PROJECT REPORT ON

IMAGE RECOGNITION AND SIMPLE NOTIFICATION SYSTEM USING AWS

SUBMITTED BY:

17BIT0337

JASHKUMAR SHAH

17BIT0368

OM PUROHIT

SUBMITTED TO:

PROF. PRIYA V

WINTER SEMESTER 2018-2019

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

April 2019

TABLE OF CONTENT

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
AWS	Amazon Web Services
SNS	Simple Notification Service
S3	Simple Storage Service
SES	Simple Email Service
IAM	Identity and Access Management

ABSTRACT

The image recognition detects all parts of the image and it is filtered to produce necessary information only as required by the user, from a large dataset of images of a particular category the product will analyse and check for if the required details of the user matches the image properties and whether to send an email or not if it does not contain the required specifications. This project consists of an image recognition system and a notification system where the user gets notified via email the result of the image recognition using the Amazon Web Services (AWS).

Amazon Web Services (AWS) is a cloud service from Amazon, which provides services in the form of building blocks, these building blocks can be used to create and deploy any type of application in the cloud. These services or building blocks are designed to work with each other, and result in applications which are sophisticated and highly scalable

Amazon Simple Notification Service (SNS) is a cloud service for coordinating the delivery of push messages from software applications to subscribing endpoints and clients. All messages published to Amazon SNS are warehoused across several availability zones to prevent loss. Amazon SNS allows users to push messages to Windows, Google, Apple and certain internet-connected smart devices by using an application programming interface (API) or the AWS Management Console. Once a message is published to the service, it can be sent multiple times to different recipients. Service users also have the flexibility to send direct messages to several devices or to one subscriber by using a sole publish request.

This project is aimed at performing image recognition on different images using the cloud services effectively provided by Amazon. Along with this, all the concepts and necessary steps to perform has been described in detail.

CHAPTER 1

INTRODUCTION

Image recognition is used to perform a large number of machine-based visual tasks, such as labelling the content of images with meta-tags, performing image content search and guiding autonomous robots, self-driving cars and accident avoidance systems.

The work-flow is divided into 4 phases, each having its own importance and provides good coupling with the other phases.

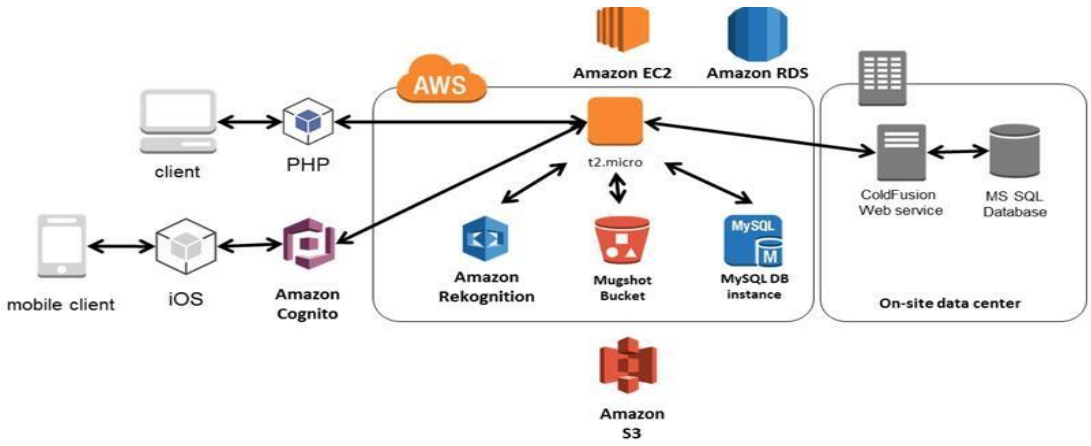

Figure 1 - Overview of AWS

The phase -1is creating a portal or an instance for storing our images that the user shall upload. It requires a convenient, fast and always available mechanism. This is provided by Amazon Web Service in the S3(simple storage) section the S3 Bucket will be sufficient for our storage needs, but the main advantage is that the S3 bucket will make phase-2 less tedious.

In phase-2 we focus on implementing a charge-free message sending model to notify the user of the results of our whole process and the outcome he/she desires. It will require the user to register it's email first for the service to work and for this we use AWS called SNS (simple notification service) where we will use SES simple email service where the email of our making will be sent to the registered email.

