

ANALYZING IMAGES WITH AN AWS LAMBDA FUNCTION AND DEPLOY IMAGE IN S3 BUCKET.

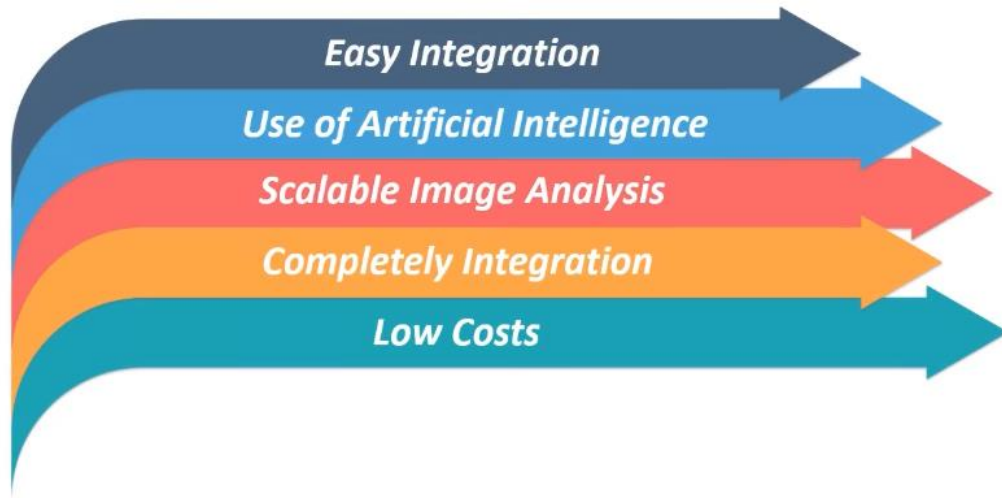
what is aws Rekognition?

Rekognition is one of the AWS services to perform image and video analysis. So here all we need to provide is the image or video to the AWS Rekognition service and it will help us to identify an object, people, text, activities, and scenes.



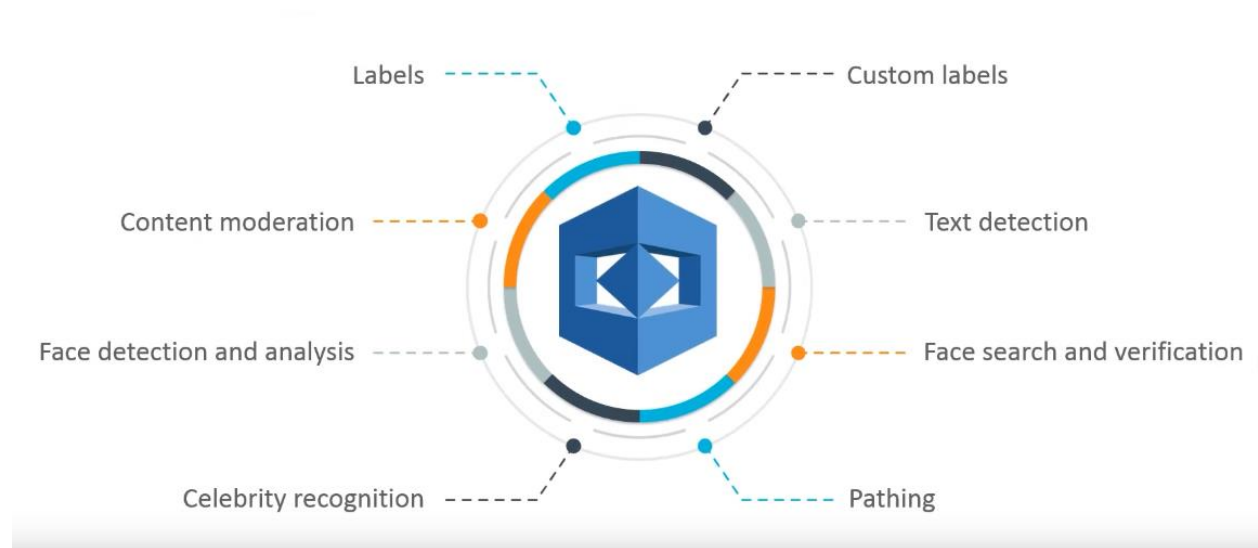
Amazon Rekognition makes it easy to add image and video analysis to your applications using proven, highly scalable, deep learning technology that requires no machine learning expertise to use

