Problem Statement: Create a lambda function which will analyze the image stored in S3 bucket and return matching results using the amazon service Rekognition.

Scenario 1: The **first scenario** is just uploading the images and test the code manually

Scenario 2: Automatically detect labels from images uploaded to a specific S3 **input folder** and write the label results to a **CSV file** in an **output folder**.

Amazon Service:

1. Amazon Rekognition

Pre-requisites:

1. Amazon Free Tier Account.
2. Knowledge on how to create IAM users and add permissions.
3. Python Code
4. Image in S3 bucket

Architecture Overview:

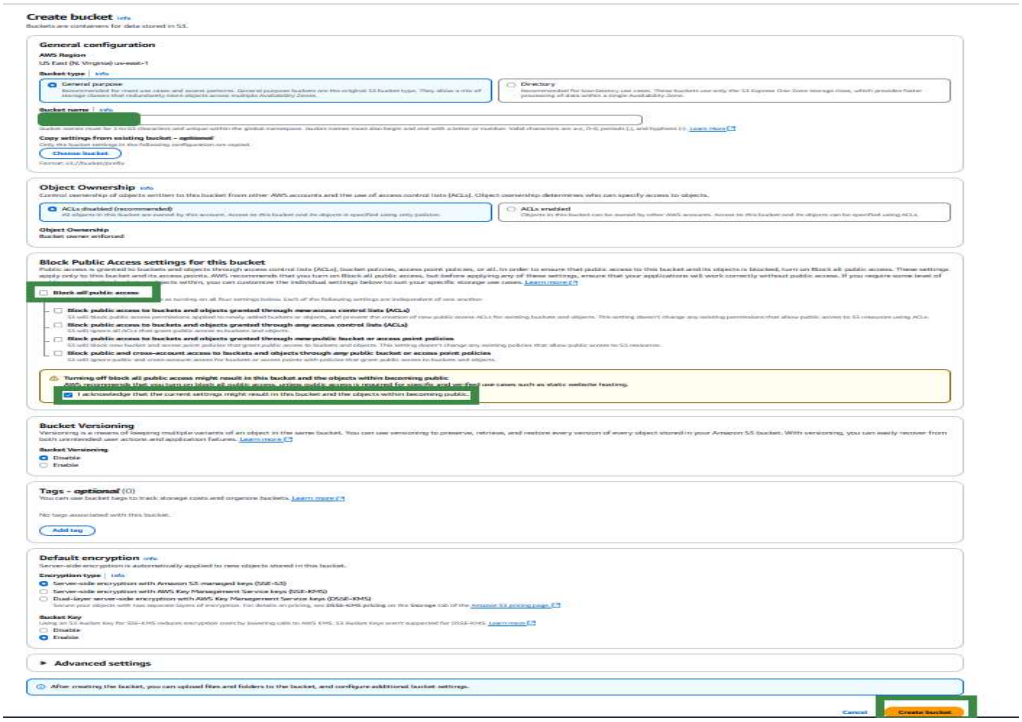
1. S3 Bucket stores the image
    a. Input Folder Name – input
    b. Output Folder Name - output
2. Lambda function is triggered with the image details (manually)
3. Amazon Rekognition to analyze the image
4. Output is a list of detected labels in csv format

Steps:

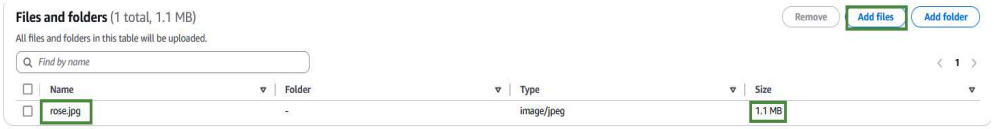
1. Create an S3 bucket
2. Select General Purpose
3. Provide a unique bucket name aws_rekognition
4. Object ownership – Select ACLs disabled
5. Uncheck Block public access settings for this bucket
6. Check the box “I acknowledge”
7. Leave other settings as default
8. Click on create bucket
9. Will use .jpg or .jpeg images for the demo

