uuid>*/*<filename>* /vtt/celeb/*<name>*.vtt S3://*<proxy-bucket>* /*<uuid>*/*<filename>* /vtt/face/*<name>*.vtt S3://*<proxy-bucket>* /*<uuid>*/*<filename>* /vtt/moderation/*<name>*.vtt S3://*<proxy-bucket>* /*<uuid>*/*<filename>* /vtt/person/*<name>*.vtt S3://*<proxy-bucket>* /*<uuid>*/*<filename>* /vtt/label/*<name>*.vtt S3://*<proxy-bucket>* /*<uuid>*/*<filename>* /vtt/segment/*<name>*.vtt S3://*<proxy-bucket>* /*<uuid>*/*<filename>* /vtt/transcribe/*<name>*.vtt

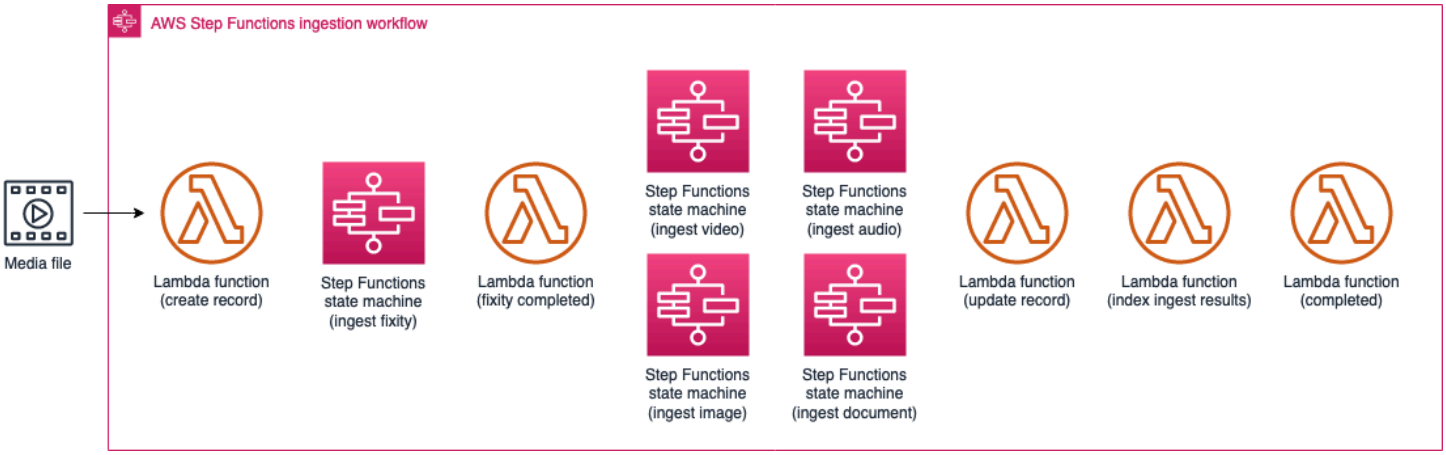
File type	File path
Metadata JSON documents generated by analysis state machine	S3://*<proxy-bucket>* /*<uuid>*/*<filename>* /analysis/vtt/celeb/*<name>*.vtt S3://*<proxy-bucket>* /*<uuid>*/*<filename>* /analysis/vtt/entity/*<name>*.vtt S3://*<proxy-bucket>* /*<uuid>*/*<filename>* /analysis/vtt/keyphrase/*<name>*.vtt
Access logs for the web bucket, ingest bucket, proxy bucket, and CloudFront OIA	S3://*<logs-bucket>* */access_logs_cloudfront/uuid.*<date>*.uuid.gz* S3://*<logs-bucket>* */access_logs_ingest_bucket/*<date-time> *-uuid* S3://*<logs-bucket>* */access_proxy_bucket/*<date-time> *-uuid* S3://*<logs-bucket>* */access_logs_web_bucket/*<date-time> *-uuid*

Ingestion state machine

The Media2Cloud on AWS solution ingests videos, images, audio, and documents to extract media information and generate proxies using AWS Step Functions nested state machines and AWS Lambda functions. When a new media file is uploaded through the web interface, the solution sends an HTTPS request to the Amazon API Gateway RESTful API endpoint to start the ingestion process. A Lambda function invokes the main state machine that starts the nested ingestion state machine. Media2Cloud on AWS sends state machine progress and status to an AWS IoT topic that activates the web interface to display updated results to the user.

The ingestion state machine is composed of the following processes involving AWS Step Functions nested state machines and Lambda functions.

- **Create record** - Creates a record of the uploaded file to the Amazon DynamoDB ingestion table. Information includes the S3 object key of the file, universally unique identifier (UUID), and MD5 checksum value.
- **Start fixity (nested)** – A nested state machine that perform the fixity check to restore the uploaded file based on its storage class.
- **Choose by media type** - Delegates the proxy generation workflow based on the media type of the file. The media type can be image, video, audio, or document.
- **Start image ingestion (nested)** – A sub-state machine to process an image file.
- **Start video ingestion (nested)** - A sub-state machine to process a video file.
- **Start audio ingestion (nested)** - A sub-state machine to process an audio file.
- **Start document ingestion (nested)** - A sub-state machine to process a document.
- **Update record** - Collects all results from the states such as locations of the proxies, thumbnail images, and either MediaInfo (for videos) or embedded technical metadata within videos or images and updates the results to the ingest DynamoDB table.
- **Index ingestion results** - Indexes the technical metadata to OpenSearch Service cluster.

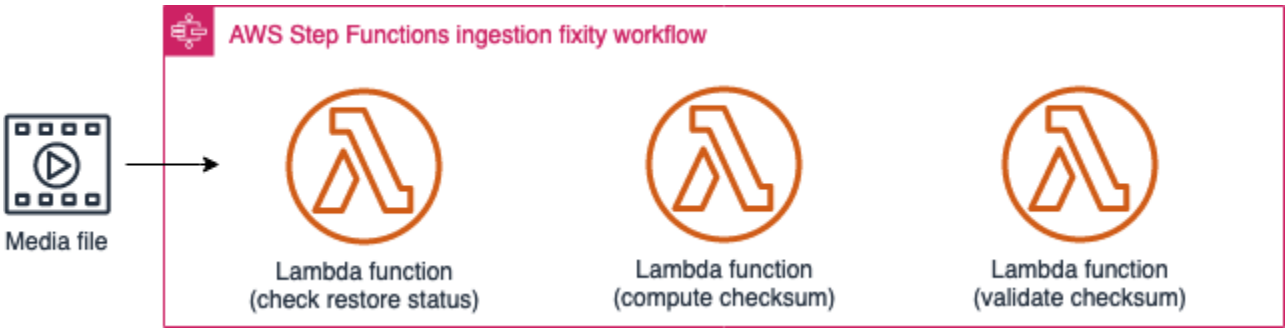


Ingestion state machine workflow

Ingestion fixity sub-state machine

The ingestion fixity sub-state machine uses AWS Lambda functions to compute and validate the file checksum. If the uploaded file is stored in either of the Amazon Glacier Flexible Retrieval S3 Glacier Deep Archive storage classes, the state machine temporarily restores the object before the checksum is performed. The ingestion fixity sub-state machine manages the following processes.

- **Check restore status -** Checks the storage class of the uploaded file using the `S3.HeadObject` API. If the file is in either the `GLACIER` or `DEEP_ARCHIVE` storage class, the Lambda function starts the restore process using the `S3.RestoreObject` API. This process also involves a state to check if the restore process is complete and transitions to computing the checksum. If not, it moves to one of the following wait states: a four-minute wait state if the restore tier is set to `Expedited`, a 12 hour wait state if the storage class is set to `DEEP_ARCHIVE` and the restore tier is set to Bulk, or a four-hour wait state for the rest of the file.
- **Compute checksum** - Incrementally computes the MD5 checksum of a 20 GB chunk of the file using the `S3.GetObject` byte range. This process also involves a choice state to check if the checksum is completed. If not, it computes the MD5 checksum on the next 25GB chunk of the file.
- **Validate checksum** – Compares the computed checksum value of the file against the previously computed MD5 checksum that is stored in the object metadata.

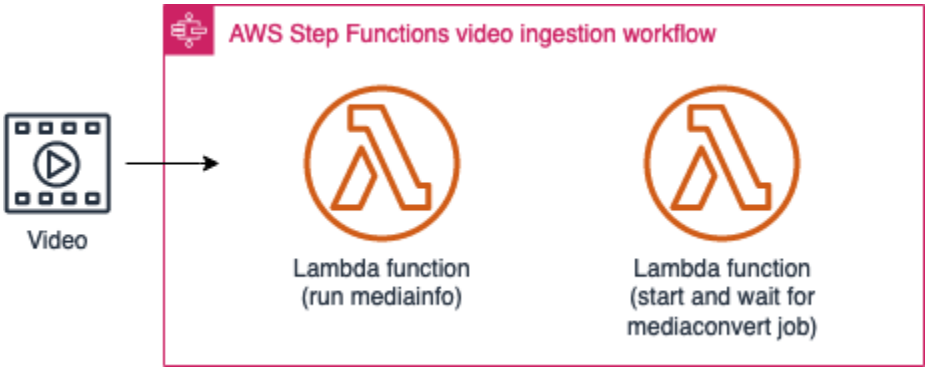


Ingestion fixity sub-state machine workflow

Video ingestion sub-state machine

The video ingestion sub-state machine processes video files using AWS Lambda functions. This state machine coordinates the following processes:

- **Run mediainfo** – Runs the MediaInfo tool to extract technical metadata from the video. The raw MediaInfo XML result is stored in an S3 proxy bucket.
- **Start and wait for mediaconvert job** - Creates a job template based on the media information extracted by MediaInfo. If the video file contains multiple audio tracks (an MXF file can contain eight to 16 audio tracks), the Lambda function selects the best combination of audio tracks, and runs AWS Elemental MediaConvert to create the proxy files and thumbnails. The proxy files and thumbnail images are stored in an S3 proxy bucket.



Video ingestion sub-state machine workflow

Image ingestion sub-state machine

The image ingestion sub-state machine processes images using AWS Lambda. This state machine coordinates the following process:

- **Run imageinfo** - Runs exiftool to extract EXIF information from the image file, generates an image proxy file, and stores the proxies to an S3 proxy bucket.

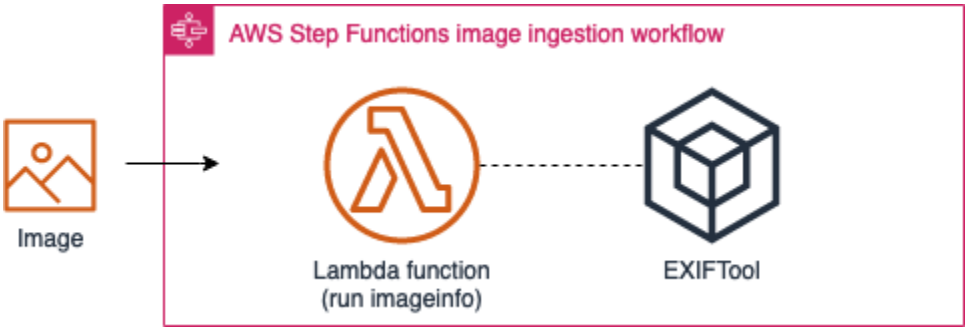


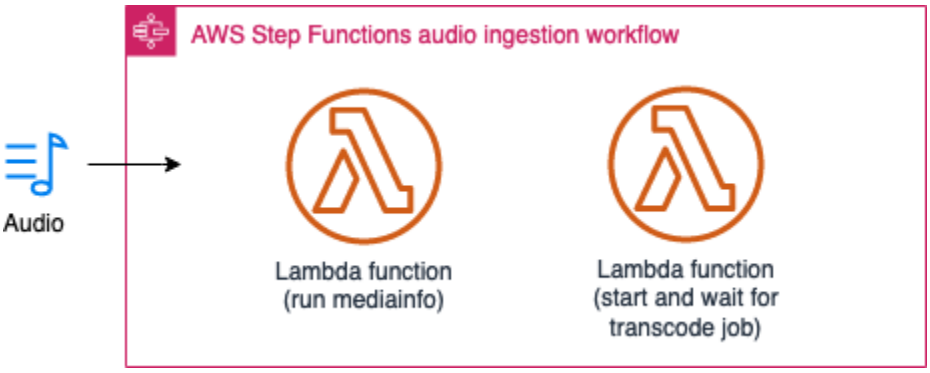
Image ingestion sub-state machine workflow

Audio ingestion sub-state machine

The audio ingestion sub-state machine processes audio files using AWS Lambda functions. This state machine coordinates the following processes:

- **Run mediainfo** – Runs the MediaInfo tool to extract technical metadata and cover art from the audio file. The raw MediaInfo XML result is stored in an S3 proxy bucket.