Benefits of using Amazon Rekognition are as follows:



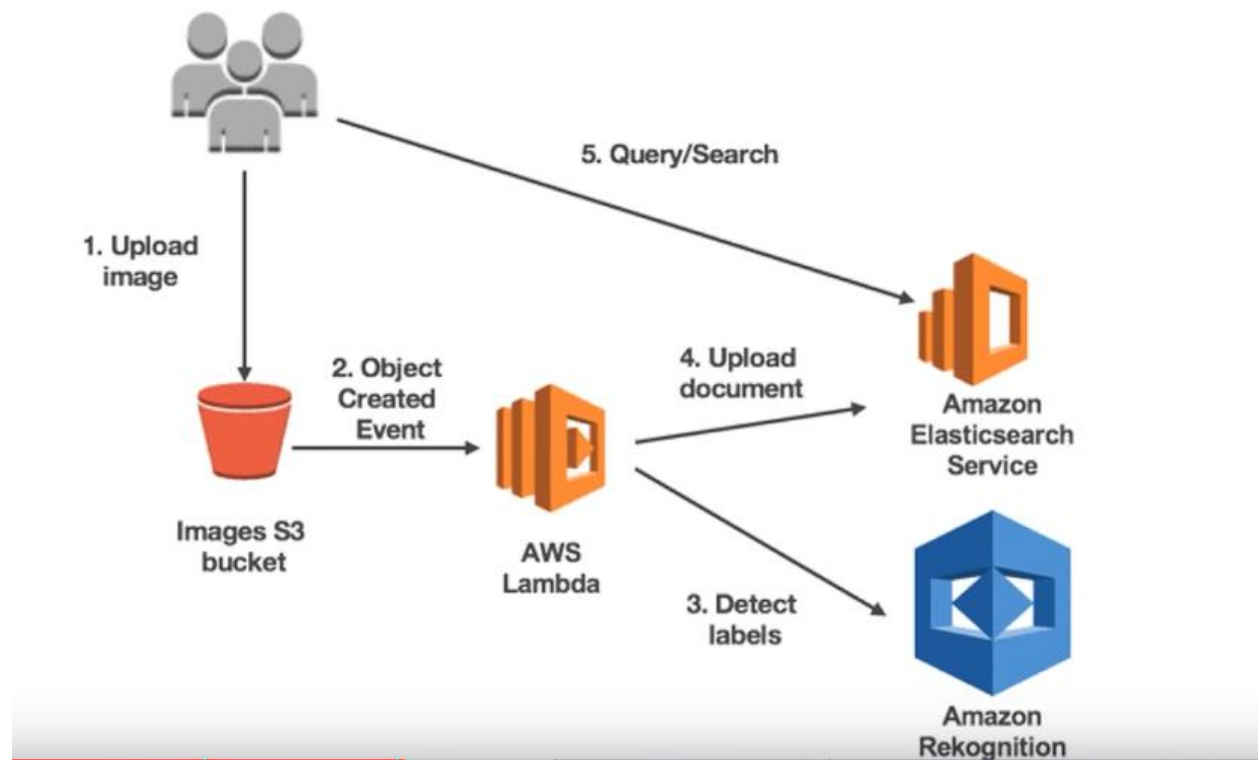
1. Integrating powerful image and video analysis into your apps.
2. Deep learning-based image and video analysis.
3. Scalable image analysis.
4. Integration with other AWS services.
5. Low cost

Common use cases for using Amazon Rekognition mentioned in the following:



1. Searchable image and video libraries
2. Face-based user verification
3. Sentiment and demographic analysis
4. Facial Search
5. Unsafe content detection
6. Celebrity recognition
7. Text detection
8. Custom labels

How Amazon Rekognition works



For Image analysis, we are using four services of AWS.