In phase-3 we require to implement the core process of our product that is the recognition of images. With various possible and efficient available machine learning models for image recognition we choose to go with the famous AWS Recognition service as it is trusted and will help our project to be centred across a common service which is AWS. This phase is just the AWS recognition doing the magic that it does and giving a detailed description and identifying our image

In Last phase, i.e. phase-4 our project but definitely not the least we need to couple all our phases using the Lambda service available and we choose Python as programming language due to its extensive libraries and especially BOTO3 in particular. The lambda function will tie all loose ends of our project and will create the domino effect we so require.

SOME IMPORTANT CONCEPTS

5.1 <u>AMAZON WEB SERVICES (AWS)</u> - Amazon Web Services (AWS) is a secure cloud services platform, offering compute power, database storage, content delivery and other

functionality to help businesses scale and grow. One can explore how millions of customers are currently leveraging AWS cloud products and solutions to build sophisticated applications with increased flexibility, scalability and reliability. The AWS Cloud provides a broad set of infrastructure services, such as computing power, storage options, networking and databases, delivered as a utility: on-demand, available in seconds, with pay-as-you-go pricing.

5.2 AWS LAMBDA AND AWS RECOGNITION - APIs that will tell us what the image is about. For example, whether something is pizza or not. Rekognition is used for image analysis and recognition it creates a dataset consisting all the details detected by the machine learning algorithm. Lambda is where we will create a function code via Python where all the various services are synced and used, almost acting as the brain of the system.

5.3 AWS SNS - Amazon Simple Notification Service (SNS) is a highly available, durable, secure, fully managed pub/sub messaging service that enables you to decouple microservices, distributed systems, and serverless applications. Amazon SNS provides topics for high-throughput, push-based, many-to-many messaging.

5.4 AMAZON S3 - Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. This means customers of all sizes and industries can use it to store and protect any amount of data for a range of use cases, such as websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics.

5.5 PYTHON - Using python for programming. The result will be emailed to the subscribed email containing a detailed description of the image and its contents that is visible to the human eye but hard to detect via any non-machine learning algorithm.