- **Start and wait for transcode job** – Uses AWS Elemental MediaConvert service to create a M4A audio proxy file for Amazon Transcribe processing and for streaming to the web application. The proxy files are stored in an S3 proxy bucket.

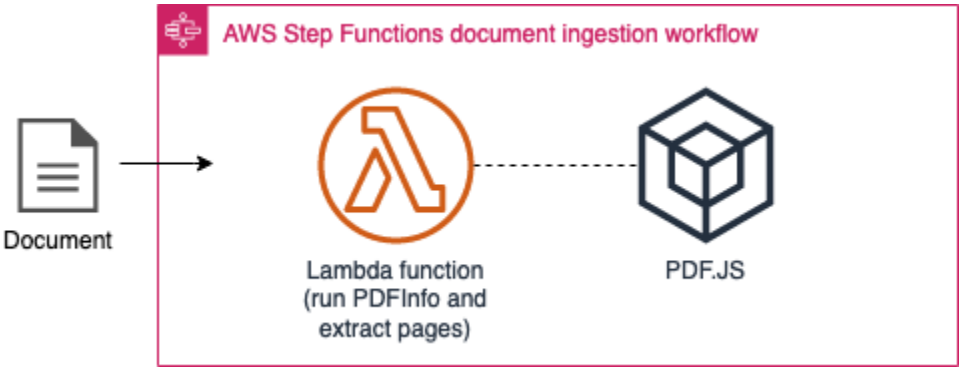


Audio ingestion sub-state machine workflow

Document ingestion sub-state machine

The document ingestion sub-state machine processes document files using AWS Lambda. This state machine coordinates the following process:

- **Run PDFinfo and extract pages** – Runs the PDF.JS tool to extract document metadata and converts pages to PNG image proxies. The PNG image proxies are stored in an S3 proxy bucket.



Document ingestion sub-state machine workflow

Analysis state machine

The Media2Cloud on AWS solution includes an analysis state machine that is composed of four different AWS Step Functions sub-state machines and a set of AWS Lambda functions that start,

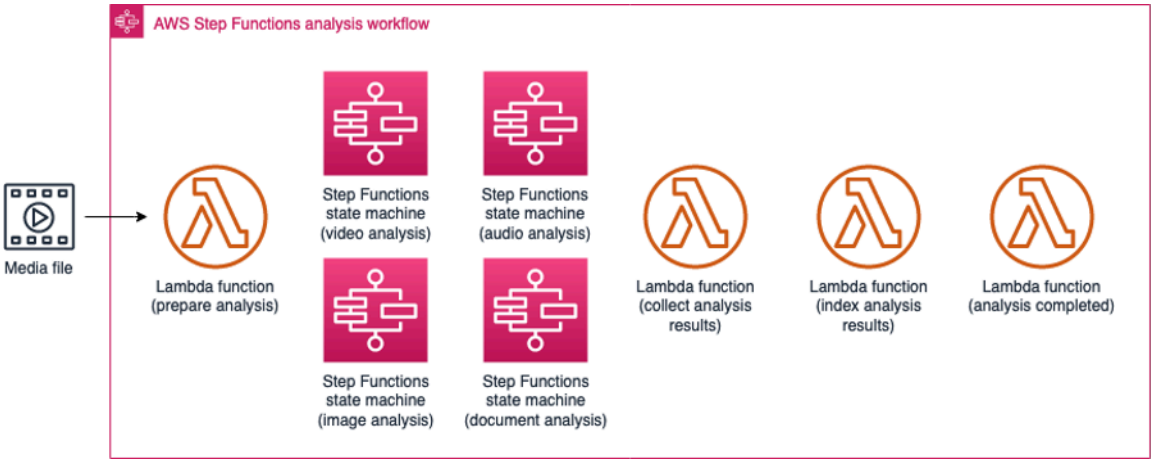
monitor, and collect results from the sub-state machines. The analysis state machine consists of the following sub-state machines:

- A video analysis sub-state machine that manages the video-based analysis process
- An audio analysis sub-state machine that manages the audio-based analysis process
- An image analysis sub-state machine that manages the image-based analysis process
- A document analysis sub-state machine that manages the document-based analysis process

When the ingestion process is completed, the solution automatically starts the analysis process. Similar to the ingestion state machine, the analysis state machine publishes progress and status to an AWS IoT topic. The web interface processes the progress status and displays it to the customer.

- **Prepare analysis** - Checks the incoming analysis request and prepares the optimal AI/ML analysis options to run based on the media type of the file and the availability of specific detections.
- **Video analysis enabled?** - Checks whether video analysis is activated by checking the `$.input.video.enabled` flag.
- **Start video analysis and wait (nested)** – Starts and waits for the video analysis sub-state machine where it runs Computer Vision (CV) analysis on the video using Amazon Rekognition if the media type of the file is a video. This follows a state to check if video analysis is activated.
- **Skip video analysis** - An end state to indicate video analysis is not activated.
- **Audio analysis enabled?** – Checks whether audio analysis is activated by checking the `$.input.audio.enabled` flag.
- **Start audio analysis and wait (nested)** - Starts and waits for the audio analysis sub-state machine where it runs speech-to-text and Natural Language Processing (NLP) analysis using Amazon Transcribe and Amazon Comprehend if the file is an audio file. This follows a state to check if audio analysis is activated.
- **Skip audio analysis** - An end state to indicate audio analysis is not activated.
- **Image analysis enabled?** - Checks whether image analysis is activated by checking the *$.input.image.enabled* flag.
- **Start image analysis and wait (nested)** - Starts and waits for the image analysis sub-state machine where it runs Computer Vision (CV) analysis using Amazon Rekognition if the file is an image type. This follows a state to check If image analysis is activated.
- **Skip image analysis** - An end state to indicate image analysis is not activated.

- **Document analysis enabled?** - Checks whether document analysis is activated by checking the `$.input.document.enabled` flag.
- **Start document analysis and wait (nested)** - Starts and waits for the document analysis sub-state machine where it runs Optical Character Recognition (OCR) analysis using Amazon Textract. This follows a state to check if document analysis is activated.
- **Skip document analysis** - An end state to indicate image analysis is not activated.
- **Collect analysis results** - Collects outputs from each sub-state machine by calling the `Step Functions.DescribeExecution` API, parses, and joins the results.
- **Analysis completed** - Updates the `analysis` field of the DynamoDB ingestion table to indicate the types of analysis that have been run. The Lambda function also creates records on the DynamoDB `aiml` table with information including *start time* and *end time* of each analysis detection, *pointers* to where the analysis metadata JSON results are stored in an Amazon S3 proxy bucket, the *job name* of the detection, and the ARN of the state machine run.



Analysis workflow

Video analysis sub-state machine

The video analysis sub-state machine is managed by the analysis state machine. It runs a series of Amazon Rekognition async processes to extract *faces*, *celebrities*, *labels*, *moderation*, and *face match* data from the video file. This sub-state machine consists of three parallel branches where each branch runs and monitors a specific Amazon Rekognition async process:

1. **Frame-based detectors:** This branch uses image frames of the video and Amazon Rekognition image APIs to analyze a video file (celebrity recognition, label detection, or moderation detection).

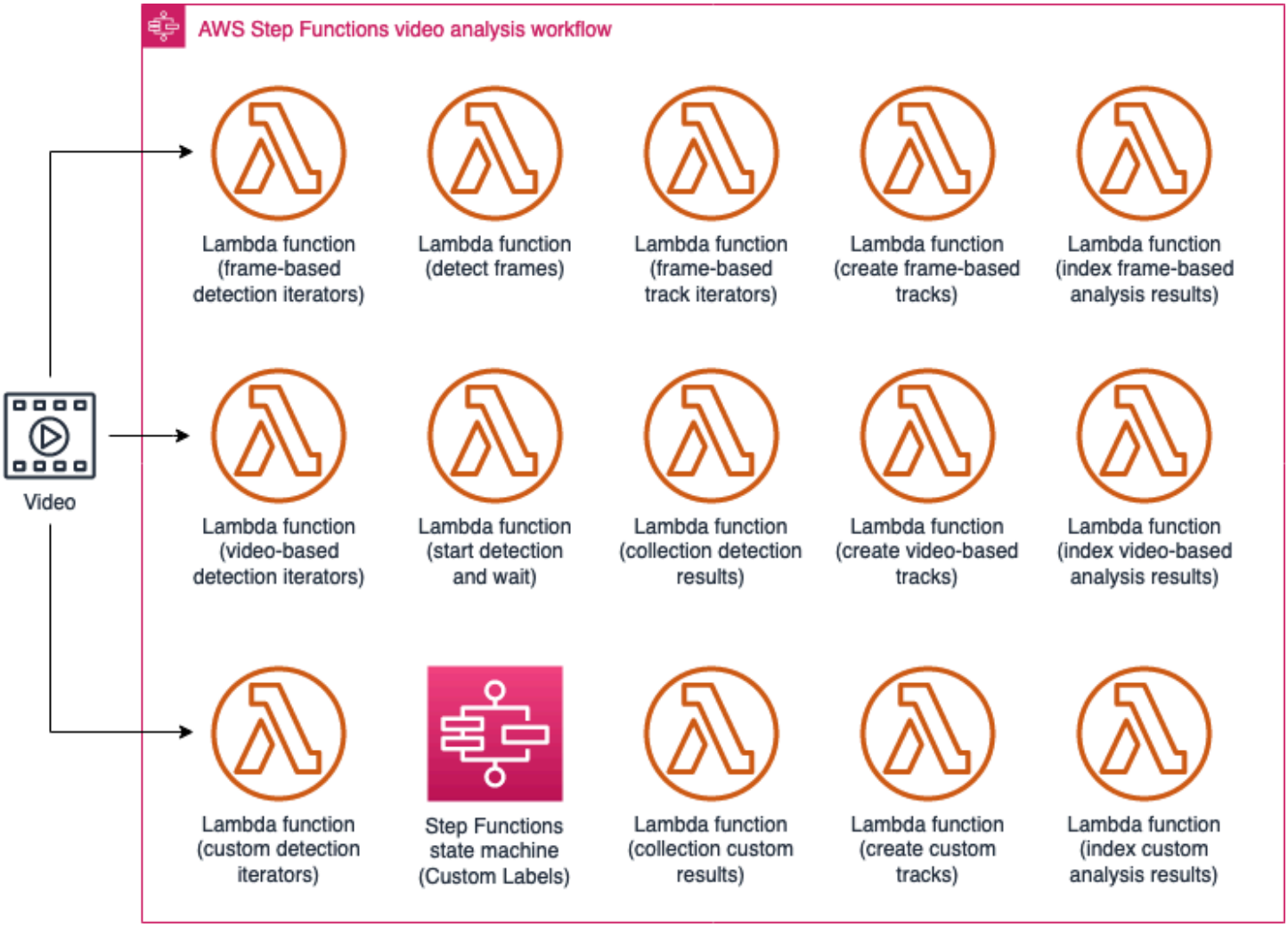
2. **Video-based detectors:** This branch uses Amazon Rekognition video APIs (segment detection or people pathing detection) to analyze a video file.
3. **Custom detectors:** This branch activates Amazon Rekognition Custom Labels detection using the image frames of a video file.

The branches that run during analysis depend on the APIs the user selects to use to process content. In the most complex scenario, the user selects *frame-based* analysis but also activates *segment* and *people pathing* detections which are available with Amazon Rekognition Video APIs, and Custom Label detection to detect specialized objects in the video. In this case, all the branches run where the frame-based branch runs analysis for *celebrity*, *label*, and *moderation* detections using Amazon Rekognition image APIs, the video-based branch runs *segment* and *people pathing* detections with Amazon Rekognition video APIs, and the custom-based branch runs Custom Labels detection using Amazon Rekognition Custom Label APIs. The state machine uses Step Functions Map state to parallelize the detection processes.

