Building Image Label Generator using Amazon Rekognition

Aim:

This project aims to build an image label generator.

Process:

The user uploads an image as an object to an S3 bucket. The image label generator application (a Python script) is then used to label the image, providing the labeled image as the output. This application uses the boto3 and matplotlib libraries. Before running the application, the user must configure their account by entering the Access Key and Secret Access Key in the terminal or command prompt.

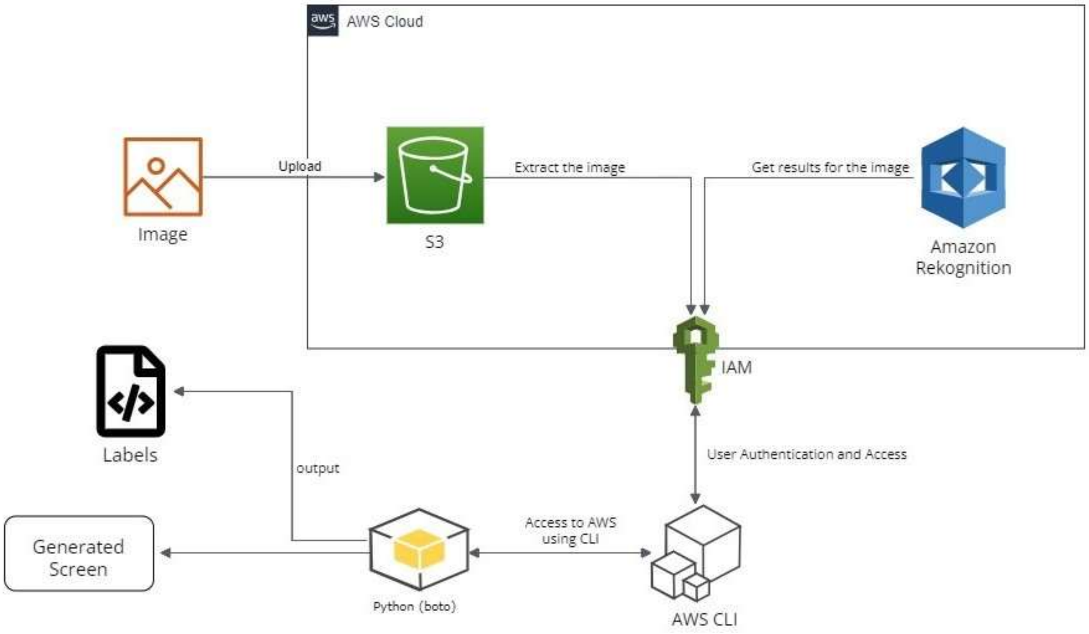
Steps:

1. *Create an S3 bucket.*
2. *Upload images to the bucket.*
3. *Configure the AWS CLI (using 'aws configure' command)*
4. *Write and execute the Python file.*

Services used:

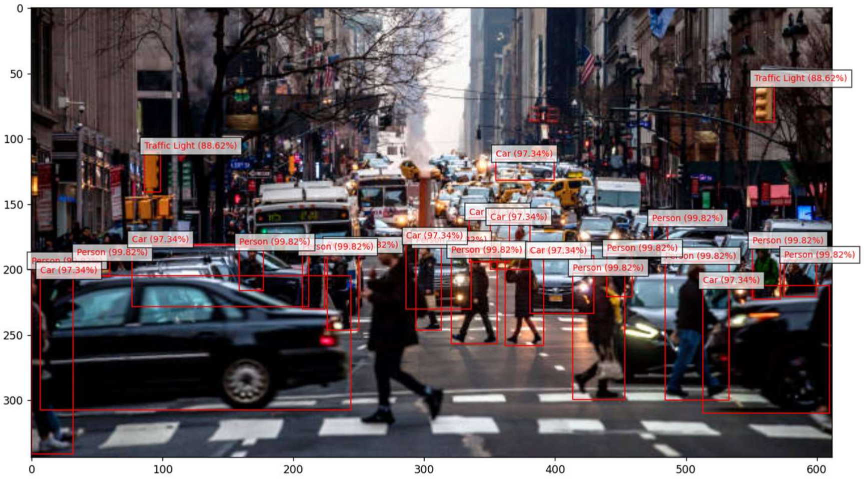
1. *Amazon S3*
2. *Amazon Rekognition*
3. *IAM*

Architecture Diagram:



Result:

An image named "istockphoto.jpg" is uploaded into the S3 bucket. Now, upon running the application, the result is as follows.



app.py

```
import boto3
import matplotlib.pyplot as plt
import matplotlib.patches as patches
from PIL import Image
from io import BytesIO

def detect_labels(photo, bucket):
    client = boto3.client('rekognition')

    response = client.detect_labels(
        Image={'S3Object': {'Bucket': bucket, 'Name': photo}},
        MaxLabels=10)

    print('Detected labels for ' + photo)
    print()

    # Print label information
    for label in response['Labels']:
        print("Label:", label['Name'])
        print("Confidence:", label['Confidence'])
        print()

    # Load the image from S3
    s3 = boto3.resource('s3')
    obj = s3.Object(bucket, photo)
    img_data = obj.get()['Body'].read()
    img = Image.open(BytesIO(img_data))

    # Display the image
    plt.imshow(img)
    ax = plt.gca()

    # Plot bounding boxes
    for label in response['Labels']:
        for instance in label.get('Instances', []):
            bbox = instance['BoundingBox']
            left = bbox['Left'] * img.width
            top = bbox['Top'] * img.height
            width = bbox['Width'] * img.width
            height = bbox['Height'] * img.height

            rect = patches.Rectangle((left, top), width, height, linewidth=1,
edgecolor='r', facecolor='none')
            ax.add_patch(rect)

            label_text = label['Name'] + ' (' + str(round(label['Confidence'], 2))+'%)'
```

```python
            plt.text(left, top - 2, label_text, color='r', fontsize=8, bbox = dict(facecolor='white', alpha=0.7))

    plt.show()

    return len(response['Labels'])

def main():
    photo = 'istockphoto.jpg'
    bucket = 'project1images'
    label_count = detect_labels(photo, bucket)
    print("Labels detected:", label_count)

if __name__ == "__main__":
    main()
```