*Wright, Iona J*

*S1108900 | Software Development for Business*

*Cloud platform development*

Coursework

Table of Contents

[Problem and cost-optimised solution (10 marks) 2](#_Toc101049831)

[Description of problem and your approach to solve it. 2](#_Toc101049832)

[Syntax Errors 2](#_Toc101049833)

[Bucket Name 2](#_Toc101049834)

[Subscription to SNS 3](#_Toc101049835)

[Cloud Formation Template 4](#_Toc101049836)

[Description of how the application can be cost optimised 5](#_Toc101049837)

[Security features and application testing (10 marks) 6](#_Toc101049838)

[Brief description of how the application can be secured using the features available in AWS 6](#_Toc101049839)

[Application Testing 6](#_Toc101049840)

[References: 7](#_Toc101049841)

I declare that all work submitted for this coursework is the work of Iona Wright alone unless stated otherwise.

# 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)

## Syntax Errors

**Installing boto3**

When I got cloud9 running I initially had to install boto3 but made the mistake of putting:

sudo pip install bojo3

sudo pip3 install bojo3

## **Bucket Name**

The bucket name can be between 3 and 63 characters long, and can contain only lower-case characters, numbers, periods, and dashes. I didn’t realise that this was a problem when creating the bucket inside the python script I was presented with the error when running the script.

Text

Description automatically generated

From research online I found the answer here:

<https://docs.aws.amazon.com/AmazonS3/latest/userguide/bucketnamingrules.html>

Python and YAML are fussy

Error EOL – end of line

Single quotes instead of double (add screenshot)

## Subscription to SNS

Btw one important hint for when you're setting up the queue and subscription make sure you tick the box that says enable raw message delivery. You'll get all sorts of weird characters in the messages that will make extracting file names a pain in the arse if you don't.

OS function for uploading an image makes it compatible with Linux, OS & windows

Lambda function:

A Lambda function can process items from multiple queues (using one Lambda event source for each queue). <https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-configure-lambda-function-trigger.html>

Notification from s3 bucket to SNS

When attempting to create a notification from the S3 bucket to the SNS it was erroring with a destination failure.

Text

Description automatically generated with medium confidence

From a google search I found documentation:

<https://docs.aws.amazon.com/AmazonS3/latest/userguide/ways-to-add-notification-config-to-bucket.html#step1-create-sns-topic-for-notification>

“Because of the way that AWS CloudFormation handles dependency ordering, Amazon S3 event notifications are defined as an attribute of the S3 bucket. These notifications are established when the S3 bucket resource is created.”

Therefore to avoid getting this error I created the resources in a specific order, firstly the SNS topic (as the S3 bucket references this), then created the S3 bucket. To do this I updated my python script so that the SNS topic was code was above the bucket creation.

Once I created the resources in the correct order I needed to then specify the topics access policy with permissions. I made the mistake of creating the subscription first before the topic policy.

notification needed an updated polcy

Step 2: Create an Amazon SNS topic

Follow the steps to create and subscribe to an Amazon SNS topic.

1. Using Amazon SNS console, create a topic. For instructions, see [Creating an Amazon SNS topic](https://docs.aws.amazon.com/sns/latest/dg/CreateTopic.html) in the Amazon Simple Notification Service Developer Guide.
2. Subscribe to the topic. For this exercise, use email as the communications protocol. For instructions, see [Subscribing to an Amazon SNS topic](https://docs.aws.amazon.com/sns/latest/dg/sns-create-subscribe-endpoint-to-topic.html) in the Amazon Simple Notification Service Developer Guide.

You get an email requesting you to confirm your subscription to the topic. Confirm the subscription.

1. Replace the access policy attached to the topic with the following policy. In it, provide your SNS topic ARN, bucket name, and bucket owner's account ID.

<https://docs.aws.amazon.com/AmazonS3/latest/userguide/ways-to-add-notification-config-to-bucket.html>

## Cloud Formation Template

The Dynamo DB & SQS queue was to be created using the cloud formation template. This meant creating a YAML file to upload to via cloud formation stack. On the first upload the stack failed and rolled back the complete meaning it could be deleted straight away.

Graphical user interface, website

Description automatically generated

Before deleting, I checked the events tab highlight in figure … to find a meaningful error. In this scenario there was an attribute for the SQS queue policy as I added in an attribute that was no recognised:

ContentBasedDeduplication

As this was not a required field I was able to just remove it from the yaml file and take the default settings in AWS.

## Description of how the application can be cost optimised

(considering many image files uploads and running over a longer period of time)

Log events show how much you are billed for

Amazon SNS provides in-transit encryption by default but not at-rest. Enabling at-rest to the server side will increased the security.

SQS queue

KmsDataKeyReusePeriodSeconds

The length of time in seconds for which Amazon SQS can reuse a data key to encrypt or decrypt messages before calling AWS KMS again. The value must be an integer between 60 (1 minute) and 86,400 (24 hours). The default is 300 (5 minutes).

MaximumMessageSize

The limit of how many bytes that a message can contain before Amazon SQS rejects it. You can specify an integer value from 1,024 bytes (1 KiB) to 262,144 bytes (256 KiB). The default value is 262,144 (256 KiB).

# Security features and application testing (10 marks)

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

AWS IAM – security credentials: <https://www.youtube.com/watch?v=5q7FtT_DyME>

Create our own (SSH) key – don’t use the default

## Application Testing

Graphical user interface, text, application

Description automatically generated

# References:

Create SQS

<https://boto3.amazonaws.com/v1/documentation/api/latest/guide/sqs.html>

Cloud Formation:

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/Welcome.html>

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-sqs-queue.html#cfn-sqs-queue-deduplicationscope>

Upload a file boto3

<https://boto3.amazonaws.com/v1/documentation/api/latest/guide/s3-uploading-files.html>

Working with SNS Boto

<https://towardsdatascience.com/working-with-amazon-sns-with-boto3-7acb1347622d>

SQS boto3

<https://www.learnaws.org/2020/12/17/aws-sqs-boto3-guide/>

S3 bucket notification

<https://docs.aws.amazon.com/AmazonS3/latest/userguide/NotificationHowTo.html>

<https://docs.aws.amazon.com/AmazonS3/latest/userguide/ways-to-add-notification-config-to-bucket.html#step1-create-sns-topic-for-notification>

AWS Recoginition

<https://docs.aws.amazon.com/rekognition/latest/dg/ppe-detection.html>