Guide to Create a MediaConvert, Transcribe, and CloudFront Stack

Hosting Infrastructure · https://www.mooplugins.com/docs/video-lesson-activity/hosting-infrastructure/guide-to-create-a-mediaconvert-transcribe-and-cloudfront-stack/

This guide walks through setting up a stack for processing media files via AWS MediaConvert, triggering transcription via AWS Transcribe, storing results in S3, and delivering output securely via CloudFront.

It focuses on:

- Required AWS services
- How those services connect
- Roles and permissions involved
- Event wiring and data flow

This guide does **not** include code or step-by-step CloudFormation instructions. It describes **what must exist and how it fits together**, allowing AWS administrators to implement the stack using their preferred tooling.

Who this documentation is for

This guide is intended for:

- AWS Administrators
- DevOps or Infrastructure teams
- Technical partners deploying Video Lesson Activity in a self-hosted AWS environment

It assumes:

- An active AWS account with billing enabled
- Permission to create IAM roles, policies, and services

- Familiarity with AWS services such as S3, Lambda, and IAM

High-level architecture

The stack processes uploaded videos, generates subtitles, and delivers outputs securely using an event-driven AWS architecture.

1. Overview of Stack Components

Your stack will include:

1. **S3 Buckets**
   - **Input Bucket** – Where raw media files are uploaded.
   - **Output Bucket** – Where converted videos and transcripts are stored.

2. **IAM Roles and Policies**
   - Roles for Lambda functions to access S3, MediaConvert, Transcribe, DynamoDB, CloudFront.
   - Role for MediaConvert service to access S3.
   - Managed policy for a user to access S3 and DynamoDB.

3. **Lambda Functions**
   - **MediaConvert Trigger Lambda** – Triggered by S3 uploads to start MediaConvert jobs.
   - **Transcribe Trigger Lambda** – Triggered by SNS events or CloudWatch Events to start transcription.

4. **SNS Topic**
   - Sends notifications from Transcribe jobs to the Lambda function.

5. **CloudWatch / EventBridge Rules**
   - Monitor MediaConvert job state changes.

- Monitor Transcribe job completions.

6. **CloudFront Distribution**
   - Serves the output bucket securely to clients.
   - Uses Origin Access Control (OAC) to prevent direct S3 access.

7. **DynamoDB Table**
   - Tracks MediaConvert job metadata.
   - Supports a secondary index for content hash lookups.

8. **IAM User + Access Keys**
   - For external systems (e.g., Moodle) to access input/output buckets.

2. Parameters Section

Parameters allow your stack to be flexible:

- **Resource Bucket**: The S3 bucket containing your Lambda function ZIP files.
- **Lambda MediaConvert Trigger ArchiveKey**: The S3 object key for MediaConvert Lambda.
- **Lambda Transcribe Trigger ArchiveKey**: The S3 object key for Transcribe Lambda.

3. IAM Roles and Policies

Roles are assigned to services (like Lambda or MediaConvert) so they can perform actions. Key roles:

1) Lambda MediaConvert Trigger Role

Permissions to:

- Read/write S3 buckets (input/output)
- Start MediaConvert jobs
- Trigger Transcribe jobs

- Write to DynamoDB
- Publish to SNS
- Create CloudFront invalidations
- Attached policy: LambdaMediaConvertTriggerPolicy

2) MediaConvert Role

- Assigned to MediaConvert jobs.
- Permissions: read/write input/output S3 buckets.

3) Managed Policy for Users

- Gives an external user access to S3 and DynamoDB.
- Can create IAM user access keys.

Always scope permissions narrowly where possible, but some actions like mediaconvert:CreateJob or transcribe:StartTranscriptionJob may need "*" resource.

4. Lambda Functions

MediaConvert Trigger Lambda

- Triggered when a new object is uploaded to the input S3 bucket.
- Starts MediaConvert jobs.
- Writes job metadata to DynamoDB.

Transcribe Trigger Lambda

- Triggered by SNS notifications (from Transcribe) or EventBridge events.
- Starts transcription jobs and stores results in S3.

Key Configurations:

- Assign proper IAM roles.

- Set environment variables for bucket names and role ARNs.
- Timeout should allow long-running media jobs (~10 min or more).
- Runtime: Python 3.x.

5. S3 Buckets

Input Bucket

- Trigger Lambda on ObjectCreated events.
- Set proper bucket policy so Lambda can read/write objects.

Output Bucket

- Enable CORS if needed.
- Attach a bucket policy allowing CloudFront to fetch objects securely via OAC.

6. CloudFront Distribution

- Create an Origin Access Control (OAC) to secure S3 content.
- Set Default Cache Behavior to:
  - Redirect HTTP → HTTPS
  - Allow GET, HEAD, OPTIONS
  - Compress content
- Connect the output bucket via the OAC.

7. DynamoDB Table

- Table stores MediaConvert job info.
- Key attributes: job_id (primary key), contenthash (secondary index for lookup).
- Set billing mode to pay-per-request for flexibility.
- Allow Lambda to read/write/query the table.

8. SNS Topic and Subscriptions

- **SNS Topic** for Transcribe job notifications.
- Lambda subscription ensures function is triggered when jobs complete.
- Lambda permissions allow invocation by SNS.

9. EventBridge / CloudWatch Rules

- **MediaConvert**: Trigger Lambda on job state changes: PROGRESSING, COMPLETE, ERROR.
- **Transcribe**: Trigger Lambda when transcription completes (COMPLETED or FAILED).

10. IAM Users and Access Keys

- Create IAM user for external systems (e.g., Moodle).
- Attach the managed policy.
- Generate access keys for programmatic access.

11. Outputs

Your stack should output key ARNs, IDs, and access details for external use:

- S3 Buckets
- DynamoDB Table
- SNS Topic ARN
- Lambda function ARNs
- CloudFront distribution domain name and ARN
- IAM access keys

This allows downstream systems to reference these resources easily.

12. Permissions checklist

Component	Required permissions
Media Processing Lambda	S3 read/write, MediaConvert, Transcribe, DynamoDB, SNS, EventBridge
Subtitle Processing Lambda	S3 read/write, Transcribe, translation services

Component	Required permissions
MediaConvert role	S3 Read/Write for input/output
IAM user	Scoped S3 and metadata access
SNS	Lambda invoke
EventBridge	Lambda invoke
CloudFront	S3 GetObject via OAC

Tip: Make sure roles have trust relationships allowing the correct service to assume the role (Lambda → Lambda role, MediaConvert → MediaConvert role).

13. Connecting Resources

1. Input S3 bucket → triggers MediaConvert Lambda.
2. MediaConvert Lambda → starts MediaConvert jobs → stores metadata in DynamoDB.
3. MediaConvert jobs → produce output in Output S3 bucket.
4. Output S3 bucket → served via CloudFront (OAC secures bucket).
5. Transcribe jobs → SNS topic → triggers Transcribe Lambda → stores subtitles in Output bucket.
6. External system → uses IAM User credentials to interact with buckets and DynamoDB.
7. Lambda Environment Variables → Set required variables

14. Summary

By following this guide:

- You define parameters for flexibility.
- Set up resources: S3, Lambda, IAM, SNS, EventBridge, MediaConvert, CloudFront, DynamoDB.
- Assign permissions carefully.
- Connect triggers: S3 → Lambda → MediaConvert → DynamoDB → S3 → CloudFront.
- Outputs allow external access and integration.