CHAPTER 3

ANALYSIS AND DESIGN

6.1 Requirement Analysis –

6.1.1 Account on Amazon Web Services (AWS) Console. For trial basis, free trial period is recommended.

Access to services on AWS -

6.1.2 AWS Lambda – where the function code will be created and it is the brain of the system.

6.1.3 AWS Recognition – contains the dataset of all the images.

6.1.4 AWS SNS – sends message to the user.

6.1.5 Amazon S3 Bucket – space where user can upload and store the images.

6.1.7 Python platform – python programming is done.

6.2 System Design –

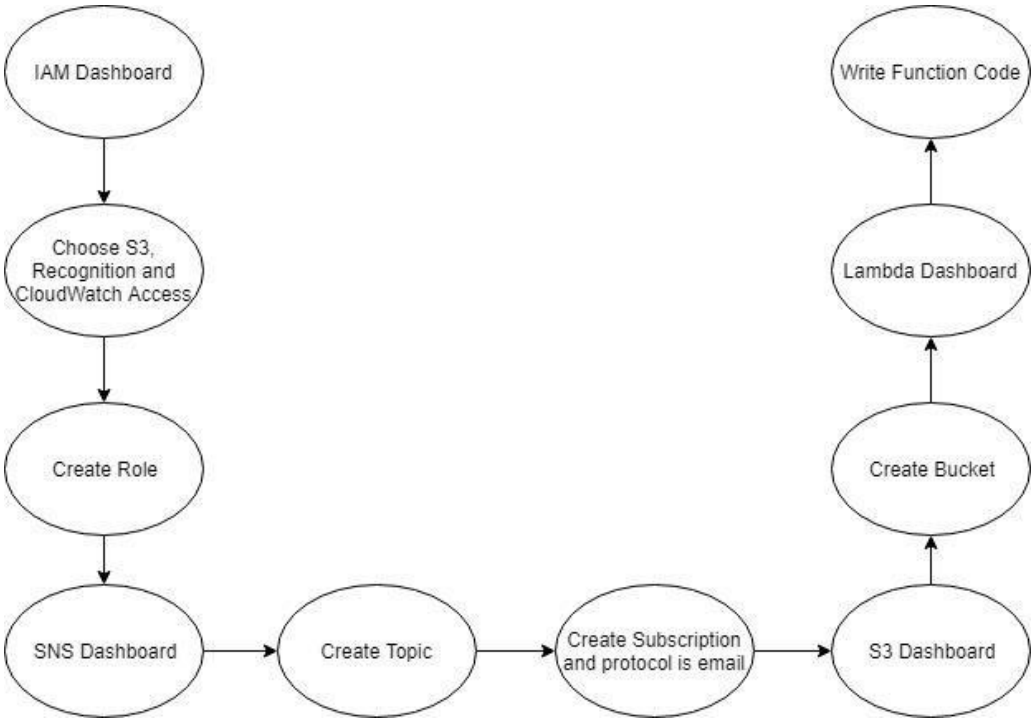

Figure 2 – System Design

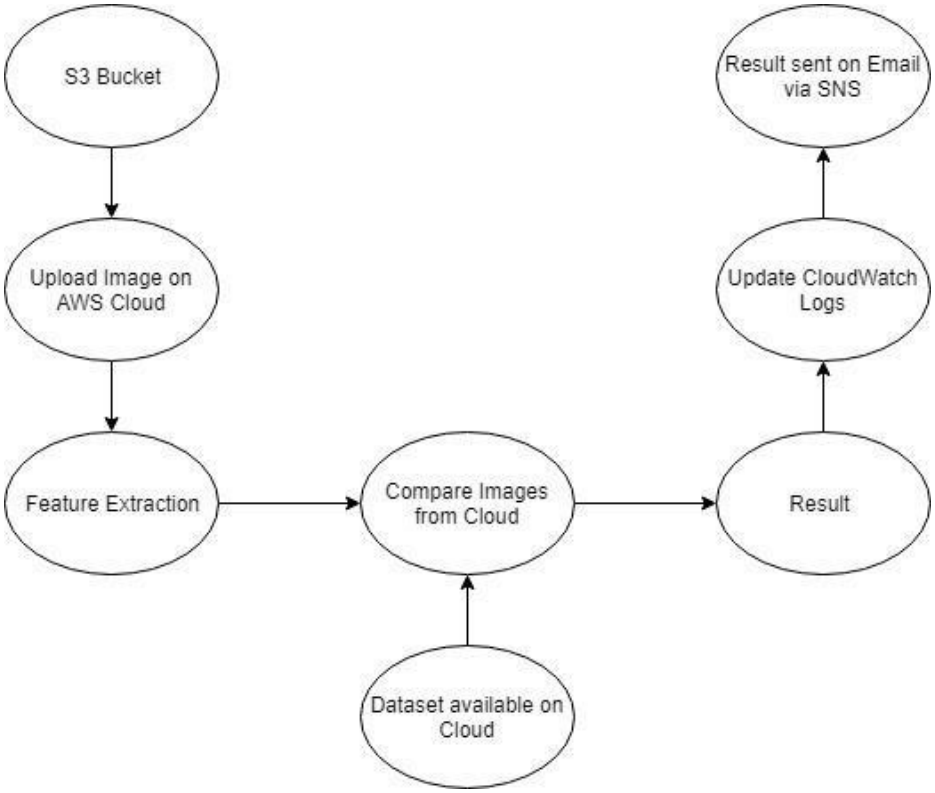

Figure 3 - Image Recognition Design

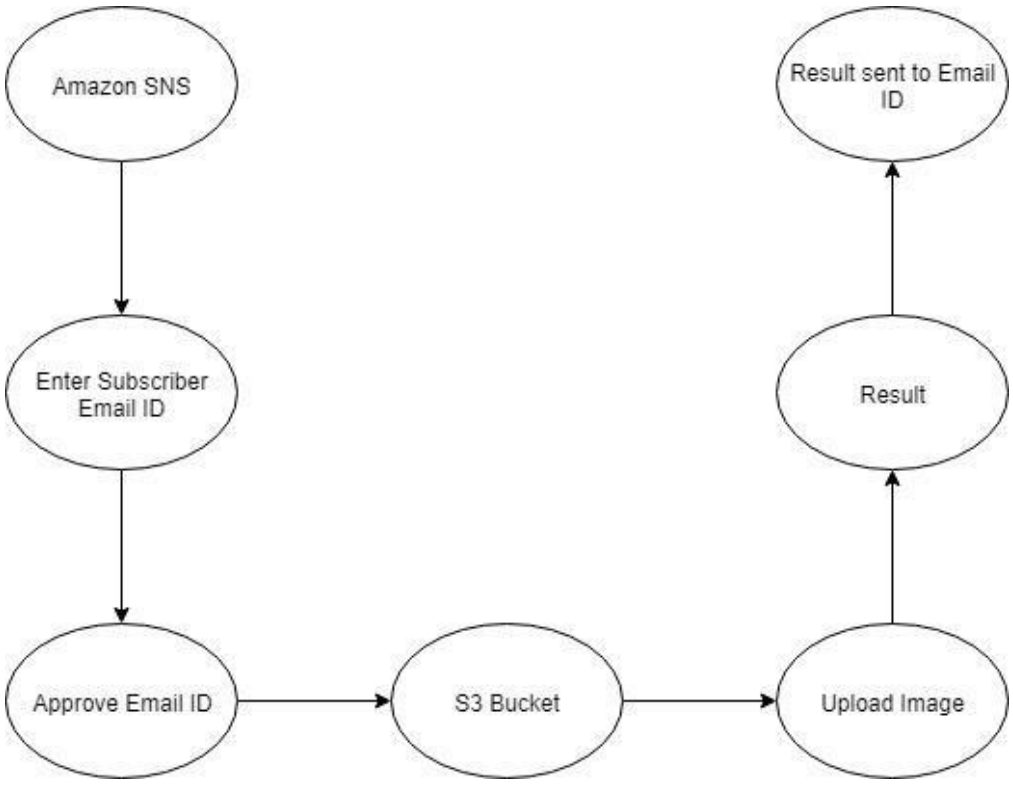

Figure 4- SNS Design

IMPLEMENTATION