The video analysis sub-state machine orchestrates the following tasks using AWS Lambda functions:

- **Frame-based detection iterators** - Formats the user's selection as input into an array of a set of iterators for the map state. With the `Map` state, each iterator specifies the type of detection to run, for example, `celeb`, `label`, or `moderation` and pointers to the image frames.
- **Detect frame (iterator) -** Runs the detection specified in the iterator's input. For example, if the type is $.data.celeb, the Lambda function uses the `RecognizeCelebrities` API to detect celebrities. The Lambda function also stores the detection JSON result to an S3 proxy bucket. This process repeats until all the image frames are processed or when the Lambda runtime is close to the 15- minute limit.
- **Detect frames completed** - Indicates the detection of the specific type has completed.
- **Frame-based track iterators** – Joins the results from the previous iterators and prepares for the next Map state where it generates WebVTT tracks and timeseries tracks.
- **Create frame-based track (iterator)** - Fetches and parses the detection JSON result to create WebVTT track and timeseries track based on the timestamps of each of the detected item. The WebVTT track is used by the web application to activate on-screen display of the detection result. The timeseries data track is used by the web application to activate charts.
- **Index frame-based analysis (iterator)** - Uses the timeseries tracked created earlier and indexes the timestamped metadata document to an OpenSearch Service index. Each type of the detections, specifically, `celeb`, `label`, `text`, `moderation` has its own index.

- **Video-based detection iterators** - Formats the input into an array, a set of iterators for the Map state where each iterator specifies the type of detection to run specifically, `celeb`, `label` or `moderation` and pointers to the video file.
- Start detection and wait (iterator) - Starts and waits for a specific Amazon Rekognition video job for example, StartCelebrityDetection and StartLabelDetectionasynchronously.
- **Collect detection results (iterator)** - Collects the detection JSON results by calling Amazon Rekognition `GetLabelDetection` and `GetPersonTracking`. It then stores the JSON results to an S3 proxy bucket.
- **Create video-based track (iterator)** - Similar to **Create frame-based track (Iterator)** state.
- **Index video-based analysis (iterator)** - Similar to **Index frame-based analysis (Iterator)** state.
- **Custom detection iterators** - Formats the input into an array, a set of iterators for the `Map` state. The difference is that each iterator specifies the type of Custom Labels model to run.
- **Start custom and wait (iterator)** - Similar to **Frame-based detection iterators**, formats the input into an array, a set of iterators for the Map state. The difference is that each iterator specifies the type of Custom Labels model to run.
- **Collect custom results (iterator)** - Collects the Custom Label JSON results.
- **Create custom track (iterator)** - Similar to **Create frame-based track (Iterator)** state.
- **Index custom analysis (iterator)** - Similar to **Index frame-based analysis (Iterator)** state. The difference is that it indexes the document to the customlabel index in the OpenSearch Service cluster.
- **Video analysis completed** – Merges all outputs from the parallel branches.



Video analysis sub-state machine workflow

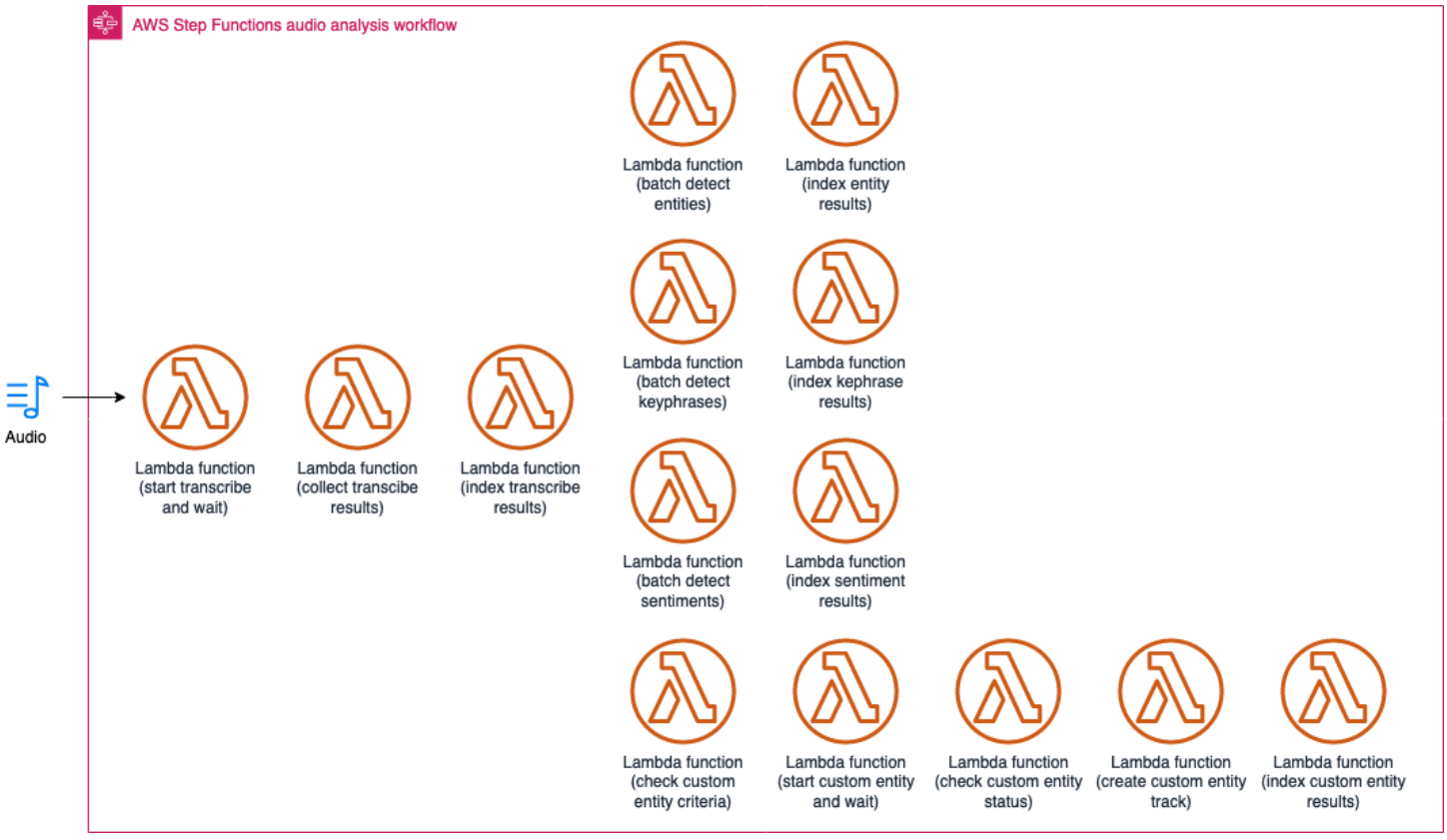
Audio analysis sub-state machine

The audio analysis sub-state machine is managed by the analysis state machine. This sub-state machine runs Amazon Transcribe and Amazon Comprehend to extract *transcription*, *entities*, *key phrases*, *sentiments*, *topic*, and *classification* metadata. This sub-state machine first runs Amazon Transcribe to convert speech to text and starts a number of branches in parallel where each branch runs and monitors a specific Amazon Comprehend process.

- **Start transcribe and wait** - Calls the Amazon Transcribe `StartTranscriptionJob` async API to start an Amazon Transcribe job for the audio file in batch mode and waits for the job to complete.

- **Collect transcribe results** – Ensures that the transcription job is completed (not failed). Also ensures that the transcription results (JSON file) and the Amazon Transcribe generated subtitle file (WebVTT format) are present and stored in an S3 proxy bucket.
- **Index transcribe results** - Downloads and parses the WebVTT file using an open-source package node-webvtt. Indexes the entities to the OpenSearch Service cluster under the `transcribe` index.
- **Batch detect entities** - Runs Amazon Comprehend BatchDetectEntities to extract entities from the transcription and converts the character-based offset to timestamp-based results. The Lambda function preserves and stores the *original* entity results from Amazon Comprehend service to an S3 proxy bucket and stores the timestamped version of metadata result (JSON) which is used to index the entities later on in an S3 proxy bucket.
- **Index entity results** - Downloads the metadata result and indexes the entity results to the OpenSearch Service cluster under the `entity` index.
- Batch detect key phrases - Runs Amazon Comprehend BatchDetectKeyPhrases and stores the original key phrase results and the timestamped metadata JSON file in an S3 proxy bucket.
- **Index key phrase results** - Downloads the metadata result and indexes the key phrase results to the OpenSearch Service cluster under the `keyphrase` index.
- **Batch detect sentiments -** Runs Amazon Comprehend BatchDetectSentiment and stores the original sentiment results and the timestamped metadata JSON file in an S3 proxy bucket.
- Index sentiment results – Downloads the metadata result and indexes the sentiment results to the OpenSearch Service cluster under the `sentiment` index.
- **Check custom entity criteria** - Ensures that the Custom Entity Recognizer is runnable and the language code of the transcription result is same as the language code of the trained recognizer. If the criteria are met, the state Lambda function prepares the documents to be analyzed, stores them in an S3 proxy bucket - *s3://PROXY_BUCKET/UUID/FILE_BASENAME/*raw/*DATETIME*/comprehend/customentity/*document-XXX.txt* and sets the `$.data.comprehend.customentity.prefix` field to indicate there are documents to be processed.
- **Can start custom entity?** - Determines custom entity detection can start by checking the `$.data.comprehend.customentity.prefix` flag. If the flag is present, transitions to **Start and wait custom entity** state. Otherwise, transitions to **Custom entity skipped** state.
- **Start and wait custom entity** - Starts the custom entity detection by invoking `StartEntitiesDetectionJob` API asynchronously.

- **Wait for custom entity statues (3 mins)** - A wait state for 3 minutes and transitions to a state to **Check custom entity status** state.
- **Check custom entity status** - Checks the entity job by calling `DescribeEntitiesDetectionJob` API and stores the job status to $.status.
- **Custom entity completed** - Checks `$.status`. If it is set to **COMPLETED**, transitions to **Create custom entity track** state. Otherwise, transitions back to **Wait for custom entity status (3mins)** state.
- **Create custom entity track** - Gets the entity JSON result, parses and converts the word-offset based results into a timestamp-based metadata results, and stores the JSON results to an S3 proxy bucket.
- **Index custom entity results** - Downloads the metadata result and indexes the custom entity results to the OpenSearch Service cluster under the `customentity` index.
- **Custom entity skipped** - An end state indicates that there is no custom entity being detected.



Audio analysis sub-state machine workflow

Image analysis sub-state machine

The image analysis sub-state machine is managed by the analysis state machine. It runs a series of Amazon Rekognition image (synchronized) processes to extract faces, celebrities, labels, moderation, face match, and texts from the video or image file.

Start image analysis - Runs the Amazon Rekognition Image APIs such as `RecognizeCelebrities`, `DetectFaces`, `SearchFacesByImage`, `DetectLabels`, `DetectModerationLabels`, and `DetectText` APIs to extract visual metadata from the image file and stores raw results to the proxy bucket: *s3://PROXY_BUCKET/UUID/FILE_BASENAME/raw/DATETIME/rekog-image/ANALYSIS_TYPE/output.json* where ANALYSIS_TYPE is `celeb`, `label`, `face`, `faceMatch`, `text`, or `moderation`.

- **Index analysis results** - Indexes all the JSON results to the OpenSearch Service indices.

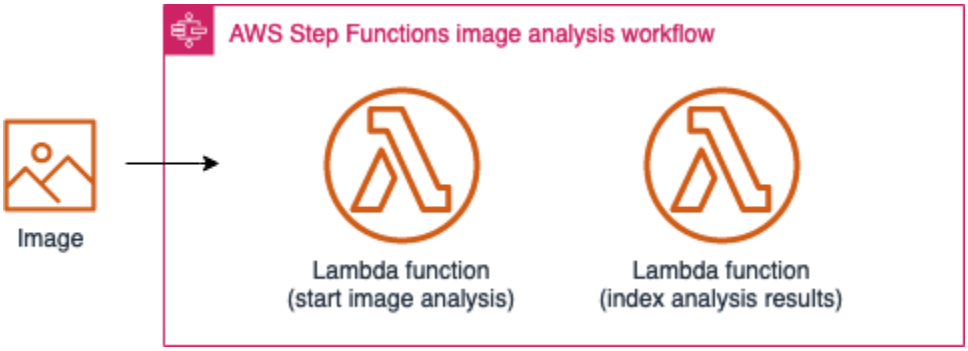
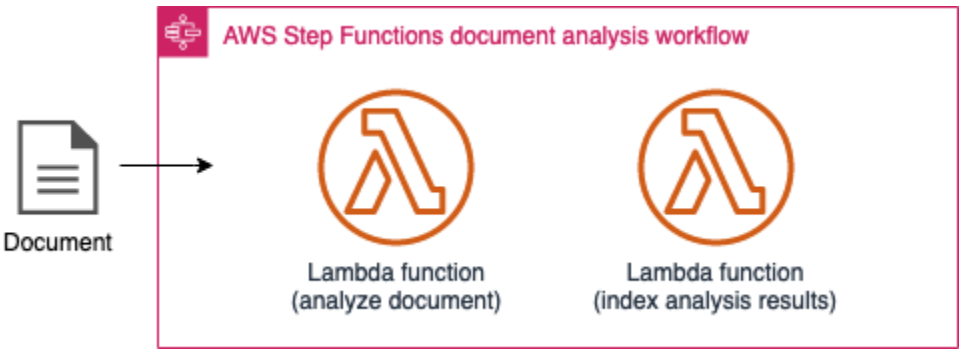


Image analysis sub-state machine workflow

Document analysis sub-state machine

- **Analyze document** – Calls the `AnalyzeDocument` API to extract tabular metadata of the document and stores the results in an S3 proxy bucket. The process repeats until all pages of the document has been processed or when the Lambda runtime is close to the 15-minute limit. In a case where more pages to be processed and the Lambda is approaching the 15-minute limit, the Lambda function sets $.status to **IN_PROGRESS** and `$.data.cursor` to the current page index to prepare for re-entering the same state to continue where it leaves off.
- **Index analysis results** - Indexes the metadata to the OpenSearch service `textract` index.

