CPD Coursework Report

BSc(Hons) Computing

“I declare that all work submitted for this coursework is the work of Faisal Khan alone unless stated otherwise.”

Faisal Khan

Student ID: S1828698

Table of Contents

[Description of the problem 2](#_Toc37623598)

[The approach to solve it 2](#_Toc37623599)

[Implementation in another cloud platform 3](#_Toc37623600)

[Azure Blob Storage 3](#_Toc37623601)

[Azure Queue Storage 3](#_Toc37623602)

[Azure Functions 3](#_Toc37623603)

[Cognitive Services 3](#_Toc37623604)

[Cosmos DB 4](#_Toc37623605)

[Securing the application 5](#_Toc37623606)

[AWS S3 5](#_Toc37623607)

[SQS 5](#_Toc37623608)

[Lambda 5](#_Toc37623609)

[Rekognition 6](#_Toc37623610)

[DynamoDB 6](#_Toc37623611)

[Application Testing 7](#_Toc37623612)

[Console UI 7](#_Toc37623613)

[Upload File to S3 8](#_Toc37623614)

[Download File from S3 12](#_Toc37623615)

[List all files in S3 bucket (Extended Functionality) 14](#_Toc37623616)

[List all buckets in S3 (Extended Functionality) 14](#_Toc37623617)

[Quit 15](#_Toc37623618)

[AWS – Image Upload Notification 16](#_Toc37623619)

[Receive and process message from SQS 17](#_Toc37623620)

[Pass Image Details to Rekognition 18](#_Toc37623621)

[Receive and process image labels from Rekognition 18](#_Toc37623622)

[Store the labels in a DynamoDB table 19](#_Toc37623623)

[References 20](#_Toc37623624)

## Description of the problem

The problem that was issued in this coursework was to enable a user to upload images to a cloud-based system which is then processed through an artificial intelligence service to output a number of labels describing what the picture is or contains.

## The approach to solve it

The solution to this problem is to implement an image recognition application with AWS(Amazon Web Services) using their trusted AI services such as AWS Rekognition. AWS Rekognition is an artificial intelligence service which can identify people, objects, text and activities in images and videos. Through the use of machine learning, Rekognition also has the ability to detect, analyse and compare faces.

The first stage in the solution was to create a python application which was capable of uploading images to AWS S3 (Simple Storage Service) and downloading them back onto the local machine. AWS S3 is a cloud storage service that offers many use cases for a range of different clients. It can store data securely without effecting performance as well as possessing scalability and availability making S3 an ideal storage solution for applications.

Stage two consisted of sending a notification to SQS(Simple Queue Service) once an image had been uploaded to the S3 bucket. SQS is a message queuing service which transfers any amount of data, at any throughput, without losing messages, thus ensuring that messages are delivered consistently. The ability to use server-side encryption (SSE) for messages guarantee that sensitive data is kept secure whilst in a queue. Furthermore, SQS can adapt and scale dynamically based on demand.

The third stage involved triggering the Lambda function via a message in the SQS queue. AWS Lambda is a service which allows you to compile code without the need to administer servers. Lambda also has the capability to automatically scale applications to the capacity required by executing code in response to each trigger.

The final stage involved creating a Lambda function which would receive and process the message from SQS parsing it to extract relevant information such as image name. Once complete, the extracted information should be sent to AWS Rekognition in order to obtain image labels. The labels gathered from Rekognition must be processed to extract image label data with only a maximum of five labels per image. The label data should then be stored in a DynamoDB table. The database table should contain a single primary key as image name.

## Implementation in another cloud platform

This application could also be implemented using Microsoft’s Azure cloud platform as both AWS and Azure offer similar services. Comparable services have been identified and discussed below.

### Azure Blob Storage

Comparing storage solutions, S3 is comparable to Azure’s Blob Storage. Blob is an object storage service which is suitable for cloud-based applications along with storing large volumes of unstructured data, such as text or binary. Blob is intended for many use cases such as storing files, streaming audio and video and storing data for analysis by other Azure services.

### Azure Queue Storage

AWS SQS is equivalent to Azure’s Queue Storage service. This service provides regulated communication between decoupled application components leading to independent scalability, similar to SQS. Asynchronous message queuing is utilized between components regardless of which device they are using.

### Azure Functions

Azure Functions can be compared to AWS Lambda. Both services execute code in response to triggers or specific events without requiring you to manage or control servers. Similarly, in Lambda, Functions can automatically scale depending on the workload. Both also offer a diverse selection of programming languages for consumers.

### Cognitive Services

Microsoft has their own take on AWS Rekognition known as Cognitive Services. Likewise, they offer similar solutions such as Computer Vision, Face, and Emotions. Computer Vision is used to extract and analyse content from images. Face detects, identifies and analyses faces in images and Emotions which recognises emotions such as being happy or angry. There are many other varieties of services including: Decision, Language, Speech and Web search offered by Azure.

