

OPT Volunteering Progress Report

Project Title: Serverless application that uses the Amazon Polly service to convert text to speech

Week: 02/11/2025 – 02/24/2025

Report Date: 02/25/2025

INTRODUCTION

Project Overview:

This section of the report involved testing of the Lambda Functions Workflow and implementation of a Lambda function (PostReader_GetPost) to retrieve post details from DynamoDB.

Work Progress:

1. Tested the Lambda Functions Workflow

- Triggered the New Post Lambda function manually, which:
 - Stored data in DynamoDB.
 - Sent a message to the SNS topic.
- Verified SNS functionality, which:
 - Triggered the Convert To Audio Lambda function.
 - Used Amazon Polly to convert text into an audio file.
 - Stored the generated MP3 file in an S3 bucket.
- Checked DynamoDB Entries:
 - Verified that two test entries existed (indicating successful execution).

- Confirmed the second execution updated the post with the S3 audio file URL.
- Monitored the Convert To Audio Lambda function execution:
 - Used AWS Lambda monitoring to ensure successful function invocation.
 - Checked CloudWatch logs in case of errors and verified that the correct S3 bucket name was used.
- Validated S3 Storage:
 - Navigated to the S3 bucket (audioposts-<number>).
 - Located and downloaded the MP3 file.
 - Played the file to confirm successful text-to-speech conversion using Polly's Joanna voice.

2. Implemented the Get Post Lambda Function

Created a Lambda function (PostReader_GetPost) to retrieve post details from DynamoDB.

Configured the function:

- Used Python 3.12 runtime.
- Assigned an existing IAM execution role (Lab-Lambda-Role).

Developed functionality:

- Fetched a post's details using its postId from DynamoDB.
- If postId is *, retrieved all posts (useful for small datasets).
- Returned stored metadata, including the S3 URL for the generated MP3 file.

Set up environment variables:

- Added DB_TABLE_NAME as an environment variable with the value posts.

Tested the function:

- Created and executed a test event (AllPosts) with {"postId": "*"}
- Verified that all posts, including their associated audio file links, were correctly retrieved from DynamoDB.

Completion Summary:

Successfully tested the end-to-end workflow for storing, processing, and retrieving text-to-speech converted posts. The New Post Lambda function, SNS

integration, Convert To Audio function, and S3 storage were validated. The Get Post function was also implemented and tested, enabling efficient retrieval of stored posts.

Conclusion

I have successfully done the following:

Created an Amazon DynamoDB table to store data

Created an Amazon API Gateway RESTful API

Created AWS Lambda functions triggered by API Gateway

Connected AWS Lambda functions with Amazon Simple Notification Service (SNS)

Used Amazon Polly to synthesize speech in a variety of languages and voices

In this project, I have created an application that converts text into speech in dozens of languages and voices. Although the application converts blog posts into speech, it can be used for many other purposes, such as converting text on websites or adding speech functionality to web applications.

The application is completely serverless. There are no servers to maintain or patch. By default, the application is highly available because AWS Lambda, Amazon API Gateway, Amazon S3, and Amazon DynamoDB use multiple Available Zones.