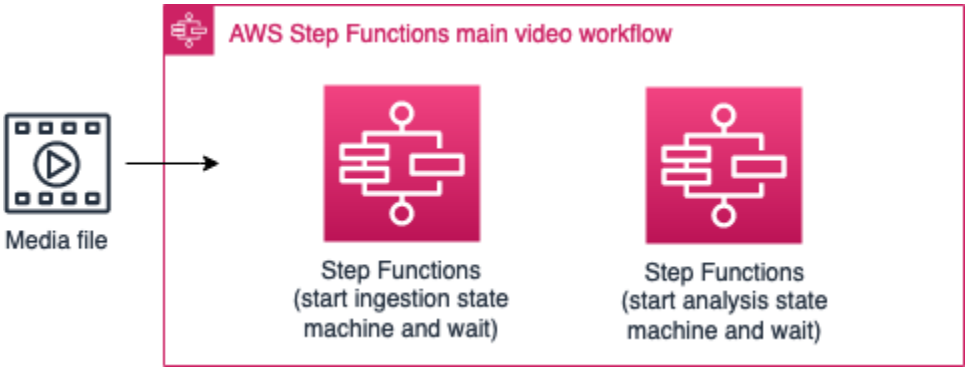


Document analysis sub-state machine workflow

Main state machine

The main state machine is the entry point to the backend ingestion and analysis workflows. The main state machine chains the backend ingestion and analysis workflows as sub-state machines:

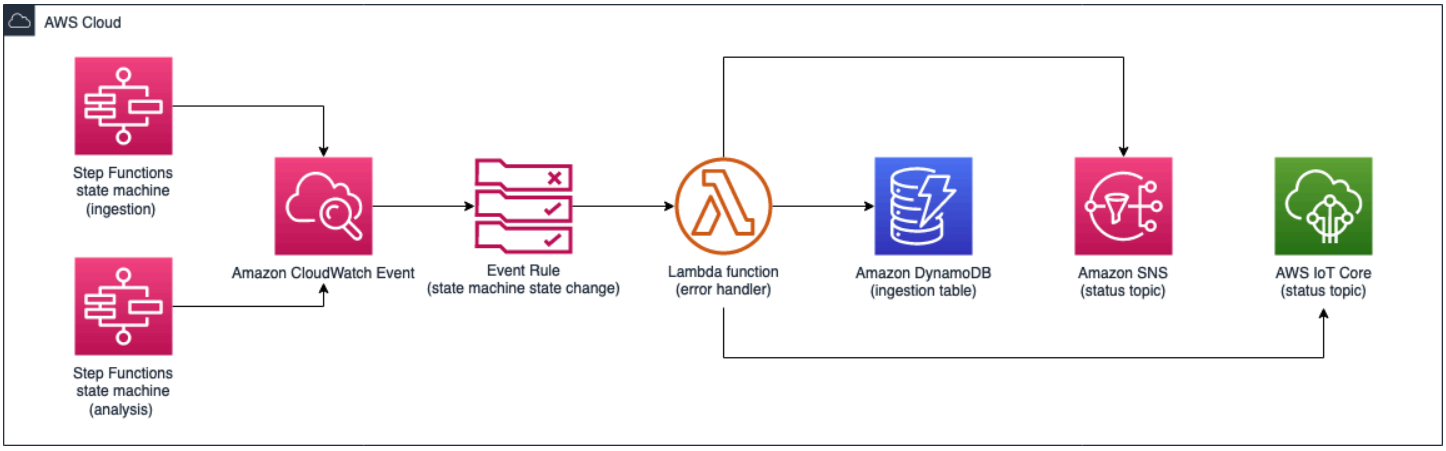
- **Start ingestion state machine (nested)** – Runs the ingestion process using the ingestion state machine when a customer uploads a media file to Media2Cloud on AWS.
- **Start analysis state machine (nested)** - Runs the analysis process using the analysis state machine after the ingestion state machine completes the ingestion process.



Main state machine workflow

State machine error handling

The error handling of the main state machine is handled by using an Amazon CloudWatch Event Rule attached to an AWS Lambda function.



State machine error handling

The event rule is configured to listen to `Step Functions Run Status Change` events with error statuses of **FAILED**, **ABORTED**, and **TIMED_OUT** of the ingestion and analysis state machine runs. It then invokes a Lambda `error-handler` function to process the state machine run error.

The CloudWatch Events pattern is defined as follows:

```
{
    "detail-type": [
        "Step Functions Execution Status Change"
    ],
    "source": [
        "aws.states"
    ],
    "detail": {
        "stateMachineArn": [
            "<INGEST_STATE_MACHINE_ARN>",
            "<ANALYSIS_STATE_MACHINE_ARN>"
        ],
        "status": [
            "FAILED",
            "ABORTED",
            "TIMED_OUT"
        ]
    },
    "region": [
        "<REGION>"
    ]
}
```

The `error-handler` Lambda function parses the run histories to find the last error from the state machine, updates the status in the Amazon DynamoDB ingestion table, publishes the status to an Amazon SNS status topic to notify subscribers, and publishes the status to an AWS Iot Core publish/subscribe service (a status topic) to notify the front-end web application.

Lifecycle policy

The Media2Cloud on AWS solution turns on Amazon S3 Intelligent Tiering storage class for the Amazon S3 ingestion, proxy, and web buckets.

For the S3 ingestion bucket, the solution applies additional lifecycle policy to transition objects to Amazon Glacier storage class after 90 days and Amazon Glacier Deep Archive storage class after 180 days.

For the S3 log bucket, the solution configures the lifecycle policy to keep the logs for seven days and turns off the versioning.

Amazon SNS notifications

The Media2Cloud on AWS solution sends Amazon SNS a publish notification to the subscriber when the Step Functions ingestion, analysis, and labeling state machines are completed, or when there is an error. You can automate other workflows, such as importing data to a Media Asset Management (MAM) system, by using the Amazon SNS topic subscription created by the solution. The following JSON notification messages can be customized to your needs.

Ingestion state machine notification message:

```
{
  "uuid": "<uuid>",
  "stateMachine": "<ingest-main-state-machine>",
  "operation": "job-completed",
  "overallStatus": "PROCESSING",
  "status": "INGEST_COMPLETED",
  "progress": 100,
  "input": {
    "bucket": "<ingest-bucket>",
    "duration": 91973,
    "framerate": 29.97,
    "destination": {
      "bucket": "<proxy-bucket>",
```

```
      "prefix": "<uuid>/Demo1/"
    },
    "attributes": {},
    "type": "video",
    "uuid": "<uuid>",
    "key": "Demo1/Demo1.mp4",
    "aiOptions": {
      "sentiment": true,
      "textROI": [
        false,
        false,
        false,
        false,
        false,
        false,
        false,
        false,
        false
      ],
      "framebased": false,
      "celeb": true,
      "frameCaptureMode": 0,
      "keyphrase": true,
      "label": true,
      "facematch": true,
      "transcribe": true,
      "face": true,
      "customentity": false,
      "person": false,
      "minConfidence": 80,
      "textract": true,
      "moderation": true,
      "segment": true,
      "customlabel": false,
      "text": true,
      "entity": true,
      "faceCollectionId": "FilmActors"
    }
  },
  "data": {
    "checksum": {
      "comparedResult": "MATCHED",
      "storeChecksumOnTagging": true,
      "computed": "3839bf07b1a69f15b5f1bb3356f5e500",
```

```
      "fileSize": 14392096,
      "startTime": 1638305617773,
      "endTime": 1638305618232,
      "comparedWith": "object-metadata",
      "tagUpdated": true,
      "algorithm": "md5"
    },
    "transcode": {
      "output": "<uuid>/Demo1/transcode/",
      "jobId": "1638305622955-7qvt1m",
      "startTime": 1638305623059,
      "endTime": 1638305641739
    },
    "restore": {
      "tier": "Bulk",
      "startTime": 1638305617599,
      "endTime": 1638305617599
    },
    "mediainfo": {
      "output": [
        "<uuid>/Demo1/mediainfo/mediainfo.json",
        "<uuid>/Demo1/mediainfo/mediainfo.xml"
      ]
    },
    "indexer": {
      "terms": [
        "overallStatus",
        "lastModified",
        "status",
        "timestamp",
        "basename",
        "attributes",
        "bucket",
        "fileSize",
        "mime",
        "framerate",
        "uuid",
        "key",
        "duration",
        "md5",
        "type"
      ]
    }
  }
```

```
}
```

Table 2: Ingestion state machine notification message key name descriptions

Key name	Description
operation	Last state of the state machine: `job-completed`
uuid	UUID of the file
stateMachine	Ingestion state machine name
overallStatus	Status of the next step of the workflow
status	Status of the state machine: `INGEST_COMPLETED`
progress	Progress of the state machine
input	Input parameters
input.bucket	Ingestion bucket where the uploaded file is stored
input.duration	Duration of the file in milliseconds
input.framerate	Frame rate of the uploaded file
destination	Location where proxy files will be stored
destination.bucket	S3 Bucket created to store proxy files
destination.prefix	Folder structure created within the S3 proxy file bucket
attributes	Object attributes
type	MIME type of the file that has been uploaded

Key name	Description
uuid	UUID of the file
key	Path to the uploaded file in the S3 proxy file bucket
aiOptions	User selected AI services to run on the uploaded asset
aiOptions.sentiment	Boolean value indicates if sentiment detection is activated
aiOptions.textROI	Boolean value indicates if text regions of interest is activated
aiOptions.framebased	Boolean value indicates if frame-based analysis is activated
aiOptions.celeb	Boolean value indicates if celebrity detection is activated
aiOptions.frameCaptureMode	Integer value indicates if frame capture mode is activated
aiOptions.keyphrase	Boolean value indicates if Amazon Comprehend key phrase detection analysis is activated
aiOptions.label	Boolean value indicates if label detection is activated
aiOptions.facematch	Boolean value indicates if face match is activated
aiOptions.transcribe	Boolean value indicates if Amazon Transcribe is activated
aiOptions.face	Boolean value indicates if face detection is activated

Table 2: Ingestion state machine notification message key name descriptions

Key name	Description
aiOptions.customentity	Boolean value indicates if Amazon Comprehend custom entity recognizer is activated
aiOptions.person	Boolean value indicates if person detection is activated
aiOptions.minConfidence	An integer representing the confidence level results to display in the web UI
aiOptions.textract	Boolean value indicates if Amazon Textract is activated
aiOptions.moderation	Boolean value indicates if Amazon Rekognition moderation detection is activated
aiOptions.segment	Boolean value indicates if segment detection is activated
aiOptions.customlabel	Boolean value indicates if custom labels is activated
aiOptions.text	Boolean value indicates if text detection is activated
aiOptions.entity	Boolean value indicates if entity detection is activated
aiOptions.faceCollectionId	If a user activated the use of a face collection to match faces found in a file the name of the face collection will be displayed
data.checksum	Gathers the information from the checksum comparison process
data.checksum.compareResult	Status of the checksum comparison. `MATCHED` is success

Table 2: Ingestion state machine notification message key name descriptions

Key name	Description
data.checksum.storeChecksumOnTagging	A flag indicates if the checksum value is stored in Amazon S3 object tagging
data.checksum.computed	The computed MD5 or SHA1 checksum
data.checksum.fileSize	File size of the uploaded file
data.checksum.startTime	Start time of the checksum process
data.checksum.endTime	End time of the checksum process
data.checksum.comparedWith	Indicates how we compare the *computed* checksum object-metadata refers to `x-amz-metadat-md5` is used for comparison object-tagging refers to an existing `computed-md5` object tag is used for comparison object-etag refers to the object `ETag` value is used for comparison
data.checksum.tagUpdated	Indicate if the solution successfully updates the object tagging with the computed checksum value
data.checksum.algorithm	Checksum algorithm type: `md5` or `sha1`
transcode.output	Location of the transcoded proxies in an S3 proxy bucket
transcode.jobId	AWS Elemental MediaConvert job ID when the solution creates proxies
transcode.startTime	Start time of AWS Elemental MediaConvert job