10. Upload the rose image into the bucket
11. The image will be available in the bucket
12. Create a role with the following permissions and name the role
    a. **AmazonRekognitionFullAccess**
    b. **AmazonS3ReadOnlyAccess**
13. After IAM role creation, create a lambda function
14. Click on Create a function
15. Select Author from Scratch
16. Enter a function name
17. Select runtime as Python
18. Click on Use an Execution Role
19. Select the IAM Role Created
20. Leave other settings as default
21. Click on Create Function
22. In the code tab, paste the python code
23. Refer the below screenshots

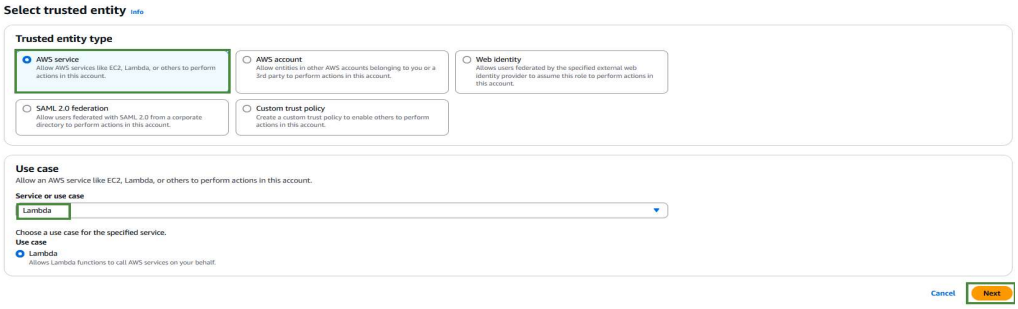
S3 Bucket Creation

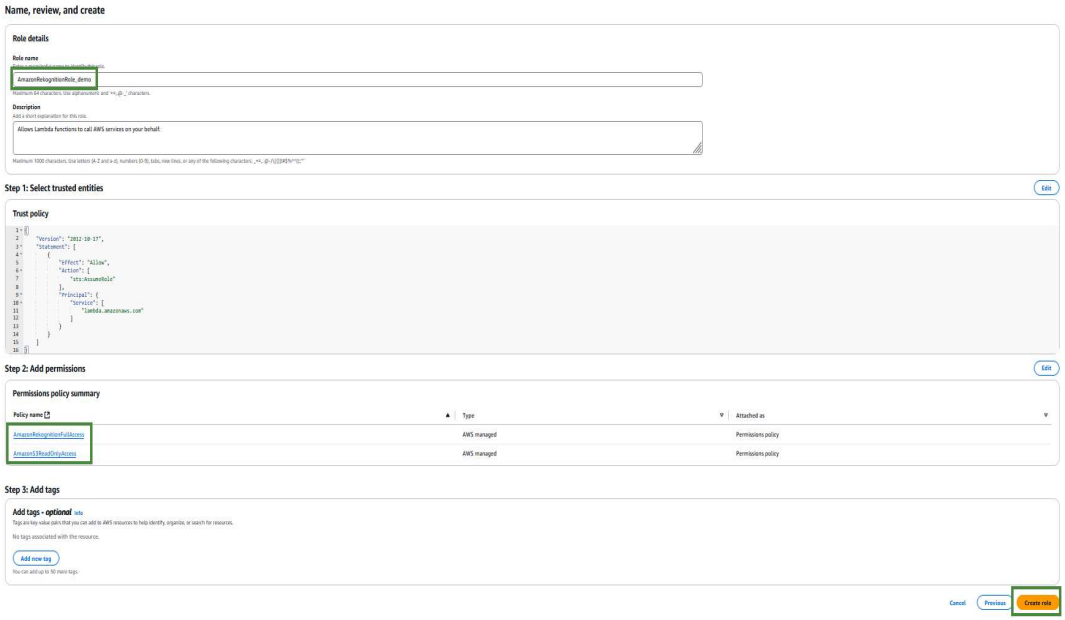


S3 Bucket – Image Upload

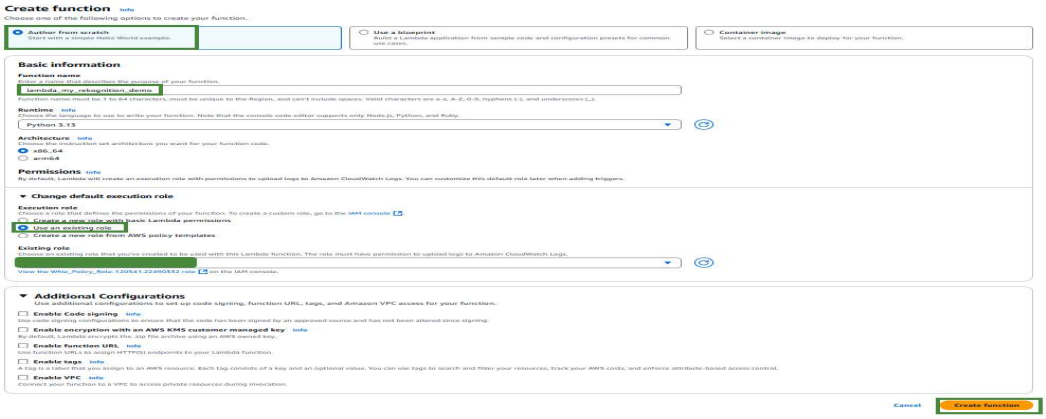


IAM Role Creation – Lambda Function





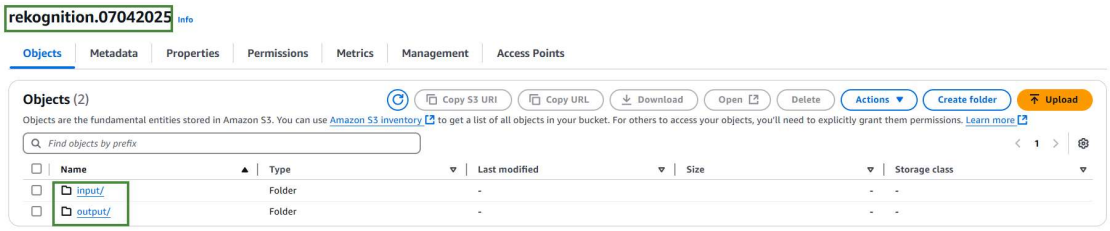
Create a Lambda Function

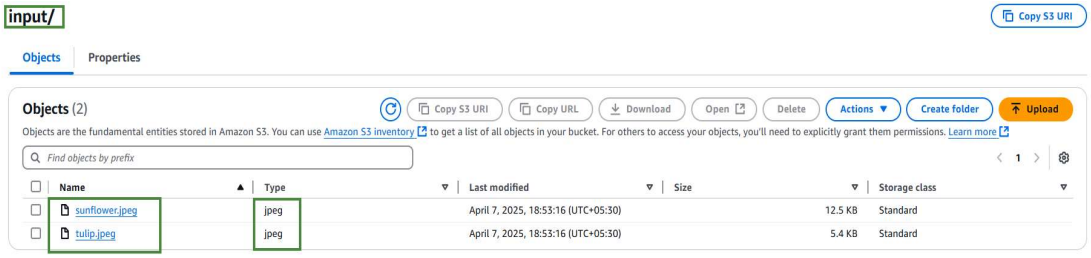


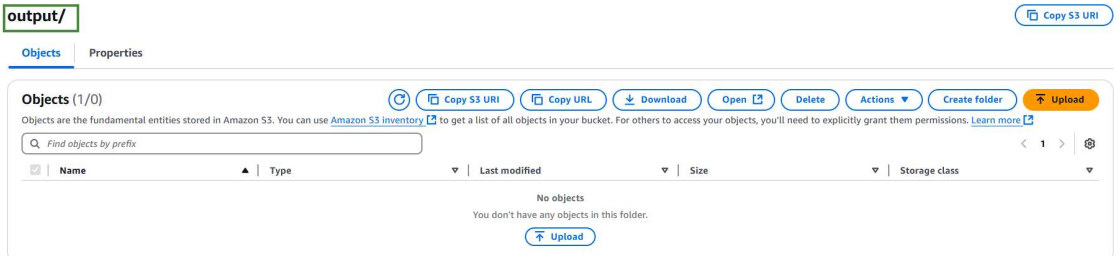
Scenario 1 - Python Code – Upload the code and click test the output

```
import json # module is used to work with JSON data

import boto3 # Python SDK - allows Python to talk to AWS Services such as S3, Lambda

def lambda_handler(event, context):  # function manually triggered


  bucket_name = "Enter your bucket name"  # Enter the bucket name

  image_obj_name = "rose.jpeg"  # Enter the image name along with the extension exact filename / This is manual method


  try:

    rkClient = boto3.client("rekognition", region_name="Enter your region")  # Creates a connection with the AWS Service and calls the service to analyze the image and detect objects

    try:

      rkResponse = rkClient.detect_labels(

        Image={

          'S3Object': {

            'Bucket': bucket_name,

            'Name': image_obj_name

          }

        },

      )

      print(rkResponse['Labels'])  # contains the list of things the services see such as Flower, Nature , Plant

      return rkResponse['Labels']  # Prints the labels

    except Exception as e:

      print("Get labels failed because ", e)

  except Exception as e:

    print("Client connection to Rekognition failed because ", e)
```