7.1 Create an IAM Role with recognition, cloud watch and S3 access.

7.1.1 Open up the IAM dashboard.

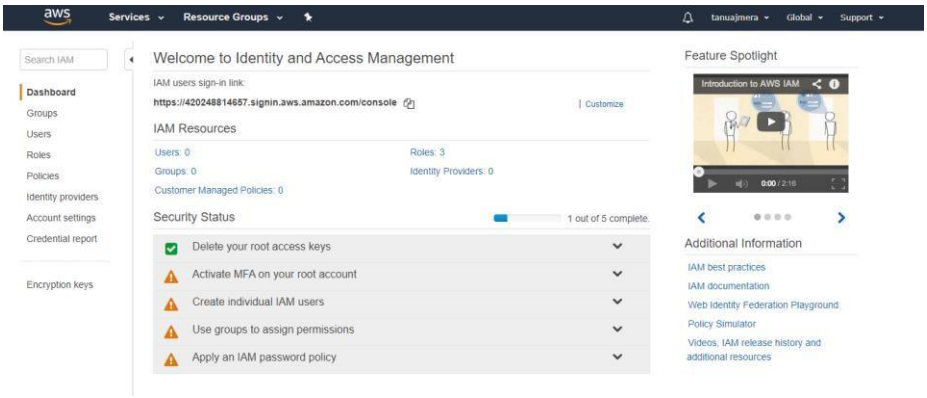

Figure 5 - IAM Dashboard

7.1.2 Click on roles and click on create a role.

7.1.3 Choose AWS Role and choose lambda. We choose lambda because we are trying to give lambda the permissions to access various other resources.

7.1.4 Then, we go for next permissions.

7.1.5 We find S3 in search bar and choose “AmazonS3FullAccess”.

7.1.6 After this, we find cloud watch access and choose “CloudwatchFullAcsess”.

7.1.7 Finally, we find recognition and choose “AmazonRekognitionFullAcess”. This is the machine learning API, where the image recognition happens.

7.1.8 We go to next step and we give the Role-Name and Role-Description, both with the same attribute. And finally, click on create role.