Key name	Description
transcode.endTime	End time of AWS Elemental MediaConvert job
restore.tier	If object is in `GLACIER` or `DEEP_ARCHIVE` storage the process for retrieval
restore.startTime	If object is in `GLACIER` or `DEEP_ARCHIVE` storage, `startTime` indicates the start time of the restore process
restore.endTime	If object is in `GLACIER` or `DEEP_ARCHIVE` storage, `endTime` indicates the end time of the restore process
mediainfo.output	Location of the raw mediainfo XML output in an S3 proxy bucket
indexer.terms	A list of terms indexed to Amazon OpenSearch Service cluster

Analysis state machine notification message:

```
{
 "uuid": "<uuid>",
 "stateMachine": "<analysis-main-state-machine>",
 "operation": "job-completed",
 "overallStatus": "COMPLETED",
 "status": "ANALYSIS_COMPLETED",
 "progress": 100,
 "input": {
   ...,
   "video": {
     "enabled": true,
     "key": "<uuid>/Demo1/transcode/aiml/Demo1.mp4"
   },
   "audio": {
     "enabled": true,
     "key": "<uuid>/Demo1/transcode/aiml/Demo1.m4a"
   },
```

```
    "image": {
      "enabled": false
    },
    "document": {
      "enabled": false
    },
    "request": {
      "timestamp": 1638305649796
    },
    "metrics": {
      "duration": 91973,
      "requestTime": 1638305649796,
      "startTime": 1638305651875,
      "endTime": 1638305714837
    }
  },
  "data": {
    "video": {
      "status": "COMPLETED",
      "startTime": 1638305652358,
      "endTime": 1638305712306,
      "executionArn": "<analysis-video-state-machine-arn>",
      "rekognition": {
        "[type]": {
          "startTime": 1638305654505,
          "endTime": 1638305693000,
          "backlogId": "<random-id> ",
          "jobId": "<job-id>",
          "numOutputs": 1,
          "output": "<uuid>/Demo1/raw/<time>/rekognition/[type]/mapFile.json",
          "metadata": "<uuid>/Demo1/metadata/[type]/",
          "timeseries": "<uuid>/Demo1/timeseries/[type]/",
          "vtt": "<uuid>/Demo1/vtt/[type]/"
        }
      }
    },
    "audio": {
      "status": "COMPLETED",
      "startTime": 1638305652379,
      "endTime": 1638305710319,
      "executionArn": "<analysis-audio-state-machine-arn>",
      "transcribe": {
        "output": "<uuid>/Demo1/raw/<time>/transcribe/<job-id>.json",
        "jobId": "<job-id>",
```

```
        "startTime": 1638305654014,
        "endTime": 1638305707564,
        "languageCode": "en-US",
        "vtt": "<uuid>/Demo1/raw/<time>/transcribe/<job-id>.vtt"
      },
      "comprehend": {
        "[type]": {
          "startTime": 1638305708528,
          "endTime": 1638305708956,
          "output": "<uuid>/Demo1/raw/<time>/comprehend/[type]/output.manifest",
          "metadata": "<uuid>/Demo1/metadata/[type]/output.json"
        }
      }
    },
    "src": {
      "bucket": "<ingest-bucket>",
      "key": "Demo1/Demo1.mp4",
      "type": "video"
    }
  }
}
```

Table 3: Analysis state machine notification message key name descriptions

Key Name	Description
uuid	UUID of the file
stateMachine	Analysis state machine name
operation	Last state of the state machine: `job-completed`
overallStatus	Overall status of the sate machine: `COMPLETED`
status	Status of the state machine: `ANALYSIS_COMPLETED`
progress	Progress of the state machine

Key Name	Description
input	Input parameters similar to Ingestion state machine notification message with the additional parameters described below
Input.video.enabled	Boolean value indicating if the object is a video
input.video.key	Path to the file
input.audio.enabled	Boolean value indicating if the video file has audio
input.audio.key	Path to the file
input.image.enabled	Boolean value indicating if the object is an image
input.document.enabled	Boolean value indicating if the object is a document
input.request.timestamp	Time that the analysis job was submitted
input.metrics.duration	Length of the file
intput.metrics.requestTime	Time the analysis job was initiated
input.metrics.startTime	Start time of the analysis job
input.metrics.endTime	End time of the analysis job
data.video	Video analysis information
data.video.status	Status of the video analysis process
data.video.startTime	Start time of the video analysis process
data.video.endTime	End time of the video analysis process
data.video.executionArn	Video analysis state machine run ARN

Table 3: Analysis state machine notification message key name descriptions

Key Name	Description
data.video.rekognition	Amazon Rekognition results
data.video.rekognition.[type]	Amazon Rekognition detection results of a specific [type] that refers to `celeb`, `face`, `faceMatch` , `label`, `segment`, `text`, `person`, or `moderation` .
data.video.rekognition.[type].startTime	Start time of the Amazon Rekognition detection process
data.video.rekognition.[type].endTime	End time of the Amazon Rekognition detection process
data.video.rekognition.[type].backlogId	ID of the job in the backlog
data.video.rekognition.[type].jobId	Amazon Rekognition job ID
data.video.rekognition.[type].numOutputs	Integer showing the number of outputs from this analysis
data.video.rekognition.[type].output	Location of the raw results from Amazon Rekognition process stored in an S3 proxy bucket
data.video.rekognition.[type].metadata	Location of the metadata tracks the solution generated stored in an S3 proxy bucket
data.video.rekognition.[type].timeseries	Location of the time series metadata tracks the solution generated stored in an S3 proxy bucket
data.video.rekognition.[type].vtt	Location of the WebVTT tracks the solution generated stored in an S3 proxy bucket
data.audio	Audio analysis results
data.audio.status	Status of the audio analysis process

Table 3: Analysis state machine notification message key name descriptions

Key Name	Description
data.audio.startTime	Start time of the audio analysis process
data.audio.endTime	End time of the audio analysis process
data.audio.executionArn	Audio analysis state machine run ARN
data.audio.transcribe	Amazon Transcribe results
data.audio.transcribe.output	Location of the raw results from Amazon Transcribe service stored in an S3 proxy bucket
data.audio.transcribe.jobId	Amazon Transcribe job ID
data.audio.transcribe.startTime	Start time of the transcribe process
data.audio.transcribe.endTime	End time of the transcribe process
data.audio.transcribe.languageCode	The language code provided to Amazon Transcribe for the language the transcript should be in
data.audio.transcribe.vtt	Location of the WebVTT tracks the solution generated, stored in an S3 proxy bucket
data.audio.comprehend	Amazon Comprehend results
data.audio.comprehend.[type]	Amazon Comprehend detection results of a specific [type] that refers to `entity`, `keyphrase` , or `sentiment` .
data.audio.comprehend.[type].startTime	Start time of Amazon Comprehend detection process
data.audio.comprehend.[type].endTime	End time of Amazon Comprehend detection process

Table 3: Analysis state machine notification message key name descriptions

Key Name	Description
data.audio.comprehend.[type].output	Location of the raw results from Amazon Comprehend detection process, stored in an S3 proxy bucket
data.audio.comprehend.[type].metadata	Location of the metadata tracks the solution generated, stored in an S3 proxy bucket
data.source	Information about the source file
data.source.bucket	Name of the S3 ingestion bucket
data.source.key	Object key
data.source.type	MIME type of the object being analyzed

State machine error notification message:

```
{
  "uuid": "<uuid>",
  "stateMachine": "<state-machine-name>",
  "status": "ERROR",
  "errorMessage": "Cannot destructure property `workerId` of 'undefined' or 'null'.",
  "input": {
    "uuid": "<uuid>"
  }
}
```

Table 4: State machine error notification message key name descriptions

Key Name	Description
uuid	UUID of the file
stateMachine	The state machine that generated the error

Key Name	Description
status	ERROR
errorMessage	The detailed error message
input	The input parameter used to start the state machine

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 amount of data being ingested and analyzed, running the solution's OpenSearch Service cluster, and the size and length of media files analyzed with Amazon Rekognition, Amazon Transcribe, and Amazon Comprehend.

As of this revision, the cost for running this solution on 100 hours of videos totaling one terabyte with the default settings in the US East (N. Virginia) Region is **$2,149.95 (one time processing)** with **$104.60/month (recurring)** for Amazon S3 data storage and OpenSearch Service search engine.

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. For customers who want to process large-scale video archives, we recommend that you contact your AWS account representative for at-scale pricing.

Example monthly cost

The following example is for a total file size of one terabyte, which equates to one hundred total hours of video content where each video is one hour in duration. The cost is broken down to the following categories:

1. **Migration cost** – When the video files are uploaded and stored in Amazon Glacier Deep Archive storage. The cost is estimated based on the total size of the video files.
2. **Ingestion cost** – When the uploaded video files are transcoded with AWS Elemental MediaConvert to create low resolution proxy video files plus the ingestion workflow cost composed of AWS Step Functions state transitions and AWS Lambda compute runtime, and Amazon DynamoDB Read/Write request units.
3. **Analysis cost** – When proxy files are analyzed with Amazon Rekognition, Amazon Transcribe, and Amazon Comprehend plus the analysis workflow cost composed of AWS Step Functions

state transitions and AWS Lambda Compute runtime, and Amazon DynamoDB Read/Write request units.

4. **Search engine cost** – When the generated metadata are indexed to an OpenSearch Service cluster. The cost depends on the number of dedicated nodes, the number of instance nodes, and the amount of Amazon EBS volume.

AWS service	Dimensions	Cost [USD]
Migration cost (one terabyte)		
S3 Glacier Deep Archive	$0.00099 per GB / Month * 1024 GB	$1.01
Ingestion cost (100 hours)		
AWS Elemental MediaConvert (SD, AVC with Professional Tier)	$0.012 per minutes * 100 hours	$72.00
AWS Elemental MediaConvert (Audio only)	$0.003 per minutes * 100 hours	$18.00
AWS Step Functions State transitions, AWS Lambda Compute unit (MB per 1ms), and Amazon DynamoDB Read Write Request Units	Varies depending on number of state transitions, the Lambda function memory size and runtime duration, and read write request to DynamoDB tables.	~$1.05
Analysis cost (100 hours)		
Amazon Rekognition Celebrity Recognition	$0.10 per minute * 100 hours	$600.00
Amazon Rekognition Label Detection	$0.10 per minute * 100 hours	$600.00

AWS service	Dimensions	Cost [USD]
Amazon Rekognition Segment Detection (Shot and Technical Cues detections)	($0.05 + $0.05 per minute) * 100 hours	$600.00
Amazon Transcribe	$0.024 per minute * 100 hours	$144.00
Amazon Comprehend Key Phrase Extraction	$0.0001 per unit Vary depends on number of characters extracted from audio dialogue of the video files	~$5.00
Amazon Comprehend Entity Recognition	$0.0001 per unit Vary depends on number of characters extracted from audio dialogue of the video files	~$5.00
AWS Step Functions State transitions, AWS Lambda Compute unit (MB per 1ms), and Amazon DynamoDB Read Write Request Units	Varies depending on number of state transitions, the Lambda function memory size and runtime duration, and read write request to DynamoDB tables.	~$ 0.30
Search engine cost		
Amazon OpenSearch Service dedicated node (t3.small .search)	$0.036 per hour * 0 node	$0.00
Amazon OpenSearch Service instance node (m5.large .search)	$0.142 per hour * 1 node	$102.24

AWS service	Dimensions	Cost [USD]
Amazon OpenSearch Service EBS Volume (GP2)	$0.135 per GB / month * 10 GB	$1.35
Total cost (based on one terabyte with 100 hours of videos)		
Monthly recurring cost (S3 storage and Amazon OpenSearch Service cluster)	$1.01 + $102.24 + 1.35	$104.60
One-time processing cost (AWS Elemental MediaConvert, Amazon Rekognition, Transcribe, Comprehend, AWS Step Functions, AWS Lambda)	($72 + $18) + $1.05 + ($600 + $600 + $600) + $144 + ($5 + $5) + $0.30	$2,045.35
Total:		**$2,149.95**

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.

Server-side encryption

AWS highly recommends that customers encrypt sensitive data in transit and at rest. This solution automatically encrypts media files and metadata at rest with Amazon S3 server-side encryption (SSE).The solution's Amazon Simple Notification Service (Amazon SNS) topics and Amazon DynamoDB tables are also encrypted at rest using SSE.