1. IAM
2. S3
3. Lambda
4. Rekognition (This service we are using within the lambda function).

So, the flow for image analysis will be:

- Firstly, we are going to read an image from the S3 bucket via a lambda function.
- And in the second step we will pass that image to rekognition service via calling rekognition API. In response to this, rekognition API will return labels.

Step 1: Creating an IAM role:

- Go to the AWS Management console.
- Search for the **IAM** service and enter.
- In the IAM service on the left side click on **Roles** In that click on **Create Role** button.
- Select the type of trusted entity as an **AWS service** by default.
- In Choose a use case selects **Lambda** and then click on **Next: Permission** button.
- In the Attach permissions policies select two policies:

Create role

1234

▼ Attach permissions policies

Choose one or more policies to attach to your new role.


Create policy

↺

Filter policies ▼

Q AWSLambdaExecute

Showing 1 result

	Policy name ▼	Used as
<input checked="" type="checkbox"/>	▶  AWSLambdaExecute	Permissions policy (1)

* Required

Cancel

Previous

Next: Tags

Create role

1

2

3

4

▼ Attach permissions policies





Choose one or more policies to attach to your new role.

[Create policy](#)

Filter policies ▾

Q Reko

Showing 4 results

	Policy name ▾	Used as
<input type="checkbox"/>	 AmazonRekognitionCustomLabelsFullAccess	None
<input checked="" type="checkbox"/>	 AmazonRekognitionFullAccess	Permissions policy (1)
<input type="checkbox"/>	 AmazonRekognitionReadOnlyAccess	None
<input type="checkbox"/>	 AmazonRekognitionServiceRole	None

* Required

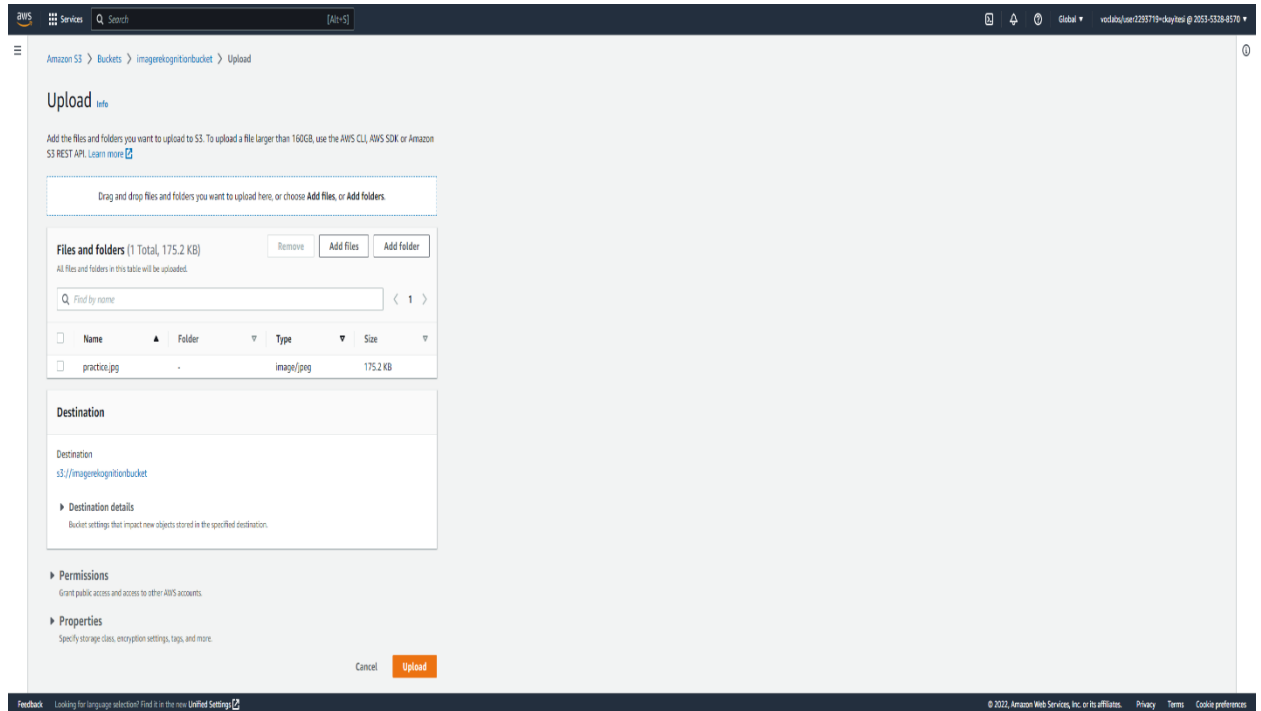
[Cancel](#)[Previous](#)[Next: Tags](#)

