

# Assignment 3

FIT5225 2024 SM1

## *PixTag*: An AWS-powered Serverless Image Storage System with Advanced Tagging Capabilities

### 1 Synopsis

This assignment is based on auto-generated code that enables clients to upload images automatically to an S3 bucket. Users can also modify and delete images in them. To achieve this, we will include the specification of the system and retrieve images from a serverless application. In the application, the application will later on, users can tag the objects present in the image (or thumbnails) that

### 2 Group

The majority of students to expect. Students will be described above. There is no desired distribution. Understanding is crucial to the success of the project. If they encounter “issues”, they should report them. The due date is 11/11/2024. If a team needs an extension, they should report this to the professor. If a team fails to report this, they will be penalized heavily, regardless of the



and retrieve images from a serverless application. In the application, the application will later on, users can tag the objects present in the image (or thumbnails) that

Therefore, to prepare as a group project. Implementing the system requires collective involvement. For agreeing on work, successful integration is required. Teams encounter any issues should be provided to resolve them. The professor will provide a 'solution' for granting an extension and handle their tasks. The professor will provide involvement in the project. The professor will be penalized heavily,

### 3 Assignment Description

Teams should develop an AWS-based solution that leverages services such as S3, Lambda, API Gateway, and database services (e.g., DynamoDB) to build a system for automated object detection tagging and query handling. The teams should produce a solution that enables end-users to upload their images into an S3 bucket. Upon uploading an image to a designated S3 bucket, a lambda function is automatically triggered, which uses the Yolo object detection feature to identify the objects in the image and stores the list of detected objects along with the image's S3 URL in a database. Furthermore, the end-user should be able to submit queries to an API endpoint using API Gateway to search for tagged images (more details to come). Table 1 provides a glossary of terms used in the assignment description.

Table 1: Glossary of terms

Term	Meaning
team	A group of 4 students who are doing the project
queries	You will implement multiple APIs to query different information including a request message by the end user to find list of images with specific tags, find images with similar tags of an image, update tags of an image, and delete record of an image.
tags	list of objects detected in an image, e.g., person, car, and cat.
federated authentication	Federated authentication is a Single sign-on (SSO) mechanism that allows you to use authentication such as services
	console in with day.

### 3.1 Authen

Security is one publicly exposed access and malicious centralized app

To leverage t  
Then, you need  
AWS services to  
Cognito service  
to access the us  
Amplify JavaSc  
However, if you  
you can check fo

Your applica

your application is  
gainst unauthorized  
forward, secure, and  
unauthorized access.  
res user credentials.  
cation and/or other  
icate with the AWS  
AWS JavaScript SDK  
ase note that AWS  
might be an option.  
further info on this

- Detect whether a user is authenticated or not. If the user has not signed in, access to all pages/endpoints except the sign-up service needs to be blocked, and the user should be redirected to the “sign-up.html” page to register a new account. For each new account, you need to record the user’s *email address*, *first name*, *last name*, and *password*. Cognito will take care of sending an email to new users, asking them to verify their email address and change their temporary password.
- Your application should include a login page that allows users to sign in. After successful authentication, users should be able to upload images, submit queries, view query results, and sign out of the application. All of these services must be protected against unauthorized access. You can implement login and sign-up web pages using either the Hosted UI feature of Cognito or your own version that calls Cognito APIs.
- Uploading files to an S3 bucket, invoking Lambda functions to execute the business logic of your

application, and accessing the database for data storage and retrieval all require fine-grained access control permissions that you need to set up via IAM roles and appropriate policies. It is important to note that IAM roles in AWS Academy have several limitations. Therefore, you should carefully consider how to perform authentication and authorization in your system while taking these limitations into account.

As an **optional** feature, you can add federated authentication using AWS Cognito Identity Pools to your application and earn bonus marks (up to 5 points shall be awarded if you add federated authentication to your project). For this purpose, you need to create a Facebook or Google app that serves as an external identity provider and authenticates users on behalf of your application, then forwards authentication tokens to your application. Note that having external authentication providers in your project is **not** mandatory. Since federated authentication might be challenging and maybe impossible with your AWS academy account. I **strongly** recommend that you finish the requirements of the assignment first, and then, if you have

### 3.2 Image

Your solution should be able to upload an image to an S3 bucket can be done directly or it can be done through an API (e.g., REST APIs) or it can be done through a function. Whenever an image is uploaded to the S3 bucket, a function should make a call to the image processing service to process the original image to a smaller file size. You can use the image processing service to grant the required permissions to the function.