7.2 Create an SNS Topic.

7.2.1 On the SNS dashboard, choose create a topic.

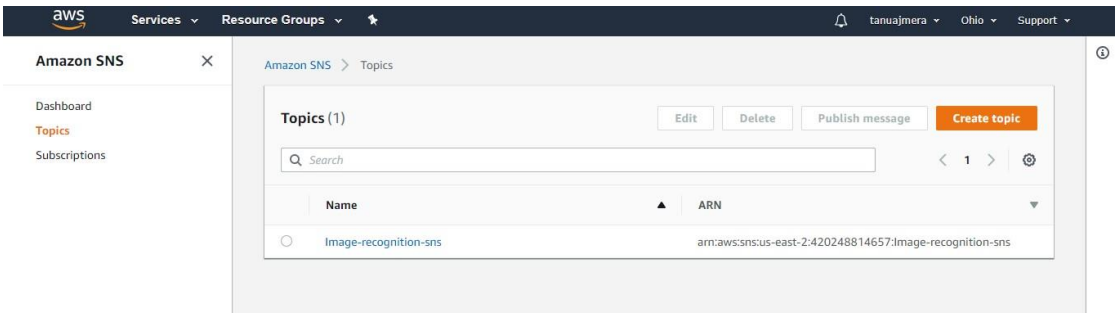

Figure 6 - SNS Dashboard

7.2.2 Give the topic name and other fields according to your choices.

7.2.3 After the topic is create, we choose create a subscription.

7.2.4 Choose protocol as “Email”.

7.2.5 Give your email. And finally, click create subscription.

7.2.6 Go on your email and check for confirmation. A message like this should appear.

Figure 7 - SNS Confirmation

7.3 Create an S3 Bucket.

7.3.1 On the dashboard, click create a bucket.

7.3.2 Give the details in fields asked for.

7.3.3 Click on next until the bucket is created.

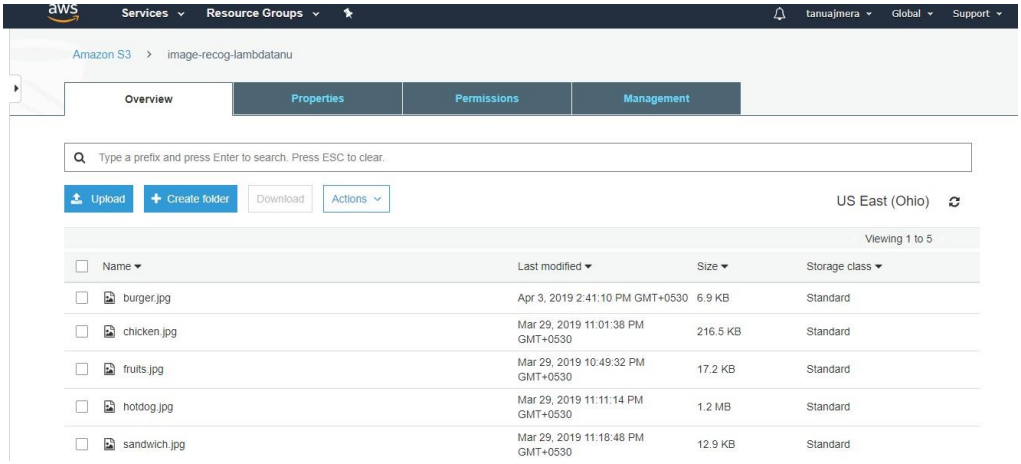

Figure 8 - S3 Bucket

7.4 Create a lambda function.

7.4.1 Finally, on the dashboard choose create a function.

7.4.2 Choose the “Blueprint” option from the given options. Blueprints are the preconfigured template that can act as point for our lambda function and saves our work.

7.4.3 In the search bar of blueprints, type “rekog”. We will find a box with recognition-python and amazon S3 trigger. This uses recognition APIs to detect faces.

7.4.4 A warning will flash, ignore the warning and provide the details for the fields that are asked for. Choose the existing role that we created earlier. Also, choose the same bucket that we created and leave event type attribute as “Object Created (All)”.