- Click on **Next: Tags** button.
- Add tags part is optional so click on **Next: Review** button.
- Give a **name** to your role. You can give any name to your role[for eg.lamda_rekognition]and click on Create **role button**.
- Your role is ready.

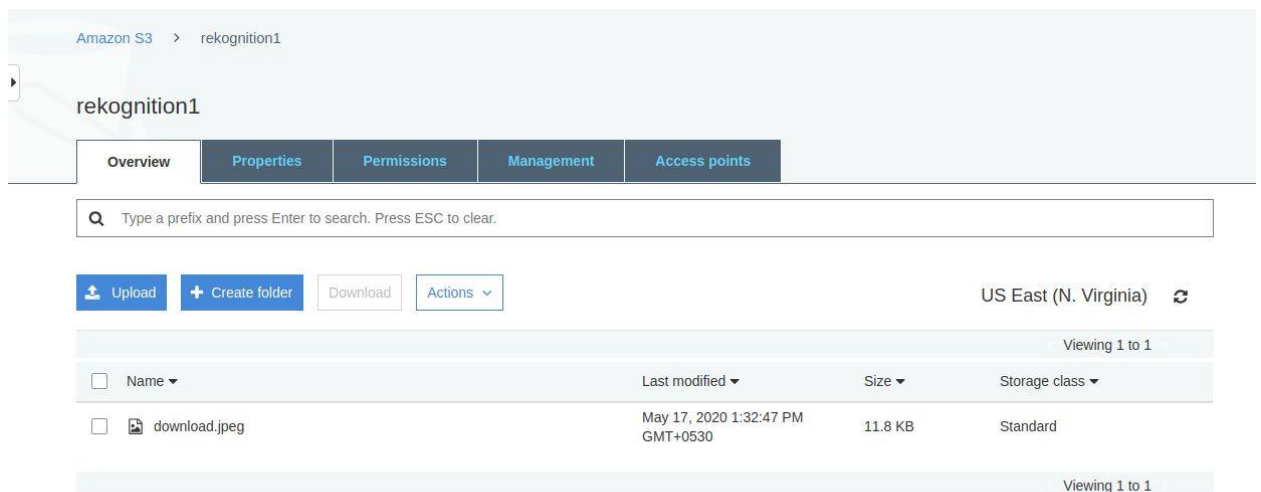
Step 2: Create an S3 bucket to store images:

- Go to the AWS Management console.
- Search for the **S3** service and enter.
- Click on the **Create bucket** button.
- Give any unique **name** to you bucket [for eg: rekognition].
- Keep all default settings as it is and click on the **Create bucket** button.

