# FIT5225 - Cloud Computing and Security

## Assignment 2 – Team Report

Based on

# Cloud-Based Image Storage and Retrieval System with Auto-Generated Tags

Akash Balakrishnan - 32192886

Ashiklal Memanaparambil - 32935064

Jason Pius Dsouza - 32346492

Sudalaimani Balaji - 32912420

Clayton A2 Group 27

## Introduction

This report's goal is to give a thorough overview of the creation of an online system that is cloud-based and enables automatic tag-based image storage and retrieval. To develop a serverless application, the system makes use of several AWS services, including S3, Lambda, API Gateway, and DynamoDB. In the report, the project's goals, architecture, and implementation specifics are covered, with an emphasis on the features and functionalities realised.

## Objectives

The following are the project's primary goals:

- Construct and put into operation a cloud-based system for archiving and retrieving photographs with auto-generated tags.
- Create a system for automatic tagging that uses object detection to find items in the submitted pictures.
- Make it possible for users to upload pictures to a specific S3 bucket, which will start an automatic tagging Lambda function.
- For effective query handling, store the identified items and picture URLs in a DynamoDB database.
- Create an API endpoint using API Gateway to enable tag-based image searches.
- Make that the system is safe, scalable, and uses federated authentication for user access.

## Architecture Overview

In order to provide the needed functionality, a number of AWS services and components are used in the system design. These are the main parts:

- **AWS S3:** The infrastructure for storing the uploaded photos is provided by Amazon S3.
- **AWS Lambda:** Utilising the Yolo object detection function, AWS Lambda executes the tagging operation.
- **DynamoDB:** This database stores the image metadata, together with the items that were found and their corresponding URLs.
- **API Gateway:** An API gateway enables communication between a client and a backend system via API endpoints.
- **Cognito:** AWS offers a fully managed solution that aids in adding user authentication and authorisation to applications. In order to manage user sign-up and sign-in, it is largely utilised in cloud computing environments.

## Architecture Diagram

The following architecture diagram depicts the high-level design of our cloud-based image tagging system using AWS Architecture Icons:

