Glasgow Caledonian University

University for the Common Good

School of Computing, Engineering and Built Environment

Department of Computing

Module Leader: Sajid Nazir

Coursework Specification

Cloud Platform Development (MHI326408/MHI325614)

Session 2021-22, Trimester B, Diet 1

Submission Due: 17th April 2022

Object and PPE Detection with AWS Artificial Intelligence (AI) Service

Introduction

The purpose of this coursework is to allow you to demonstrate your understanding of cloud applications and software development on the AWS cloud platform. You will design and implement an application for **Label and Personal Protection Equipment (PPE) Detection** of image files using the **AWS Rekognition** (AI services).

This is an individual coursework and your attention is drawn to the university rules on plagiarism and its repercussions.

'The deliberate and substantial unacknowledged incorporation in your submission of material derived from work (published or unpublished) of another is prohibited. In such cases the work submitted is classified as illegal and will result in further action being taken by the University which may include recording a failure.'

Specification

The coursework consists of 5 parts (**90%** of the available marks) and the creation of a video (**10%** of the available marks) demonstrating the application, totalling 100% of the overall coursework marks for the module. The application should be implemented on the **AWS Academy Learner Lab** environment using **Boto3 Amazon Web Services** (**AWS) SDK** as specified below**.** A set of image files will be provided and further details to guide you are provided in the 'Hints and Resources' document.

An EC2 instance simulates a camera system providing images for label and PPE detection. You will be provided with some images that should be sent by your code running on EC2 to S3 bucket. The flow indicating the sequence of actions to be implemented is shown in Figure 1. The detailed steps from 1-5 are described below:

1. Resource creation with Boto3 and CloudFormation (20 marks)

You must use Python (Boto3) to create the EC2 instance, SNS topic and the S3 bucket. You are not required to create subnets, route tables etc. and should use the default VPC and subnet for this EC2.

The SQS queue and DynamoDB table must be created using a CloudFormation template.

2. Image file upload to S3 and Lambda trigger from SQS (10 marks)

A set of image files will be provided for detecting the labels and PPE. The Python application (Boto3) running on the EC2 instance should upload these files one after the other at 30 secs intervals to the S3 bucket. The image file upload must send a message to the SNS which in turn notifies the SQS

queues. The SQS queue must trigger the corresponding Lambda function. You do not need to send the message through your code however this would need to be configured by you.