Amazon CloudFront

This solution deploys a static website hosted in an Amazon S3 bucket. To help reduce latency and improve security, this solution includes an Amazon CloudFront distribution with an origin access identity, which is a special CloudFront user that helps restrict access to the solution's website bucket contents. For more information, refer to Restricting access to Amazon S3 content by using an origin access identity.

Amazon OpenSearch Service

Documents indexed to the Amazon OpenSearch Service cluster are encrypted at rest. Node-to-node communication within the cluster is also encrypted.

Search engine sizing

The CloudFormation template provides presets for the end user to configure different Amazon OpenSearch Service clusters: **Development and Testing**, **Suitable for Production Workload**, **Recommended for Production Workload**, and **Recommended for Large Production Workload**.

- **Development and Testing** – This preset creates an Amazon OpenSearch Service cluster in a single Availability Zone with a single `m5.large.search` data node, 10GB storage, and without dedicated primary node.
- **Suitable for Production Workflow** – This preset creates an Amazon OpenSearch Service cluster in two Availability Zones with two `m5.large.search` data nodes, 20GB storage, and three dedicated `t3.small.search` primary nodes.
- **Recommended for Production Workload** – This preset creates an Amazon OpenSearch Service cluster in two Availability Zones with four `m5.large.search` data nodes, 20GB storage, and three dedicated `t3.small.search` primary nodes.
- **Recommended for Large Production Workload** – This preset creates an Amazon OpenSearch Service cluster in three Availability Zones with six `m5.large.search` data nodes, 40GB storage, and three dedicated `t3.small.search` primary nodes.

Integrated partners

The Media2Cloud on AWS solution is designed to provide a standardized architecture to support AWS Partners to integrate with content from AWS customers. A standardized architecture helps

accelerate the migration and supply chain process, and helps Media Asset Manager (MAM) partners provide solutions for their customers.

The Media2Cloud on AWS solution integrates with the following AWS Partners:

Cloudfirst.io

Cloudfirst.io is an AWS Partner that specializes in large-scale, unstructured, active archive, and content storage management solutions for Media and Entertainment. They actively assist clients with legacy archive migrations embracing various next-generation technologies. Cloudfirst provides consulting and a product called Rapid Migrate that address the challenges of moving content out of existing LTO archives, process and move content into Amazon S3 storage in supported content and metadata formats for Media2Cloud on AWS to initiate the ingestion process.

Levels Beyond

Levels Beyond is an AWS Partner that provides a Media Asset Manager (MAM) service platform called Reach Engine. Levels Beyond can be integrated with the Media2Cloud on AWS solution through Amazon SNS and interface with the output to consume the JSON formatted metadata to provide customers with a rich search, discovery and management service to manage their content archives. Levels Beyond can support customers further by configuring the services to add additional metadata faceting as well as automating the processing of content for production, OTT, digital publishing and other content related services.

Nomad CMS

Nomad CMS is an AWS Partner that supports the ability to bring an OTT metadata enrichment and discovery system to existing Amazon S3 assets. Nomad augments Amazon S3 asset storage without requiring any changes to the existing asset structure or files themselves. Nomad also automatically integrates with Media2Cloud on AWS and other AWS AI/ML services. Confidence scores, labels, transcriptions, and other AI enrichment is used to tag each asset with appropriate discovery information. Searching and publishing activities are used to make the resulting metadata available to custom solutions or in support of other integration activities.

EditShare

EditShare is an AWS Partner that designs and delivers high-performance, scalable, shared storage solutions that allow media professionals to create outstanding content. EditShare's EFSv with

FLOW is a Media2Cloud on AWS activated, end-to-end cloud production solution. It supports tiered asset storage, media management, intelligent archiving, and broad compatibility with creative tools such as the Adobe Creative Suite. Highlighting only one use case, EFSv and FLOW powered workflows have fast search and seamless switching between proxy and high-resolution editing right in the video editorial application. EditShare's Professional Services team can offer AWS customers seamless workflows designed around their business processes, leveraging solutions from EditShare and other providers.

eMAM

eMAM is an AWS Partner that powers workflows for production, post-production, sharing, and distribution: the entire lifecycle of a digital asset. eMAM provides a web interface designed to support non-technical users, providing a collaboration nexus for editors and designers using integrations into Apple Final Cut and Adobe Creative Cloud applications. eMAM is flexible, with easy configuration and scalability for the entire range of use cases and verticals, to provide customers with choice and control. eMAM provides a range of options for deployment including AWS cloud and hybrid solutions. eMAM is available as a permanent license or as a subscription in the AWS Marketplace with SaaS/PaaS-Server options.

Evertz

Evertz is an AWS Partner that provides the Mediator-X, a cohesive, highly scalable, infrastructure agnostic platform for Media Asset Management, Transmission Playout and Non-Linear delivery applications. Evertz Mediator-X allows customers to manage their *Cloud Content Factory* at scale using a rich feature set of integrations and options under the functional blocks of acquisition, processing, management, production, playout, and delivery. Utilizing Media2Cloud on AWS and other AWS services, customers can gather and store both metadata and content in highly durable cloud storage, use the intuitive user-interface to visualize machine learning data alongside other customer-specific metadata or pull data from API endpoints within the Mediator-X platform.

IMT

IMT is an AWS Partner that provides SoDA, a new way to control data movement between storage tiers, on-site and in the cloud. Since its inception 13 years ago, IMT has grown to become a leading next-gen Systems Integrator supporting over 800+ customers in Media & Entertainment, broadcast, sports, and corporate video in North America. SoDA is IMT's Intelligent Data Management Software that can be leveraged as a simple data migration tool to help customers move off from legacy archives, as well as broker data movement to and from the cloud

to various endpoints. Designed to work with all types of storage—on-premises, hybrid, and AWS—users can define rich, flexible policies or manually transfer data. SoDA plugs into multiple MAM solutions to empower end users to control their own data movement.

Quantiphi

Quantiphi is an AWS partner that provides an AI-powered Media Intelligence solution that helps media and entertainment customers unlock hidden data potential to curate better content, enhance customer targeting, and implement effective channel strategies to transform their customer experience.

Signiant

Signiant is an AWS Partner that offers fast and secure movement of large data sets over any IP network. Signiant provides foundational technology that allows content exchange within and between companies of all sizes to connect the global media supply chain. The Signiant Software-Defined Content Exchange (SDCX) SaaS platform provides people and systems with access to media assets located across disparate and distributed storage repositories # and lays the groundwork for innovations that extend beyond file transfer. Signiant's proprietary transport technology is the foundation upon which Signiant was built. Signiant's continued investment in this area has allowed them to remain a leader in the category for more than 15 years, and their software is relied upon to move petabytes of high-value content every day. Each Signiant product capitalizes on their advanced acceleration technology to transfer content up to 100 times faster than standard Internet transmission speeds, and Signiant technology is capable of moving any size of file or data set over any IP network, while taking advantage of all available bandwidth.

Starchive

Starchive is an AWS Partner that offers a command center for today's content producers. Starchive brings the power of digital asset management to the entrepreneur and small/medium business with the elegance of a modern consumer SaaS application and at a fraction of the cost of comparable solutions. Starchive helps users find the signal in the noise of their digital chaos and get back to work building their brand, business, and bottom line. In a world where every individual has the power to create and the opportunity to consume digital media 24/7—every business has the mandate to be a content powerhouse to thrive. Learn more about how Starchive used Media2Cloud on AWS to help Essence Magazine support their 50-year anniversary by improving the accessibility to their historical archive.

TrackIt

TrackIt is an AWS Advanced Consulting Partner with decades of experience in the Media & Entertainment industry and a wealth of cloud technology design and deployment work performed for many media-centric companies. TrackIt has experience building advanced pipelines that include AI/ML tools and integration with asset management systems, along with transcoding, rendering, VOD, OTT, live streaming, cloud-based editorial, and collaborative online tools. Learn more about how Trackit used Media2Cloud on AWS to help Jukin Media improve the utility of their archive.

Supported AWS Regions

This solution can be deployed to any AWS Region. If a service, such as Amazon Rekognition, is not currently available in the Region, the solution reduces its functionality. Analysis features such as *celebrity recognition*, *label detection*, and *face detection* will be turned off.

We recommend for you to launch the solution in an AWS Region where Amazon Rekognition, Amazon Transcribe, and Amazon Comprehend are available. For the most current availability of AWS services by Region, refer to the AWS Regional Services List.

As of the latest revision, this solution is fully supported in the following Regions:

Region name	
US East (Ohio)	Asia Pacific (Sydney)
US East (N. Virginia)	Canada (Central)
US West (Oregon)	Europe (Frankfurt)
Asia Pacific (Mumbai)	Europe (Ireland)
Asia Pacific (Seoul)	Europe (London)
Asia Pacific (Singapore)	AWS GovCloud (US-West)

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 quota

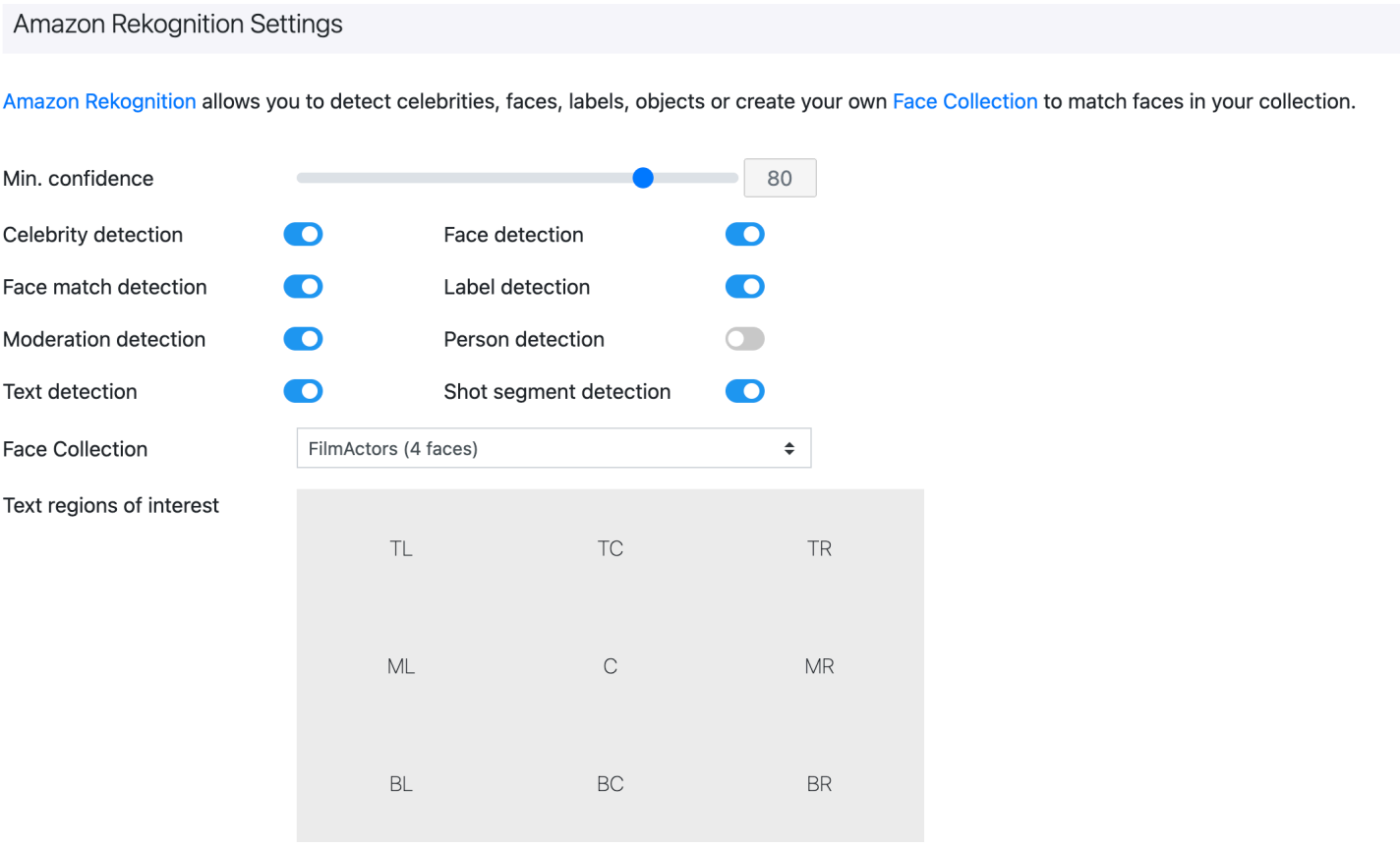
Your AWS account has AWS CloudFormation quotas that you should be aware of when launching the stack in this solution. By understanding these quotas, you can avoid limitation errors that would prevent you from deploying this solution successfully. For more information, see AWS CloudFormation quotas in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

Amazon Transcribe

Amazon Transcribe can process files up to four hours in length, this is the Maximum media duration. For more information, refer to Amazon Transcribe endpoints and quotas.