- Once your bucket is created **click on your bucket name**. In that click on the **upload button** and drag and drop any image that you want and click on the upload button directly. Once the image is uploaded you can see the image as follows



or



This is the policies I used:

Policies:

- Version: "2012-10-17"

Statement:

- Effect: Allow

Action:

- s3:GetObject

Resource: arn:aws:s3:::*

- Effect: Allow

Action:

- rekognition:DetectText

- rekognition:DetectLabels

Resource: "*" *

For the image analysis, I uploaded the following image.



Step 3: Create a Lambda Function:

- Go to the AWS Management console.
- Search for the **Lambda** service and enter.
- After coming onto the lambda service page click on **Create function** button.
- Select the **Author from Scratch** default option.
- For **function name** give any name of your choice[for eg lamda_rekognition].
- In the **Runtime** select **python 3.7**

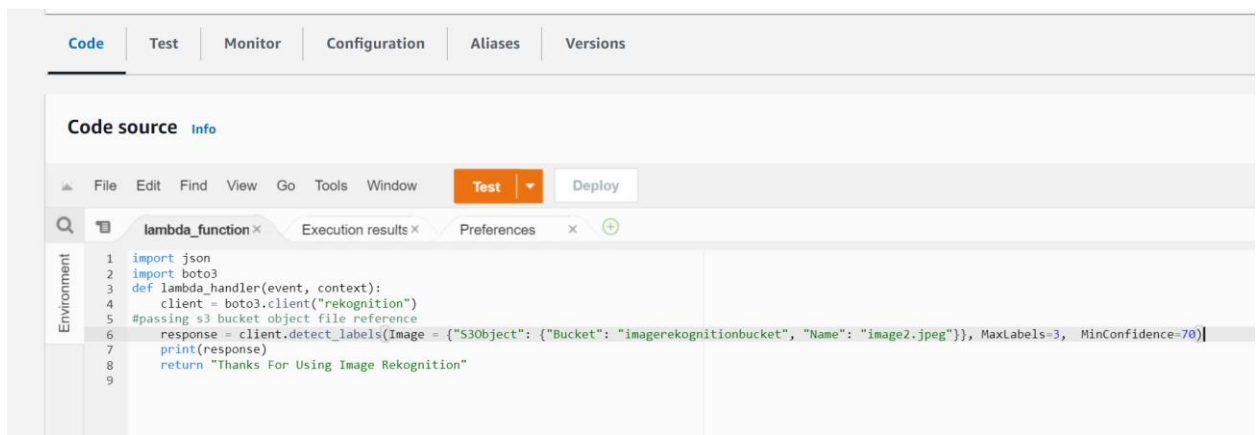
- Expand the **Choose or create an execution role.**
- In that select **Use an existing role.** And in the existing role select the role that we created in our first step[I have given the name for a role is lamda_rekognition].
- Finally, click on **create function** button.

- Once you created a lambda function then click on your function name.
- In the function, code editor type the function that I have given in the following:
- In the following code, you can directly pass the S3 references in the response using rekognition client and you will get a response.”**MaxLables=3**” term is optional using this you can able to see only three labels for the image if we did not mention the name you get more label names for your images

*Note: In place of **buket_name** and **image_name** plz. mentioned your S3 bucket name and uploaded image name.*

```
import json
import boto3

def lambda_handler(event, context):
    client = boto3.client("rekognition")
    #passing s3 bucket object file reference
    response = client.detect_labels(Image = {"S3Object": {"Bucket":
    "bucket_name", "Name": "image_name"}}}, MaxLabels=3,
    MinConfidence=70)
    print(response)
    return "Thanks"
```



- If you want to pass the byte data in the function then also you can pass then prefer the following code.

```

import json
import boto3
def lambda_handler(event, context):
    client = boto3.client("rekognition")
    s3 = boto3.client("s3")
    # reading file from s3 bucket and passing it as bytes
    fileObj = s3.get_object(Bucket = "bucket_name", Key="image_name")
    file_content = fileObj["Body"].read()
    # passing bytes data
    response = client.detect_labels(Image = {"Bytes": file_content},
    MaxLabels=3, MinConfidence=70)
    print(response)
    return "Thanks"

```

- You can add any one of the codes that I have given above you will get the same response. Just I have shown you two different kinds of code for the lambda function.
- After adding the code click on the **save** button.
- To test the function for image analysis. Click on the **Test** button given on the upper right side. Once you click on the test button it will pop up one window.
- In that select **Create new test event** and in Event Template give any name that you want.