**Building Image Thumbnail**  
You can use OpenCV library to build a thumbnail for an image. You can use the Amazon resource to learn more about OpenCV.

Another Lambda function can be used to process the original image and save the list of objects in the image in an AWS database. The objects were given in your assignment. You can use Flask-related code to process the image from S3 buckets, and store the Yolo and other objects in the database.

You may also use the image processing service. Please note that the image processing service is a paid service. However, in this assignment, you can use the image processing service for free.

### 3.3 Queries

Your solution should provide APIs which allow following queries.

1. **Find images based on the tags:** The user can send a text-based query to request URLs of images that contain specific tags with a minimum repetition number for each tag (e.g., “person, 1”, “person, 2, car, 1”). You are expected to create an API Gateway with a RESTful API that allows users to submit their requests, such as GET or POST requests.

Your application might send a list of tags via specific GET parameters in the requested URL, for example:

```
https://jyufwbv84.execute-api.us-east-1.amazonaws.com/dev/search?tag1=cat&tag2=car
```



or it can be a POST request with a JSON object including a list of tags and their counts. A sample JSON object for a query request is given below:

```
{
  "tags": [
    "person",
    "desk",
    ...
    "cat"
  ]
}
```

A response in **all** the requested tags in the

```
{
  "link"
}
```

Your query is a list of tags and finds s3-url of the images matching the query, i.e., logical **AND** operation,

Please note

For example

In your UI, the response should be thumbnails found as the results. The response should be `"tags": ["person"]}` or sending another

query to get

2. **Find image thumbnails:** The user sends a query to the system and the system returns a list of image thumbnails into the

3. **Find image thumbnails:** The user sends a query to the system and the system returns a list of image thumbnails into the part of an API call.

The list of all objects (tags) in the sent image is discovered and then all the images in *PixTag* storage containing those set of tags are found. Finally, as a response, the list of s3-URLs of the matching images thumbnails (similar to to the previous section) are returned to the user. You should make sure that the image uploaded for the query purpose is not added to the database or stored in s3.

4. **Manual addition or removal of tags with bulk tagging:** Your solution should also provide an API that allow end-user to add or remove tags of images. You are expected to create a POST API which allow users to submit their requests.

A sample JSON message sent to add/remove tags is:

```
{
  "url": [
    "https://pixtag.s3.amazonaws.com/image1-thumb.png",
  ]
}
```

```

        "https://pixtag.s3.amazonaws.com/image60-thumb.png",
        "https://pixtag.s3.amazonaws.com/image23-thumb.png",
    ]
    "type": 1, /* 1 for add and 0 for remove */
    "tags": [
        "person",
        "alex"
    ]
}

```

“type” can be set to 1 or 0 for adding or removing a tag, respectively. The above request adds “person” and “alex” tags to the tag list of the images in the list of URLs. If “type” is set to 0, the tags are removed from the tag list of the image. If a tag is not included in the list of tags requested for deletion, it remains in the tag list of the image.

5. **Delete in**  
system should  
database.

to an API and the  
ent entries from the

### 3.4 Tag-based

Users can receive  
users can receive  
You can use AWS

tags. For example,  
ated to the system.

### 3.5 User Interface

You can design  
includes the following  
easier and more  
UI for some part  
Python) to handle  
manually copy c  
with the later o

be of any form that  
makes your system  
application or have  
n write scripts (e.g.,  
t you might need to  
you choose to work

## 4 Final

You should write an **individual** and **team** report on the developed application. You should use a highly-legible font type such as: *Arial*, *Helvetica*, and *Times New Roman* with font size 12 points.

### 4.1 Team Report - 750 words

Please prepare a team report based on the following guidelines:

- Include an architecture diagram in the team report. For the architecture diagram, you must use AWS Architecture Icons (more info can be found here: <https://aws.amazon.com/architecture/icons/>).
- Include a table to describe the role of each team member in your team report, You can provide a three-column table in your report that shows student name and id, percentage of contribution and elements of the project the member contributed (maximum of 100 words per member).