

Image Label Generator Using Amazon Rekognition

Overview

This project leverages Amazon Rekognition to automatically analyze images and generate descriptive labels for the objects they contain. For example, if a user uploads a photo that includes a dog, Rekognition identifies the object and returns the label “Dog” along with a confidence score. The system not only detects objects but also provides bounding boxes to highlight them within the image, enabling accurate visualization and real-time insights.

Services Used:

- Amazon S3 for storing the images in the process of generating labels.
- Amazon Rekognition to analyse images and generate image labels.
- AWS CLI interacting with AWS services through command line interface.

Diagram

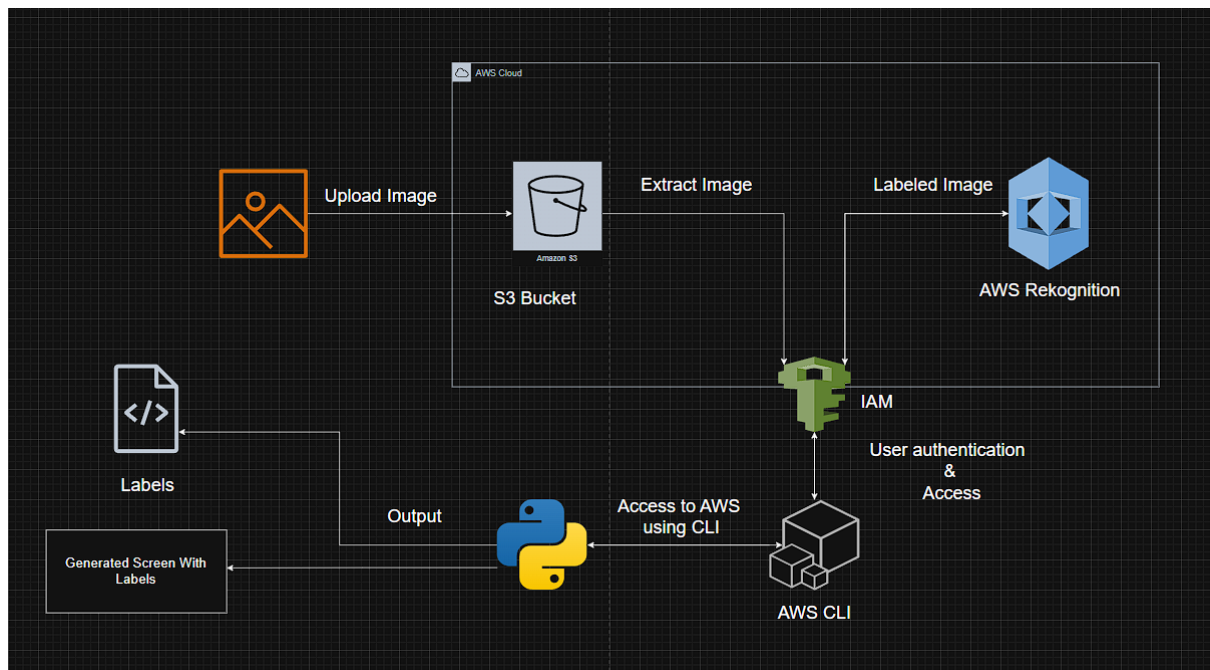


Image Label Generator Using Amazon Rekognition

This screenshot shows the Amazon S3 Buckets console. The 'General purpose buckets' tab is selected, displaying a list of buckets. The bucket 'labeledgen' is highlighted. The console includes a search bar, navigation links, and a sidebar with account information and external access summary.

Name	AWS Region	Creation date
labeledgen	US East (Ohio) us-east-2	August 15, 2025, 21:01:02 (UTC-05:00)

This screenshot shows the details for the 'labeledgen' bucket. The 'Objects' tab is selected, displaying a list of objects. The object 'IMG_1206.jpeg' is highlighted. The console includes a search bar, navigation links, and a sidebar with account information and external access summary.

Name	Type	Last modified	Size	Storage class
IMG_1206.jpeg	jpeg	August 15, 2025, 21:05:22 (UTC-05:00)	917.2 KB	Standard

This screenshot shows the AWS IAM console for the user 'AWScli_imageLabling'. The 'Summary' tab is selected, displaying the user's ARN, console access status, and access keys. The 'Permissions' tab is also visible, showing the user's permissions policies.