7.4.5 Make sure that we have enabled trigger.

7.4.6 Finally, click on create functions.

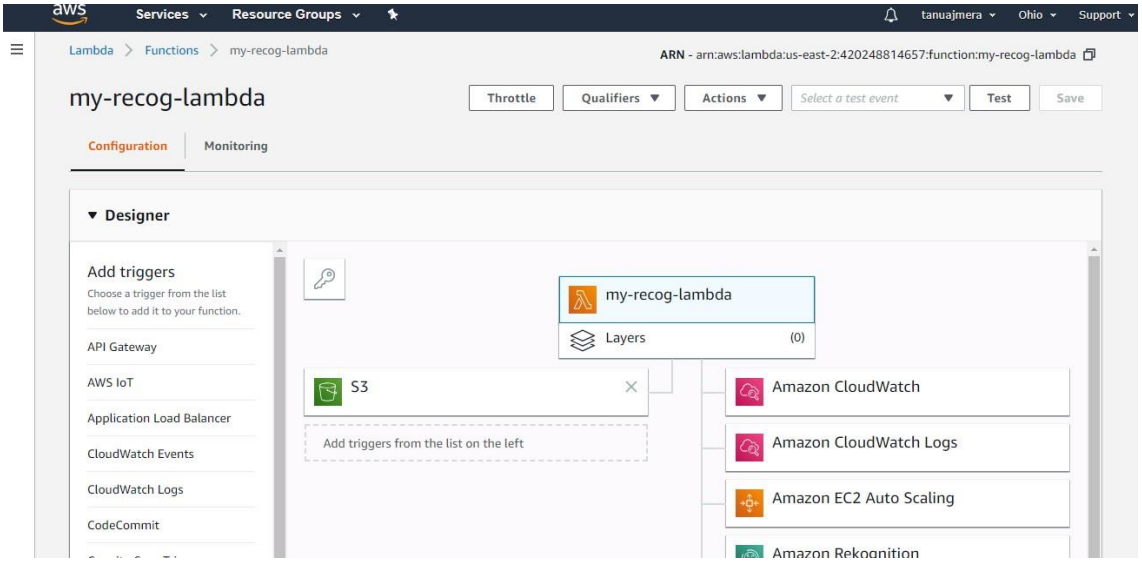

Figure 9 - Lambda Dashboard

Algorithm –

```
rekognition = boto3.client('rekognition') client = boto3.client('sns') def
detect_labels(bucket, key):     response = rekognition.detect_labels(Image={"S3Object":
{"Bucket": bucket, "Name": key}}) label_prediction['Name']} for label_prediction in
response['Labels']]     return response def lambda_handler(event, context):
   bucket = event['Records'][0]['s3']['bucket']['name']
key    =    urllib.unquote_plus(event['Records'][0]['s3']['object']['key'].encode('utf8'))
try:
      response = detect_labels(bucket, key)
tosend=""         for Label in
response["Labels"]:
         print ('{0} - {1}%'.format(Label["Name"], Label["Confidence"]))
tosend+= '{0} - {1}%'.format(Label["Name"], Label["Confidence"])
print(response)
      message = client.publish(TargetArn='arn:aws:sns:us-east-
2:420248814657:Imagerecognition-sns', Message=tosend , Subject='Uploaded Image
Labels')        return response
```

CHAPTER 4

RESULT AND DISCUSSION

Image Recognition System is successfully implemented using cloud services via Amazon Web Services (AWS). Once the user uploads the image in .jpg format on S3 Bucket, it immediately sends a notification to the specified email address regarding the details of the image, i.e., confidence of each related term. The figure given below is an example of one of the upload.

Figure 10 – Input Image

Figure 11 - Result of Uploaded Image via SNS

To check the history of all the uploaded images and the results, user can direct to the CloudWatch Logs where all the details are given with specified time. It is helpful in maintaining records and analysing the results.