### Cosmos DB

Contrasting database solutions, AWS DynamoDB is comparable to Azure’s Cosmos DB. Cosmos is a multi-model database that natively supports a range of data models such as: key-value, documents, graphs and columnar. Low latency and high availability ensure a reliable and immediate database solution.

## Securing the application

The application can be secured using many features available through AWS, some of which are considered below.

### AWS S3

In order to prevent security flaws in your application, ensure S3 buckets use the appropriate policies and are not publicly accessible. Removing policies that allow any user to perform any action on the bucket will prevent any unwanted activity. Inspect the access control list (ACL) which permits any user to read and write data to the bucket.

Another security principle which can be incorporated into the application is to use IAM (Identity and Access Management) roles. IAM roles should be used in order to manage temporary credentials for applications or services as an alternative to hard coding your credentials directly in the application. The IAM role provides temporary permissions that applications can utilize when making calls to other resources.

### SQS

Implementing least privilege access is one of many security features of SQS which can prevent malicious intent. When permissions are granted, you can decide who receives them, which queues the permissions are for and the specific API actions that you want to allow. SQS applies the producer-consumer model which entails three types of user account access: Administrators – who control the queue policies, Producers – send messages to queues and Consumers – receiving and deleting messages.

An additional security feature would be to implement server-side encryption (SSE). SQS already incorporates a level of encryption when a message is stored and decryption when the message is accessed. SSE utilises the Key Management Service in AWS to create secure cryptographic keys.

### Lambda

AWS Lambda contains many security elements to assist in keeping your application secure. AWS recommends using SSL/TLS to communicate with other resources and also utilise encryption solutions such as the Key Management Service in order to maintain data protection.

Another factor which AWS recommends using within Lambda is to apply multi-factor authentication (MFA) with each account to avert malicious users who may try to access the system using another user’s credentials. In conjunction with MFA, AWS CloudTrail could also be used with the purpose of logging user activity.

### Rekognition

AWS Rekognition includes a variety of safety measures which can be applied to ensure your data and resources are safe. A hosting service known as Amazon Virtual Private Cloud (VPC) establishes a private connection to facilitate your resources. Using this service with Rekognition gives you the ability to control your network settings such as IP address, subnets, route tables and network gateways ensuring your security requirements are thoroughly fulfilled.

Furthermore, CloudWatch can be applied, with the aim of monitoring metrics for individual or global operations. Metrics has the capability of tracking your solutions health as well as the ability to create alarms to notify you when a defined threshold has been breached. Metrics can be viewed for the number of server errors that have occurred, quantity of faces detected and the number of times an operation has succeeded.

### DynamoDB

To help anticipate and prevent security flaws within AWS DynamoDB, there are a few methods which can be applied. Controlling user access by using IAM policy conditions can greatly reduce risks from errors or malicious intent. An example of a permission is allowing read or write only access to items and attributes in a table.

An additional method to prevent security flaws in your application is to consider client-side encryption. A database with sensitive details should be encrypted to protect it from malicious users as in the event of a database breach the information will be unharmed. A library available in AWS termed DynamoDB Encryption Client, aims to protect table data before it is sent to the database.

## Application Testing

### Console UI

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Description | Expected Output | Actual Output | Screenshot |
| 1 | User starts program, UI is shown with options to select | UI is shown | UI is shown |  |

### Upload File to S3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Description | Expected Output | Actual Output | Screenshot |
| 1 | User selects option 1 and is prompted to select a file | File dialog is prompted | File dialog is prompted | A screenshot of a cell phone  Description automatically generated |
| 2 | Error message displays and user is prompted for file if cancel is chosen | Error Message | Error Message | A screenshot of a social media post  Description automatically generated |
| ID | **Description** | **Expected Output** | **Actual Output** | **Screenshot** |
| 3 | User selects jpg file type | Shows jpg images | Shows jpg images | A screenshot of a cell phone  Description automatically generated |
| 4 | User selects png files | Shows png files | Shows png files | A screenshot of a cell phone  Description automatically generated |
| ID | **Description** | **Expected Output** | **Actual Output** | **Screenshot** |
| 5 | User selects a file and is prompted to enter a file name | Prompted for file name | Prompted for file name | A screenshot of a cell phone  Description automatically generated |
| 6 | User submits with no file name | Error Message | Error Message | A screenshot of a cell phone  Description automatically generated |
| 7 | User submits with no file extension | Error Message | Error Message | A screenshot of a cell phone  Description automatically generated |
| ID | **Description** | **Expected Output** | **Actual Output** | **Screenshot** |
| 8 | Successfully submits a file with correct file name and extension | Success Message | Success Message | A picture containing sitting, black, table  Description automatically generated |
| 9 | File is uploaded to S3 | File uploaded | File uploaded | A screenshot of a cell phone  Description automatically generated |