Amazon Recognition

The Amazon Rekognition Custom Labels setting is currently limited to running up to two models. For more information, refer to Guidelines and quotas in Amazon Rekognition Custom Labels in the *Amazon Rekognition Custom Labels Guide*.



Amazon Rekognition default detection settings

Deploy the solution

This solution uses AWS 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.

Deployment process overview

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

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

Time to deploy: Approximately 25 minutes

Step 1: Launch the Stack

- Launch the AWS CloudFormation template into your AWS account.
- Enter values for required parameters.
- Review the other template parameters, and adjust if necessary.

Step 2: Upload a video or image file

- Upload a file using the web interface to begin the ingestion and analysis workflows.

Step 3: Create your face collection

- Index faces to create your face collection to improve face analysis results.

Step 4: Advanced search

- Find the specific moment you are looking for.

Step 5: Customizing AI/ML settings

- Configure the AI/ML services that you want to use in your analysis.

Step 6: Viewing statistics

- A summary of all content in your collection.

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

media2cloud.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.

Note

If you have previously deployed this solution, see Update the solution for update instructions.

Step 1: 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 25 minutes

1. Sign in to the AWS Management Console and select the button to launch the media2cloud AWS 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.
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
Email	*<Requires Input>*	Email address of the user that will be created in the Amazon Cognito identity pool and subscribed to the Amazon SNS topic. Subscribed users will receive ingestion, analysis, labeling, and error notifications. After launch, two emails will be sent to this address: one with instructions for logging in to the web interface and one confirming the Amazon SNS subscription.
Price Class	`Use Only U.S., Canada and Europe`	A dropdown box with price classes for the edge location from which Amazon CloudFront serves your

Parameter	Default	Description
		requests. Choose `Use Only U.S., Canada and Europe`; `Use U.S., Canada, Europe, Asia and Africa`; or `Use All Edge Locations` . For more information, refer to Choosing the price class.
Amazon OpenSearch Service Cluster Size	`Development and Testing`	A drop-down box with four Amazon OpenSearch Service cluster sizes: `Development and Testing`, `Suitable for Production Workloads` , `Recommended for Production Workloads` , and `Recommended for Large Production Workloads` .
Analysis Feature(s)	`Default`	A drop-down box with nine presets: `Default`, `All`, `Video analysis`, `Audio analysis`, `Image analysis`, `Document analysis`, `Celebrity recognition only` , `Video segment detection only`, and `Speech to text only`. For more information about the presets, refer to Analysis workflow.

Parameter	Default	Description
(Optional) User Defined Amazon S3 Bucket for ingest	*<Requires Input>*	If you have an existing bucket that you would like to store uploaded contents, specify the bucket name. Otherwise, leave it blank to auto create a new bucket.
(Optional) Allow autostart on ingest S3 bucket	NO	A drop-down box to specify if you would like to automatically start workflow when directly upload assets to Amazon S3 ingestion bucket.

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 will create AWS Identity and Access Management (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 25 minutes.

Step 2: Upload a video or image file

After the solution successfully launches, you can start uploading video or image files for processing. The solution sends two emails: one with the subscription confirmation for the Amazon SNS topic to send ingestion, analysis, labeling, and error notifications, and one with instructions for signing into the solution's provided web interface.

1. In the **M2CStatus** email, select **Confirm subscription** to subscribe to the Amazon SNS topic.
2. In the second email, follow the instructions to sign in to the website. You will be prompted to change the password the first time you sign in.

3. Choose **Sign in** on the upper right corner of the page and sign in using your recently created password.
4. Navigate to the **Upload** tab.
5. Drag and drop your files to the **Upload Video** box, or choose the **Browse Files** button to upload a video or image file. Once the files are uploaded, choose **Quick upload**, or select **Next** to **Start Upload**.

   Once the ingestion process is completed, a thumbnail image of the video or image is created. You can hover over the thumbnail image and select **Play now** to view the media file.

Step 3: Create your face collection

The web interface allows you to create your own Amazon Rekognition face collection and index and store faces in the collection to improve the analysis results.

1. In the web interface select **FaceCollection** in the top navigation.
2. Type in the name of the face collection in the blank field and choose **Create New Collection.**
3. In the web interface, hover over a created video or image and choose **Play**.
4. Choose the **Play** button again and then choose **Pause** once you find a face in the content.
5. Move the toggle by **Snapshot Mode** to the right to display a bounding box.
6. Adjust the size of the bounding to fit tightly over the face.
7. Type the name of the person in the **Name** box and select your **Face Collection** from the dropdown menu.
8. Once finished, choose the **Index Face** button**.**
9. Repeat steps 4-8 until you have identified all of the faces.
10. After the faces are indexed, choose **Re-analyze** to analyze the video or image using the newly indexed faces in your face collection so that all unidentified faces are recognized and indexed.

Step 4: Advanced search

Included in the web interface is the ability to search for specific moments across the analyzed content. A user has the ability to put in specific search terms and have timestamped results returned.

1. In the web interface select **Collection** in the top navigation bar.
2. On the collection page, there is a search bar in the top right-hand corner of the page. Deselect any of the attributes that you want excluded from your search and then type a term or phrase in the **Search** box and hit submit.
3. Assets matching the search term will be presented under the **Search Results** section of the page and highlight where there was a match to your search term.
4. Choose the file thumbnail in the search results to be taken to that asset.

Step 5: Customizing AI/ML settings

In this version of Media2Cloud on AWS, users have a lot of flexibility on the AI/ML services that are used. They also have the ability to configure those services for their use cases.

1. In the web interface select **Settings** from the top navigation bar.
2. In the **Amazon Rekognition Settings** section:
   - You can set the minimum confidence level that you want results from.
   - Toggle on or off specific detection types.
   - Select the face collection that you want to use when analyzing assets.
   - When using Amazon Rekognition to detect text on screen, you can select the specific regions of the screen for analysis.
   - If you have created a custom AI/ML model using Amazon Rekognition Custom Labels, you can use that model when analyzing assets.
   - The **Frame Based Analysis** section give the flexibility to switch from the Amazon Rekognition Video API to the Amazon Rekognition Image API. When you toggle the **Frame Based Analysis** button on, you can determine the frequency that frames are analyzed.
3. In the **Amazon Transcribe** settings section:
   - Select the language that you want Amazon Transcribe to create a transcript of the video in. For a complete list of supported languages, refer to Amazon Transcribe Supported Languages.
   - If you have created a **Custom Vocabulary** to improve the accuracy of Amazon Transcribe, you can select that model for the analysis of your assets.
   - If you have created a **Custom Language Model** you can activate that model for the analysis of your assets.
4. In the **Amazon Comprehend** settings section:

- Activate **Entity Detection, Sentiment Analysis,** and **Key phrase Detection**.
- If you have built a **Custom Entity Recognizer** to identify custom entities for your business needs, you can activate that as well.

5. In the **Amazon Textract** settings section, you can activate the service to extract text from documents that you are analyzing.

Step 6: Viewing statistics

Once content has been analyzed, the web interface has a way to show an aggregation of the metadata generated by the AI/ML Services. This helps to answer the question of what the most popular or frequent tags and detections are in the library.

1. In the web interface select **Stats** from the top navigation bar.
2. Pie charts show the overall and categorized statistics of your content collection.

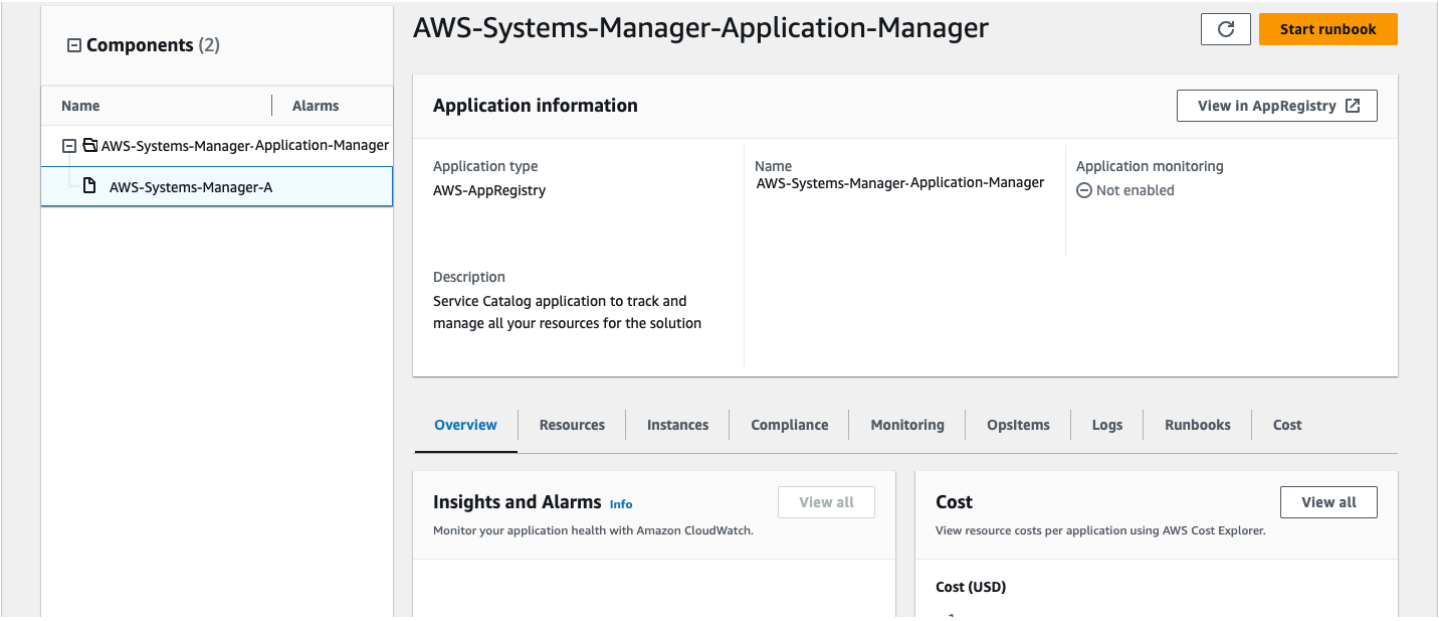
Monitor the solution with Service Catalog AppRegistry

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.



