**AWS-Photo-Album---Using-Transcribe-Rekognition**

**Introduction**

This is a photo album web application that is deployed on AWS. The application searches through photos using natural language through both text and voice. This application has been developed as a part of the project component for the course- Cloud Computing (CS-GY 9223, New York University)

Some of the important AWS services that are used are Lex, ElasticSearch, Transcribe, Cloud Formation and Rekognition. These services have helped build and create an intelligent search layer to query the photos of people, objects, actions, landmarks and more. We use Amazon Transcribe on the frontend to transcribe speech to text, then use the transcribed text to perform the search.

**Description**

It is a simple frontend application that allows users to:

i. Make search requests.

ii. Display the results (photos) resulting from the query.

iii. Upload new photos.

The frontend user is also given the choice to use voice rather than text to perform the search.

-Using ElasticSearch service, a domain called ‘photos’ is created. Whenever a photo gets uploaded to the bucket, it triggers the Lambda function (LF1) to index it.

-We also detect labels in the image, using Rekognition (“detectLabels” method).

-We store a JSON object in an ElasticSearch index (“photos”) that references the S3 object and an array of string labels is, one for each label detected by Rekognition.

-A Lambda function (LF2) called “search-photos” is used to interact with Amazon Lex and detect search keywords.

-An Amazon Lex bot is created to handle search queries. We create one intent named “SearchIntent” and add training utterances to the intent.

-In brief the user can visit the photo album application, search photos using natural language via voice and text, see relevant results based on what user searched and can upload new photos and see them appear in the search results.

**AWS Services used:**

Lex, ElasticSearch, API Gateway, Transcribe, Cloud Formation, Codepipeline and Rekognition.