### Download File from S3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Description | Expected Output | Actual Output | Screenshot |
| 1 | User selects option 2 to download files from S3 bucket | List of files shown to available download | List of files shown to available download | A screenshot of a computer  Description automatically generated |
| 2 | User submits with no file name entered | Error Message | Error Message | A screenshot of a cell phone  Description automatically generated |
| 3 | User attempts to submit with no file extension | Error Message | Error Message | A close up of a screen  Description automatically generated |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Description | Expected Output | Actual Output | Screenshot |
| 4 | Successfully downloaded file from S3 | File downloaded | File downloaded | A close up of a sign  Description automatically generated |
| 5 | File downloaded to the user’s desktop | File downloaded | File downloaded | landscape1 Properties |

### List all files in S3 bucket (Extended Functionality)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Description | Expected Output | Actual Output | Screenshot |
| 1 | User selects option 3 to view all files selected in the S3 bucket | Files Listed | Files Listed | A black sign with white text  Description automatically generated |

### List all buckets in S3 (Extended Functionality)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Description | Expected Output | Actual Output | Screenshot |
| 1 | User selects option 4 to view all buckets in S3 | Buckets Listed | Buckets Listed | A screenshot of a cell phone  Description automatically generated |

### Quit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Description | Expected Output | Actual Output | Screenshot |
| 1 | User selects option 5 to exit the application successfully | Exists application | Exists application | A close up of a logo  Description automatically generated |

### AWS – Image Upload Notification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Description | Expected Output | Actual Output | Screenshot |
| 1 | Image upload to S3 sends a message to the SQS Queue | Message delivered | Message delivered | Below |

A screenshot of a social media post

Description automatically generated

### Receive and process message from SQS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Description | Expected Output | Actual Output | Screenshot |
| 1 | Parse SQS message to extract image name | Image name | Record body | A screenshot of a social media post  Description automatically generated  A screenshot of a cell phone  Description automatically generated |

### Pass Image Details to Rekognition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Description | Expected Output | Actual Output | Screenshot |
| 1 | Image details must be sent to the AWS Rekognition service to obtain image labels. | Image Labels returned | Image Labels returned |  |

### Receive and process image labels from Rekognition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Description | Expected Output | Actual Output | Screenshot |
| 1 | The information returned from Rekognition must be processed to extract image label data but only a maximum of five labels per image should be used. | Image labels extracted with maximum of 5 labels | Image labels extracted with maximum of 5 labels | A screenshot of a social media post  Description automatically generated |

### Store the labels in a DynamoDB table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Description | Expected Output | Actual Output | Screenshot |
| 1 | Only the first five labels received from Rekognition service should be stored in the table | N/A | N/A | N/A |

# References

Amazon, 2020. *Amazon DynamoDB.* [Online]   
Available at: https://aws.amazon.com/dynamodb/  
[Accessed 19 03 2020].

Amazon, 2020. *Amazon Rekognition.* [Online]   
Available at: https://aws.amazon.com/sqs/  
[Accessed 17 03 2020].

Amazon, 2020. *Amazon S3.* [Online]   
Available at: https://aws.amazon.com/s3/  
[Accessed 12 03 2020].

Amazon, 2020. *Amazon Simple Queue Service.* [Online]   
Available at: https://aws.amazon.com/sqs/  
[Accessed 13 03 2020].

Amazon, 2020. *Amazon SQS best practices.* [Online]   
Available at: https://docs.aws.amazon.com/AmazonS3/latest/dev/s3-access-control.html  
[Accessed 23 03 2020].

Amazon, 2020. *AWS Lambda.* [Online]   
Available at: https://aws.amazon.com/sqs/  
[Accessed 14 03 2020].

Amazon, 2020. *Data Protection in AWS Lambda.* [Online]   
Available at: https://docs.aws.amazon.com/lambda/latest/dg/security-dataprotection.html  
[Accessed 24 3 2020].

Amazon, 2020. *DynamoDB Preventative Security Best Practices.* [Online]   
Available at: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/best-practices-security-preventative.html  
[Accessed 27 03 2020].

Amazon, 2020. *Identity and Access Management in S3.* [Online]   
Available at: https://docs.aws.amazon.com/AmazonS3/latest/dev/s3-access-control.html  
[Accessed 23 03 2020].

Amazon, 2020. *Monitoring Rekognition.* [Online]   
Available at: https://docs.aws.amazon.com/rekognition/latest/dg/rekognition-monitoring.html  
[Accessed 24 03 2020].

Microsoft, 2020. *AWS to Azure Services Comparison.* [Online]   
Available at: https://docs.microsoft.com/en-us/azure/architecture/aws-professional/services  
[Accessed 21 03 2020].