Scenario 2: Automatically detect labels from images uploaded to a specific S3 **input folder** and write the label results to a **CSV file** in an **output folder**.







Paste the following code in the lambda code editor and deploy the code

```
import boto3

import csv

from io import StringIO


# Initialize AWS clients

s3 = boto3.client('s3')

rekognition = boto3.client('rekognition')


def lambda_handler(event, context):

  # 🎯 Hardcoded values for simplicity

  bucket = "rekognition.07042025"       # Replace with your actual bucket name
```

```
input_folder = "input/"
output_key = "output/labels.csv"

print("Bucket:", bucket)
print("Input folder:", input_folder)
print("Output file:", output_key)

# Step 1: List image files in the input folder
image_keys = []
response = s3.list_objects_v2(Bucket=bucket, Prefix=input_folder)

if 'Contents' not in response:
  return {
    'statusCode': 404,
    'body': 'No files found in input folder.'
  }

for obj in response['Contents']:
  key = obj['Key']
  if key.endswith(('.jpg', '.jpeg', '.png')) and not key.endswith('/'):
    image_keys.append(key)

if not image_keys:
  return {
    'statusCode': 404,
    'body': 'No supported image files found in input folder.'
  }

print(f"Found {len(image_keys)} image(s): {image_keys}")

# Step 2: Prepare CSV output in memory
csv_buffer = StringIO()
writer = csv.writer(csv_buffer)
writer.writerow(['Image Name', 'Label', 'Confidence (%)'])

# Step 3: Detect labels for each image using Rekognition
for image_key in image_keys:
  try:
    rekog_response = rekognition.detect_labels(
      Image={'S3Object': {'Bucket': bucket, 'Name': image_key}},
      MaxLabels=10,
      MinConfidence=75
```

```
        )
        for label in rekog_response['Labels']:
            writer.writerow([
                image_key.split('/')[-1],
                label['Name'],
                round(label['Confidence'], 2)
            ])
    except Exception as e:
        print(f"Error processing {image_key}: {str(e)}")

# Step 4: Upload CSV to the output folder
try:
    s3.put_object(
        Bucket=bucket,
        Key=output_key,
        Body=csv_buffer.getvalue(),
        ContentType='text/csv'
    )
    print(f"CSV uploaded to s3://{bucket}/{output_key}")
except Exception as e:
    print(f"Error uploading CSV: {str(e)}")
    return {
        'statusCode': 500,
        'body': f"Failed to upload CSV: {str(e)}"
    }

return {
    'statusCode': 200,
    'body': f"CSV file with labels saved at s3://{bucket}/{output_key}"
}
```

After deploying, test the code and in the output folder a csv file will be created.

Event JSON

```
{
  "bucket": "rekognition.07042025",
  "input_folder": "input",
  "output_key": "output/labels.csv"
}
```

Executing function: succeeded (logs)

▼ Details

```
{
  "statusCode": 200,
  "body": "CSV file with labels saved at s3://rekognition.07042025/output/labels.csv"
}
```

Objects (1)

Copy S3 URI | Copy URL | Download | Open | Delete | Actions | Create folder | Upload

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Find objects by prefix

Name	Type	Last modified	Size	Storage class
labels.csv	csv	April 7, 2025, 19:14:38 (UTC+05:30)	266.0 B	Standard

A	B	C	D
Image Nan	Label	Confidence (%)	
sunflower.	Flower	100	
sunflower.	Plant	100	
sunflower.	Sunflower	100	
tulip.jpeg	Flower	100	
tulip.jpeg	Petal	100	
tulip.jpeg	Plant	100	
tulip.jpeg	Tulip	98.47	
tulip.jpeg	Food	94.9	
tulip.jpeg	Ketchup	94.9	