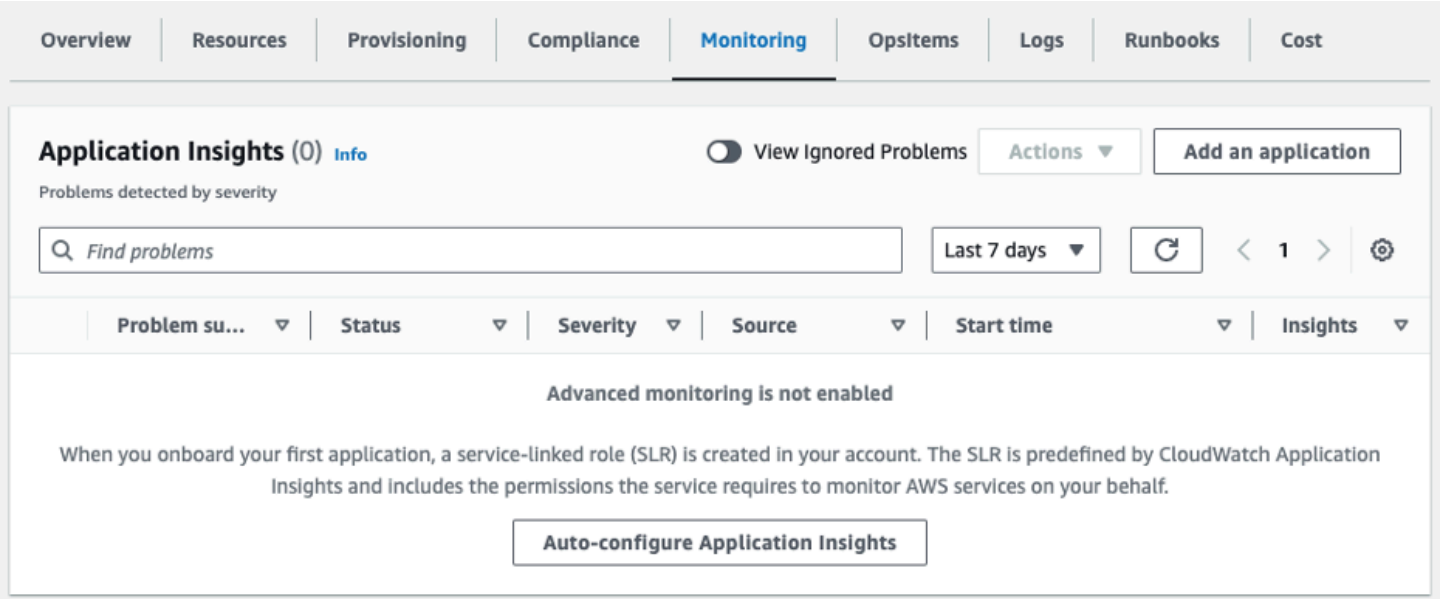
Solution stack in Application Manager

Activate CloudWatch Application Insights

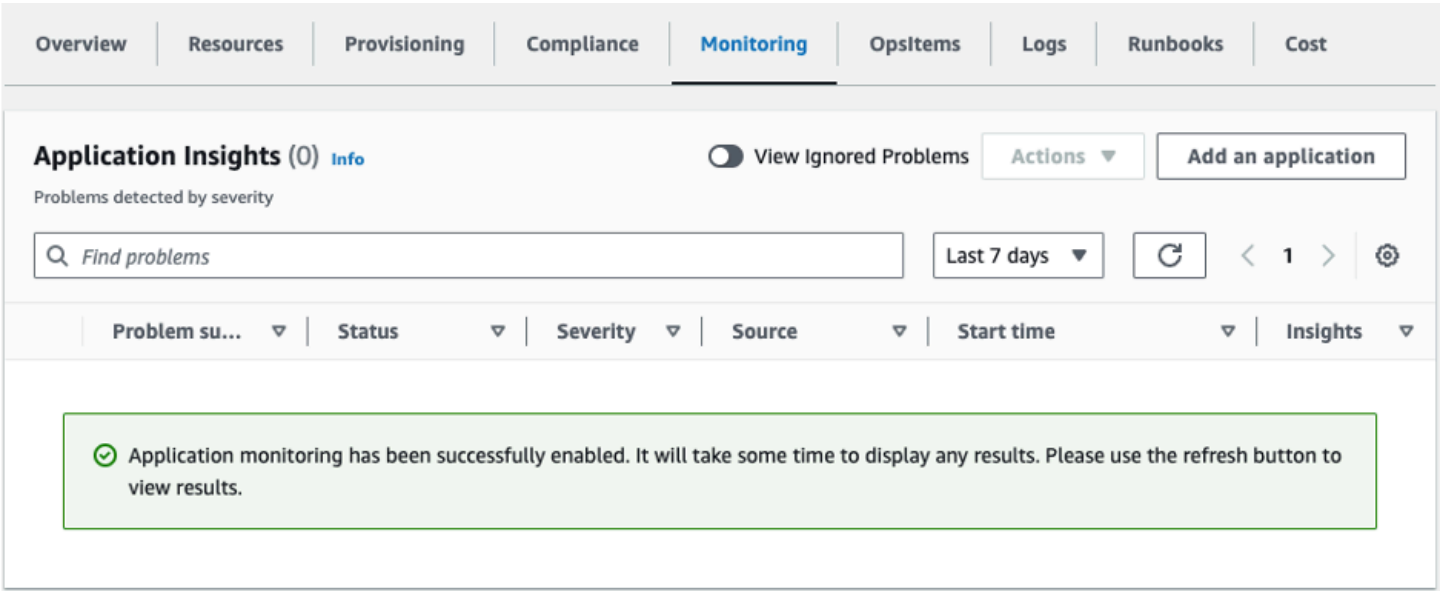
1. Sign in to the Systems Manager console.

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:

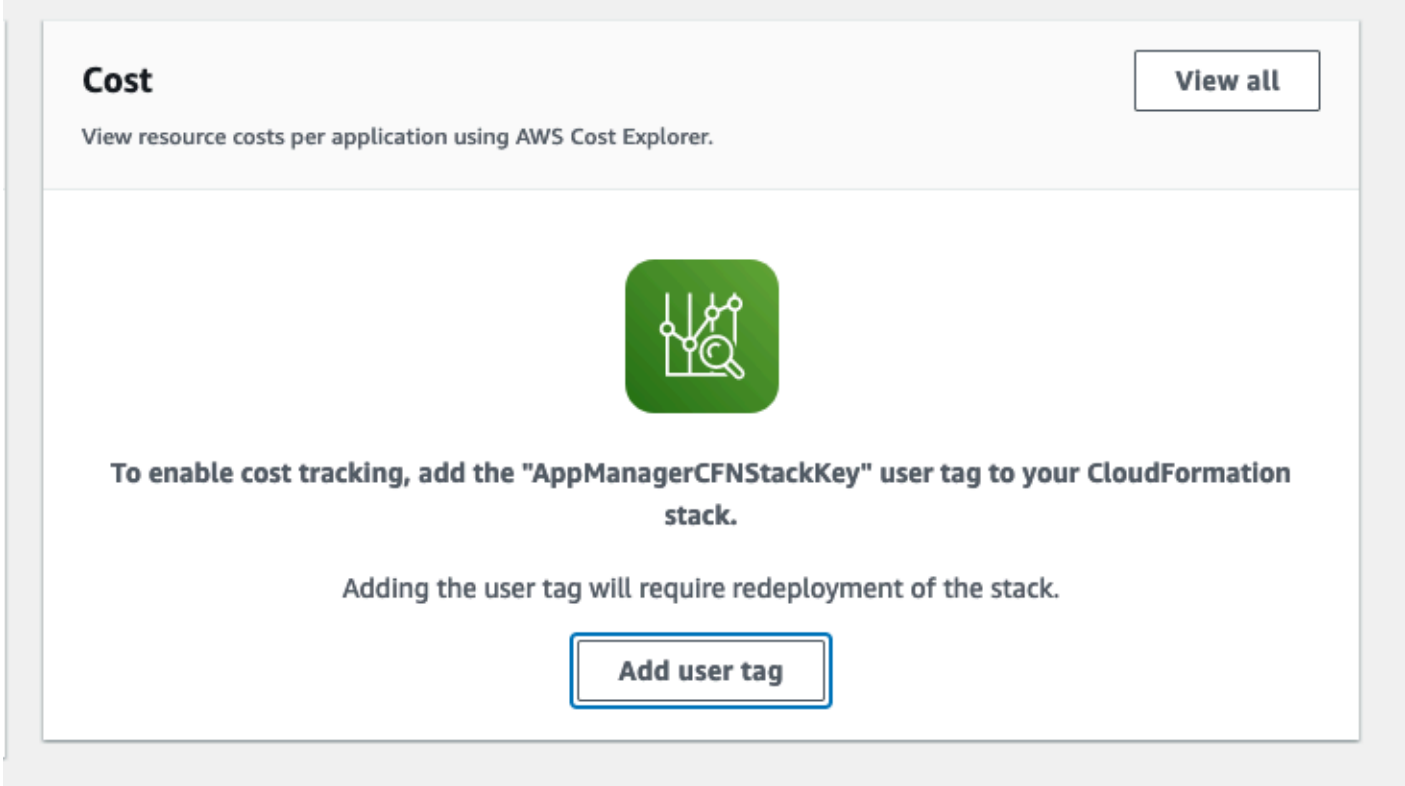


Confirm cost tags associated with the solution

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

Media2Cloud on AWS Version 3 can be deployed side-by-side with your previously deployed Media2Cloud on AWS versions (Version 2 and Version 1). However, this version is incompatible with previous versions due to changes of the Amazon DynamoDB tables and Amazon OpenSearch Service indices used by the solution.

To migrate from your existing Media2Cloud on AWS version (Version 2 or Version 1) to this version, contact your AWS account representative for assistance.

Troubleshooting

If you need help with this solution, contact Support to open a support case for this solution.

Contact 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 **Other Solutions**.
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 question 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 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 the Media2Cloud on AWS solution from the AWS Management Console or by using the AWS Command Line Interface. You must manually delete the S3 buckets, DynamoDB table, 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 Command Line Interface (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>
```

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

Deleting the Amazon S3 buckets

This solution is configured to retain the solution-created Amazon S3 bucket (for deploying in an opt-in Region) if you decide to delete the AWS CloudFormation stack to prevent accidental data loss. 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**.

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

To delete the S3 bucket 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 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:

```
$ 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.

Developer guide

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

Source code

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

Considerations for customizing the solution

Bucket Region

If you specify your own S3 ingestion (and/or proxy) buckets, ensure the bucket(s) are in the same AWS Region as the Region where the solution is deployed to avoid cross-Region data transfer costs.

Bucket CORS settings

If you decide to use your own Amazon S3 ingest (and/or proxy) bucket, the solution updates your bucket's CORS settings to allow the web interface to perform cross origin GET/POST requests. The following example shows the CORS settings:

```
<?xml version="1.0" encoding="UTF-8"?>
<CORSConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<CORSRule>
    <AllowedOrigin>https://<cf-id>.cloudfront.net</AllowedOrigin>
    <AllowedMethod>GET</AllowedMethod>
    <AllowedMethod>PUT</AllowedMethod>
    <AllowedMethod>POST</AllowedMethod>
    <AllowedMethod>HEAD</AllowedMethod>
    <AllowedMethod>DELETE</AllowedMethod>
    <MaxAgeSeconds>3000</MaxAgeSeconds>
    <ExposeHeader>Content-Length</ExposeHeader>
    <ExposeHeader>ETag</ExposeHeader>
    <ExposeHeader>x-amz-meta-uuid</ExposeHeader>
    <ExposeHeader>x-amz-meta-md5</ExposeHeader>
    <AllowedHeader>*</AllowedHeader>
</CORSRule>
</CORSConfiguration>
```

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 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:

- **Solution ID** - The AWS solution identifier
- **Unique ID (UUID)** - Randomly generated, unique identifier for each deployment
- **Timestamp** - Media file upload timestamp
- **Format** - The format of the uploaded media file
- **Size** - The size of the file the solution processes
- **Duration** - The length of the uploaded video file

AWS owns the data gathered though 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 `media2cloud.template` the section called “AWS CloudFormation template” to your local hard drive.
2. Open the AWS CloudFormation template with a text editor.
3. Modify the AWS CloudFormation template mapping section from:

```
Send:
    AnonymizedUsage:
      Data: "Yes"
```

to:

```
Send:
    AnonymizedUsage:
```

```
      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 Deployment section of this guide.

Contributors

- Ken Shek
- Liam Morrison
- Adam Sutherland
- Aramide Kehinde
- Jason Dvorkin
- Noor Hassan
- San Dim Ciin
- Eddie Goynes
- Raul Marquez

Revisions

Date	Change
January 2019	Initial release
March 2019	Modified JSON file descriptions.
November 2019	Updated the analysis workflow engine and added support for ingesting and analyzing images.
January 2021	Updated the list of AWS Partners.
February 2022	Release v3.0.0: New web user interface enhancing the user experience. New analysis features included Amazon Rekognition Custom Labels model, Amazon Rekogniti on Segment Detection, Amazon Transcribe Custom Vocabulary and Custom Language Model, Amazon Comprehend Custom Entity Recognizer. New Amazon OpenSearch Service provided deeper search results that allow customers to find search terms with the timestamps from the archived files. Frame based analysis feature allowing customers to define the time interval (rate) of running the detections on the video content by using Amazon Rekognition Image based API instead of Video based API. For more information, refer to the CHANGELOG.md file in the GitHub repository.
February 2023	Release v3.1.0: Added a Service Catalog AppRegistry resource to register the CloudFormation template and underlying resources as an application in both Service

Date	Change
	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.
April 2023	Release v3.1.1: Added package-lock.json files to all Lambda state machine functions to snapshot the dependency tree used. For more information, refer to the CHANGELOG.md file in the GitHub repository.
April 2023	Release v3.1.2: Mitigated impact caused by new default settings for S3 Object Ownership (ACLs disabled) for all new S3 buckets. Updated object ownership configuration on the S3 buckets. Updated security patching. For more information, refer to the CHANGELOG .md file in the GitHub repository.
August 2023	Release v3.1.3: Fixed an issue where media analysis result is not visible in the web application. For more information, refer to the CHANGELOG.md file in the GitHub repository.

Date	Change
September 2023	Release v3.1.4: Updated Lambda nodes to Node.js 16. Added unit tests for document ingestion, information about stack update support, and default values to stack parameters. Updated parameter names for consistency. Added fix to allow PDF conversio n to PNG files for previously unsupported font types and other bug fixes. For more informati on, refer to the CHANGELOG.md file in the GitHub repository.
November 2023	Release v3.1.5: 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.