Policy name	Type	Attached via
AdministratorAccess	AWS managed - job function	Group AdminAccess

Image Label Generator Using Amazon Rekognition

Access key created
This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

Step 1
Access key best practices & alternatives

Step 2 - optional
Set description tag

Step 3
Retrieve access keys

Retrieve access keys [Info](#)

Access key
If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key | Secret access key

[Download .csv file](#) [Done](#)

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [best practices for managing AWS access keys](#).

```
ImageRekognition.py 2
E:\Studies> AWS> AWS Projects> ImageLabelingUsingRekognition> ImageRekognition.py> detect_labels
1 import boto3
2 import matplotlib.pyplot as plt
3 import matplotlib.patches as patches
4 from PIL import Image
5 from io import BytesIO
6
7 def detect_labels(photo, bucket):
8     # Create a Rekognition client
9     client = boto3.client('rekognition')
10
11     # Detect labels in the photo
12     response = client.detect_labels(
13         Image={'S3Object': {'Bucket': bucket, 'Name': photo}},
14         MaxLabels=10)
15
16     # Print detected labels
17     print('Detected labels for ' + photo)
18     print()
19     for label in response['Labels']:
20         print("Label:", label['Name'])
21         print("Confidence:", label['Confidence'])
22         print()
23
24     # Load the image from S3
25     s3 = boto3.resource('s3')
26     obj = s3.Object(bucket, photo)
27     img_data = obj.get()['Body'].read()
28     img = Image.open(BytesIO(img_data))
29
30     # Display the image with bounding boxes
31     plt.imshow(img)
32     ax = plt.gca()
33     for label in response['Labels']:
34         for instance in label.get('Instances', []):
35             bbox = instance['BoundingBox']
36             left = bbox['Left'] * img.width
37             top = bbox['Top'] * img.height
38             width = bbox['Width'] * img.width
39             height = bbox['Height'] * img.height
40             rect = patches.Rectangle((left, top, width, height, linewidth=1, edgecolor='r', facecolor='none'))
41             ax.add_patch(rect)
42             label_text = label['Name'] + ' (' + str(round(label['Confidence'], 2)) + '%')
43             plt.text(left, top - 2, label_text, color='r', fontsize=8, bbox=dict(facecolor='white', alpha=0.7))
44     plt.show()
45
46     return len(response['Labels'])
47
48 def main():
49     photo = 'images.jpeg'
50     bucket = 'lablegen'
51     label_count = detect_labels(photo, bucket)
52     print("Labels detected:", label_count)
53
54 if __name__ == "__main__":
55     main()
56
57 Amazon S3: For storing the images in the process of generating labels.
```

```
File Edit Selection View Go Run Terminal Help
ImageRekognition.py 2
E:\Studies> AWS> AWS Projects> ImageLabelingUsingRekognition> ImageRekognition.py> detect_labels
7 def detect_labels(photo, bucket):
29
30     # Display the image with bounding boxes
31     plt.imshow(img)
32     ax = plt.gca()
33     for label in response['Labels']:
34         for instance in label.get('Instances', []):
35             bbox = instance['BoundingBox']
36             left = bbox['Left'] * img.width
37             top = bbox['Top'] * img.height
38             width = bbox['Width'] * img.width
39             height = bbox['Height'] * img.height
40             rect = patches.Rectangle((left, top, width, height, linewidth=1, edgecolor='r', facecolor='none'))
41             ax.add_patch(rect)
42             label_text = label['Name'] + ' (' + str(round(label['Confidence'], 2)) + '%')
43             plt.text(left, top - 2, label_text, color='r', fontsize=8, bbox=dict(facecolor='white', alpha=0.7))
44     plt.show()
45
46     return len(response['Labels'])
47
48 def main():
49     photo = 'images.jpeg'
50     bucket = 'lablegen'
51     label_count = detect_labels(photo, bucket)
52     print("Labels detected:", label_count)
53
54 if __name__ == "__main__":
55     main()
56
57 Amazon S3: For storing the images in the process of generating labels.
```

Image Label Generator Using Amazon Rekognition