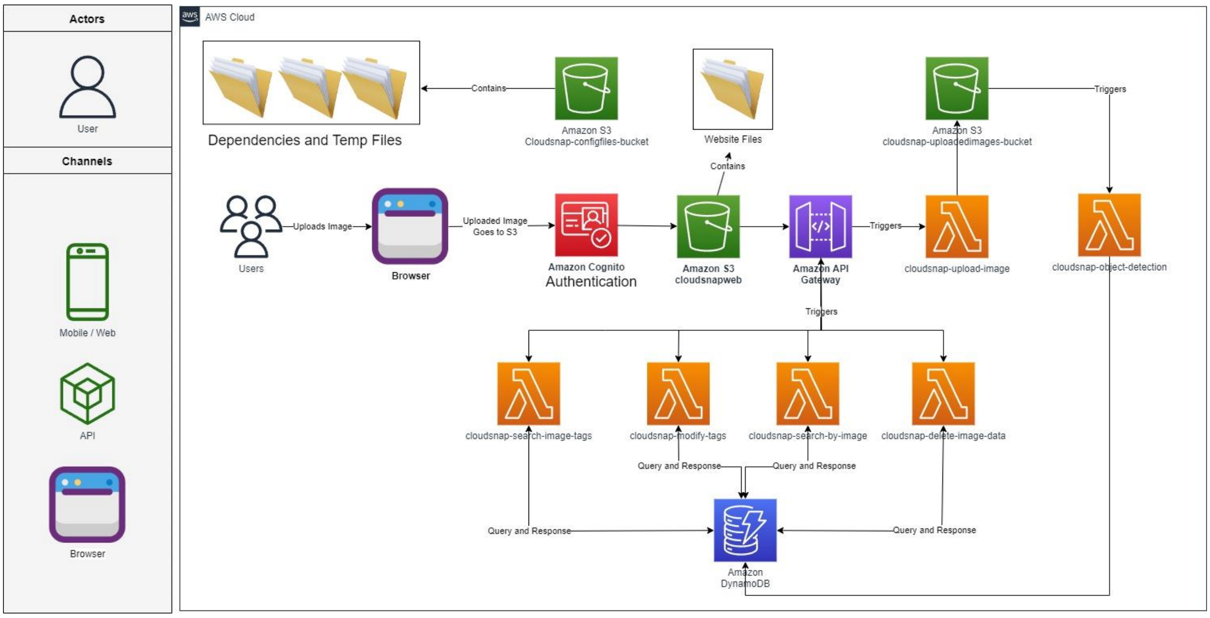

*Figure 1 Architecture Diagram*

## Team Members and Contributions

The table below provides an overview of each team member's contributions to the project:

| Student Name | Student ID | Elements Contributed |
|---|---|---|
| **Akash Balakrishnan** | [32192886] | [Lambda Functions – cloudsnap-object-detection, cloudsnap-delete-image-data] |
| **Ashiklal Memanaparambil** | [32935064] | [Lambda Functions – cloudsnap-search-image-tags, cloudsnap-modify-tags] |
| **Ashiklal Memanaparambil & Akash Balakrishnan** | [32192886, 32935064] | [Lambda Functions – cloudsnap-upload-image, cloudsnap-search-by-image] |
| **Jason Pius Dsouza** | [32346492] | [Cognito user authentication & Authorization for login and Sign-up] |
| **Sudalaimani Balaji** | [32912420] | [Report, API Gateway Setup] |

## User Guide

To test our application, please follow the steps outlined below:

- **Step 1:** Visit the link to access the website's login page.
- **Step 2:** Choose either "login" or "sign up" based on your account status.
- **Step 3:** Sign up by providing a valid email, given name, family name, and password. Verify your email using the sent code.

- **Step 4:** After successful login, you'll see five options: "Upload Image," "Get Image by Tags," "Update Image Tags," "Find Query with Image," and "Delete Image."
- **Step 5:** Select "Upload Image" to upload an image that will be tagged using the YOLO Algorithm for object detection.
- **Step 6:** Choose "Get Image by Tags" to search for images based on user-defined tags. Enter the tags and receive a list of matching images.
- **Step 7:** Use "Update Image Tags" to modify tags for an uploaded image. Provide the image URL, choose add/remove action, and specify the tags.
- **Step 8:** Opt for "Find Query with Image" to find images similar to a reference image. Upload the reference image and view the matching image URLs.
- **Step 9:** Select "Delete Image" to remove an image from the database. Enter the image URL and confirm the deletion.
- **Step 10:** Click "SIGN OUT" to log out of the website and return to the login page.

Please note that the application will remain active for two weeks after the submission deadline to allow testing and evaluation.

## Source Code Repository

The source code for our project is available in a private repository hosted on [https://git.infotech.monash.edu/fit5225-group27/cloudsnap]. Each team member has committed their code to ensure proper collaboration and accountability. The private repository has been shared with the teaching team to provide evidence of contributions and avoid plagiarism.

## Conclusion

Using AWS services, our team has successfully created a cloud-based image tagging system. Users can upload photographs to S3, the system uses Lambda to automatically identify things in the images, and DynamoDB is used to store the labelled data. Users can utilise the API Gateway to search for photos based on tags. We think that our solution provides a scalable and effective architecture for image storage and retrieval in addition to satisfying the project criteria.

### References

AWS Architecture Icons: https://aws.amazon.com/architecture/icons/

**NOTE:** I acknowledge the use of ChatGPT (https://chat.openai.com/) for generating foundational concepts, ideas, and structures throughout the development of this project. ChatGPT aided in creating lambda functions, configuring S3 buckets, setting up API Gateway, managing DynamoDB, and designing the frontend website. The outputs generated from this AI system were adapted, modified, and built upon to realize the final form of the project.