Configure test event

A function can have up to 10 test events. The events are persisted so you can switch to another computer or web browser and test your function with the same events.

☒ Create new test event

☐ Edit saved test events

Event template

mydemo

Event name

MyEventName

```
1 {  
2   "key1": "value1",  
3   "key2": "value2",  
4   "key3": "value3",  
5 }
```

- And finally, click on the **Create** button.
- Once you create the test event it will show you the test event name. So now click on the **Test** button.

Lambda > Functions > imageRekogFnc

imageRekogFnc

Throttle Copy ARN Actions

Function overview Info

imageRekogFnc

Layers (0)

+ Add trigger

+ Add destination

Description-

Last modified28 minutes ago

Function ARNarn:aws:lambda:us-east-1:205353288570:function:imageRekogFnc

Function URL Info-

Code

Test

Monitor

Configuration

Aliases

Versions

Code source Info

Upload from

File Edit Find View Go Tools Window

Test

Deploy

Settings

- Once you click on the test the response will look like below:

```
1 Response
2 "Thanks For Using Image Rekognition"
3
4 Function Logs
5 START RequestId: 6793c75b-f0f6-4432-af7f-b5ffb1e00528 Version: $LATEST
6 {'Labels': [{'Name': 'Reading', 'Confidence': 99.96436309814453,
7   'Instances': [], 'Parents': [{'Name': 'Person'}]},
8   {'Name': 'Person', 'Confidence': 99.96436309814453,
9     'Instances': [{'BoundingBox': {'Width': 0.560932993888855,
10      'Height': 0.714285525016785, 'Left': 0.2540991008281708,
11      'Top': 0.23771491646766663}, 'Confidence': 98.81011199951172}], 'Parents': []},
12     {'Name': 'Lamp', 'Confidence': 99.93045806884766, 'Instances': [],
13       'Parents': []}], 'LabelModelVersion': '3.0', 'ResponseMetadata':
14     {'RequestId': 'd1697289-5c18-41d0-a59e-9b4d8a9289c5', 'HTTPStatusCode': 200,
15       'HTTPHeaders': {'x-amzn-requestid': 'd1697289-5c18-41d0-a59e-9b4d8a9289c5',
16        'content-type': 'application/x-amz-json-1.1', 'content-length': '625',
17        'date': 'Sat, 17 Dec 2022 19:07:05 GMT'}, 'RetryAttempts': 0}}
18 END RequestId: 6793c75b-f0f6-4432-af7f-b5ffb1e00528
19 REPORT RequestId: 6793c75b-f0f6-4432-af7f-b5ffb1e00528 Duration: 2197.68 ms
20 Billed Duration: 2198 ms Memory Size: 128 MB Max Memory Used: 70 MB Init Duration: 323.39 ms
21
22 Request ID
23 6793c75b-f0f6-4432-af7f-b5ffb1e00528
```

What exactly you will get in response:

This rekognition API takes individual images as input and returns an ordered list of labels and a corresponding numeric confidence index. As you can see, I have uploaded a person image with a car. So, in response, you can see you will get all kinds of labels like a person, human, vehicle, car, and it also returns bounding boxes coordinates for items that are detected in images (for Eg: height and width of persons face).

The response you will get is based on the image that you upload in the S3 bucket.

Try adding different images in the S3 bucket and check the image analysis.

I hope you like the blog.