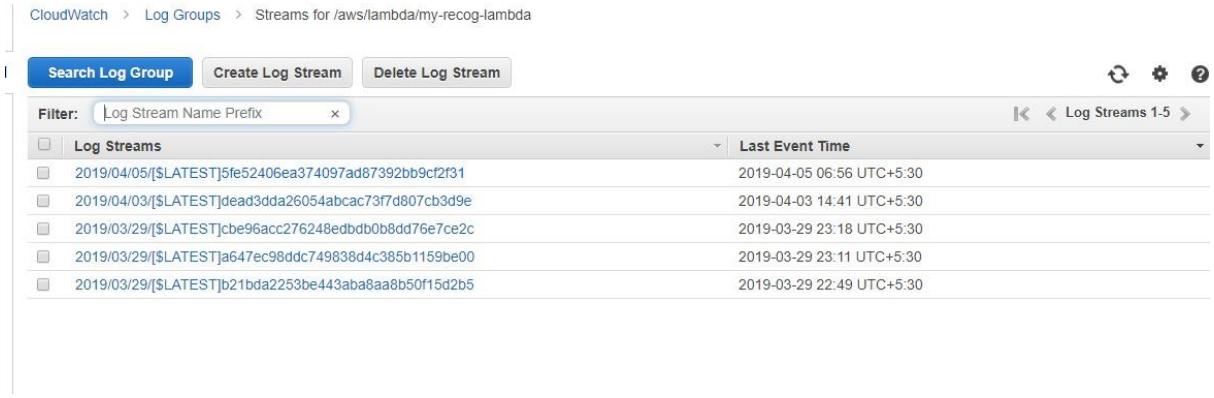

Figure 12 - CloudWatch Logs

User can even see the results in the form of graphs in the Lambda Service of AWS as well as list of logs as mentioned earlier. These graphs see the error, accuracy and the times taken to upload the image and receive notification about it. It is helpful as one can find accurate results about the functioning of the Image Recognition System.

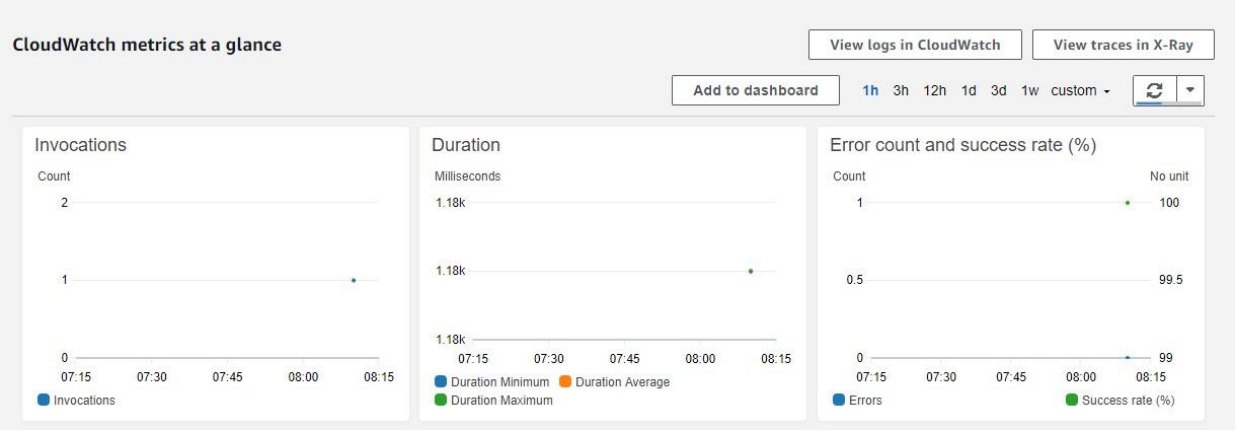

Figure 13 - Graphs on Cloud

As a result, a user can easily upload images on cloud using Amazon S3 Service by creating a bucket and receive notifications on their email ID by subscribing to it using SNS Service. The accuracy of image recognition is high as dataset is already available on cloud.

CHAPTER 5

CONCLUSION

An Image Recognition System and a Simple Notification System using Amazon Web Services (AWS) has been successfully and effectively implemented. The use of services provided by Amazon Web Services is used in an efficient manner. The ease of datasets available on the cloud has made the process of recognition easier and faster compared to other recognition systems. Amazon Web Services is a great cloud platform to implement projects due to the availability of various datasets and services provided.

As observed in the above implementation section, steps to create an image recognition system is easy though a user can modify the python code according to the need of labels of an image. The function code can be modified in the Lambda section.

A Simple Notification System has been implemented properly wherein the user is notified with the details of the image uploaded in the S3 Bucket. Amazon S3 provides storage bucket where the user can upload and store the images in a click.

Amazon Web Services is a very useful cloud platform to carry out projects and has been very helpful for us to implement the image recognition system. User can now upload the images and receive notifications on email via Simple Notification System using SNS on AWS in a faster way.

REFERENCES

1. https://www.researchgate.net/publication/255906555_Cloud-Vision_Realtime_Face_Recognition_Using_a_Mobile-Cloudlet-Cloud_Acceleration_Architecture

2. https://www.sciencedirect.com/science/article/pii/S0016510715027388
3. http://iopscience.iop.org/article/10.1088/1757-899X/308/1/012013
4. https://www.researchgate.net/publication/309609355_Push_Notification_System_to_Mobile_Game_Player_Using_Distributed_Event-Based_System_Approach
5. https://online-journals.org/index.php/i-jim/article/download/5567/4028
6. https://aws.amazon.com