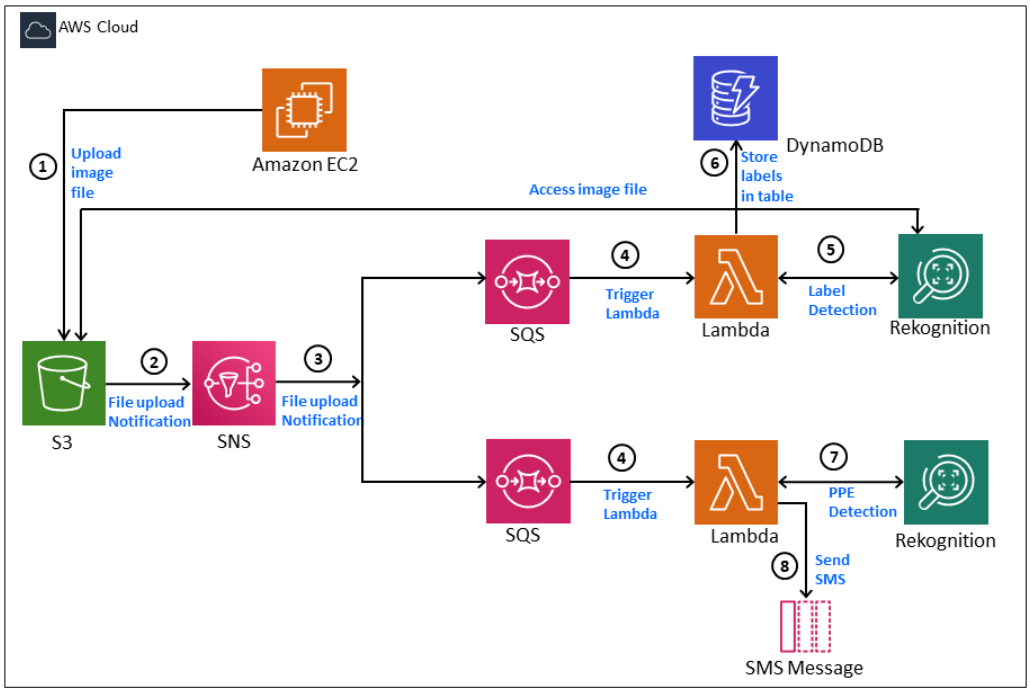

Figure 1: Architecture of the Label and PPE Detection System

3. Label and PPE Detection with AWS Rekognition (20 marks)

You can use AWS console, CLI or SDK to create a Lambda function. The Lambda function code in Python (Boto3) must extract relevant details such as image file name etc. from the SQS message. The file details must be sent by the Lambda code using an appropriate call to the AWS PPE Detection service to detect the **face** and **hand** cover, and for label detection.

4. Database Update and SMS Notification (20 marks)

The DynamoDB database should have a single table with only a single partition (primary) key as the image name.

From the Label Detection response (part 3 above), the Lambda function must extract relevant information and save the results for a maximum of five labels with the confidence scores in the DynamoDB table.

For the PPE Detection responses indicating non-adherence to face/hand covering, the Lambda function must immediately notify a given telephone number using SMS. You may use any telephone number for implementation/testing but should use ZZ-ZZZZZZZZZZ for coursework submission.

Important Instructions:

1. **Creating regular backups** of your code is your own responsibility. Accidently pressing 'reset' in the Learner Lab environment can potentially lose all your work.

2. You must append your **student ID** to all resource (S3 bucket, Lambda etc.) names that you create on AWS, e.g. for a student ID such as 1234567, a bucket would be named so that the ID is appended to it, with names such as mybucket1234567, xyz1234567. This information will be used for plagiarism checks.

3. You will need to setup IAM roles, policies and permissions for implementing your solution. These are important for the code functionality and the **details must be submitted** as part of your code or in the report.

4. The code should demonstrate your understanding of writing **good software**, including structure, comments, ease of use, and error checks.

5. **Do not submit your AWS credentials** as part of your coursework. These should never be a part of your code but if you have used them then remove AWS access keys etc. prior to submission.

6. Any access/permission policy that you create on AWS should be **most restrictive**, that is, allowing only the required degree of access to only the required entity.

   You should provide for each use how you would restrict the access to required entities and services only, in case of using the pre-created LabRole within the Learner Lab environment.

5. Report

The cover page of the report must contain the submitting student's name, student ID, programme name, and the following declaration of ownership:

"I declare that all work submitted for this coursework is the work of <*insert name of the author*> alone unless stated otherwise."

The report must be submitted in Microsoft **Word** format with a contents page and the following sections:

a. Problem and cost-optimised solution **(10 marks)**

- Description of problem and your approach to solve it. You can also suggest improvements/changes to the application architecture

shown in Figure 1 providing justifications.

- Description of how the application can be cost optimised considering many image files uploads and running over a longer period of time

b. Security features and application testing **(10 marks)**

- Brief description of how the application can be secured using the features available in AWS

- Application Testing

You should also use the report to describe any specific instructions regarding your code execution and to include evidence of a feature that is implemented by you and is not part of your code submission.

Coursework Submission

Submission deadline: 23:59 pm, Sun, 17th April 2022 to GCULearn

1. Code

The code must be implemented for AWS cloud platform in Python SDK (Boto3). The code submission should include the source code files and CloudFormation template. It can also include any roles, policies etc. in JSON.

You must submit the zip file containing all files to GCULearn using the link provided (note zip, not rar etc. and not password protected). Name the zip file as: **<student_id>CPD.zip** e.g. S1234567CPD.zip

2. Report

The report should be submitted as Microsoft Word format (not as a pdf etc.). You can cut and paste screenshots from the AWS management console into the report to illustrate the application configuration/running/testing.

Name the document as: **<student_id>CPD.doc(x) e.g. S1234567CPD.doc(x)**

3. Video Recording of Project Demonstration

You must submit a video recording of max 5-10 mins duration demonstrating the application working on the AWS platform. You must describe all the features and how they were implemented. You can use any software to create the video but it must be submitted in (.mp4) format.

Name the video file as: **<student_id>CPD.mp4 e.g. S1234567CPD.mp4**