Project Name: Image Analysis using AWS Rekognition via Lambda Function and S3 bucket

Description: The goal of this project is to build a serverless image processing system using AWS Lambda, Amazon S3, and Amazon Rekognition. The system will automatically label detection and object detection for images uploaded to an S3 bucket.

By completing this project, you will have implemented a serverless image processing system that automatically analyzes and tags images using AWS Lambda, Amazon S3, and Amazon Rekognition. This project demonstrates your skills in serverless architecture, event-driven computing, and image processing using AWS services.

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.

Benefits of using Amazon Rekognition are as follows:

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

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

- Searchable image and video libraries
- Face-based user verification
- Sentiment and demographic analysis
- Facial Search
- Unsafe content detection
- Celebrity recognition
- Text detection
- Custom labels

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

- IAM
- S3
- Lambda
- Rekognition

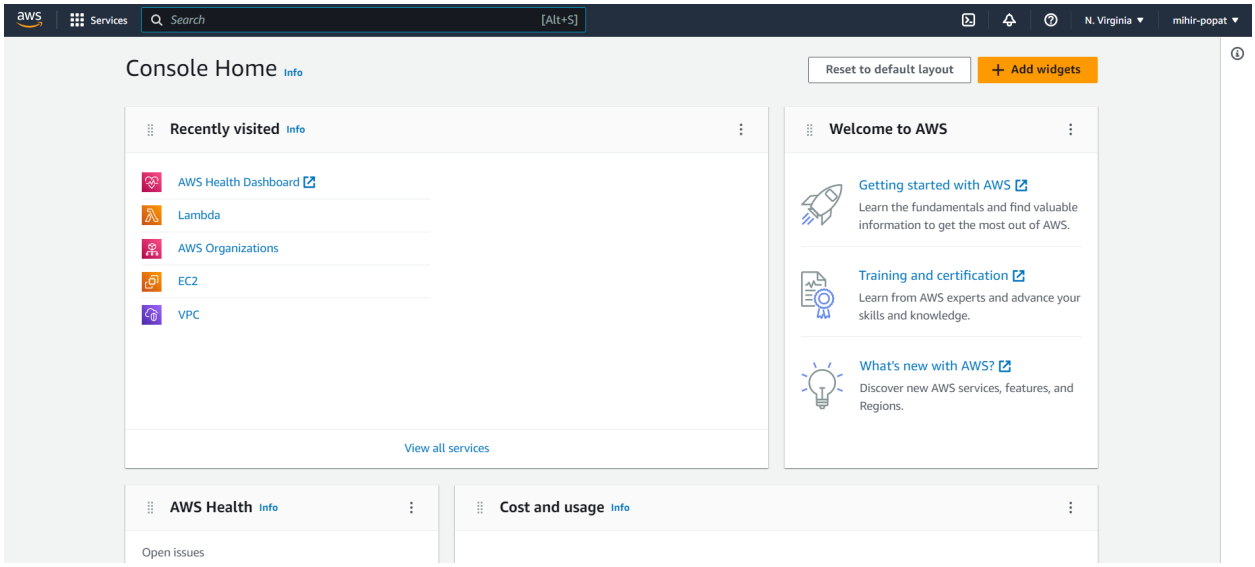
Flow for image analysis will be

- 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.

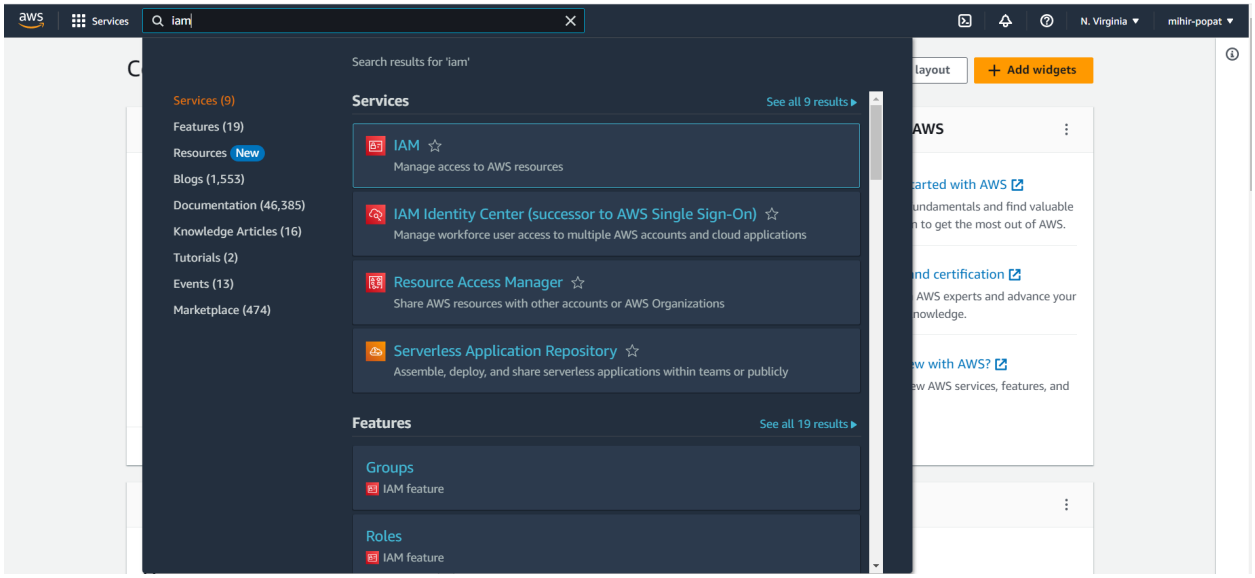
Implementation Steps:

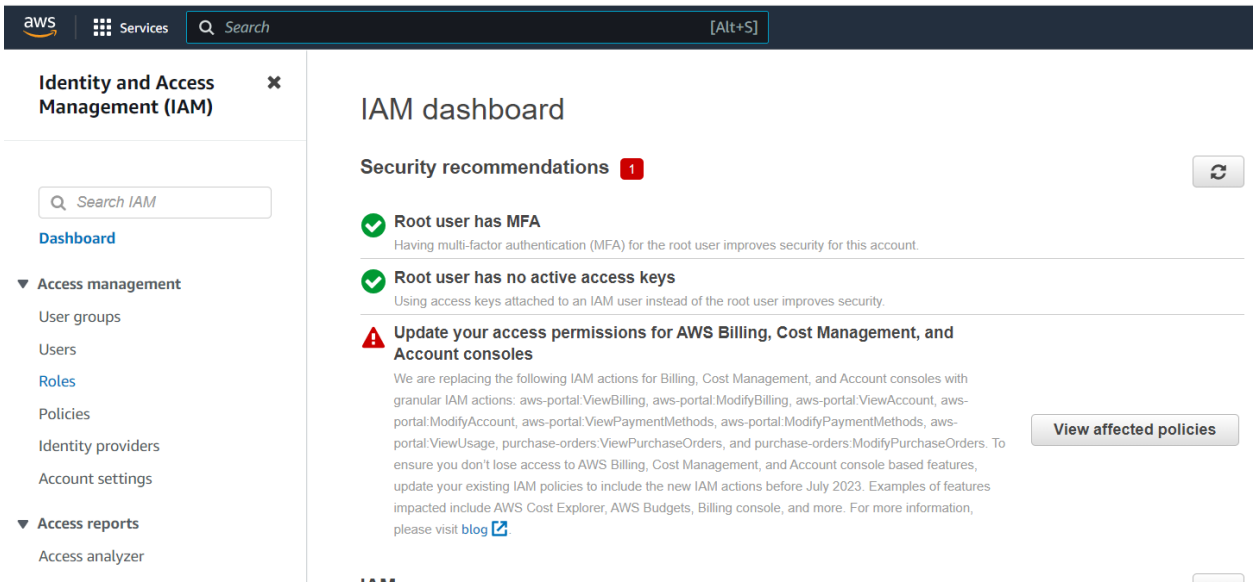
Step 1: Creating an IAM role:

- Go to the AWS Management console.

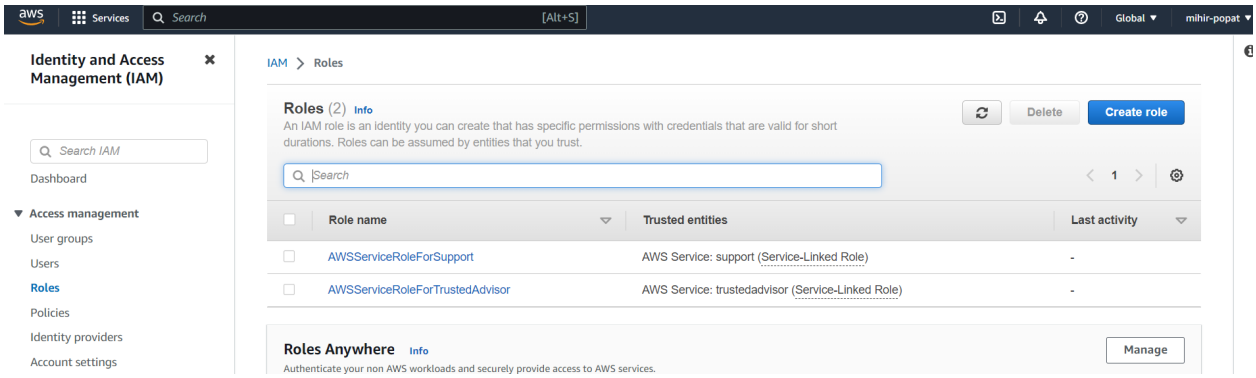


- Search for the IAM service and enter.

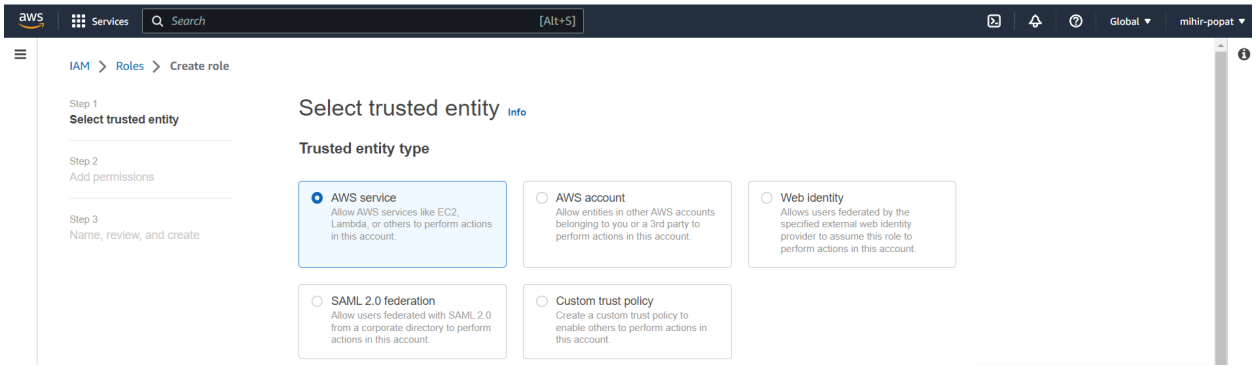




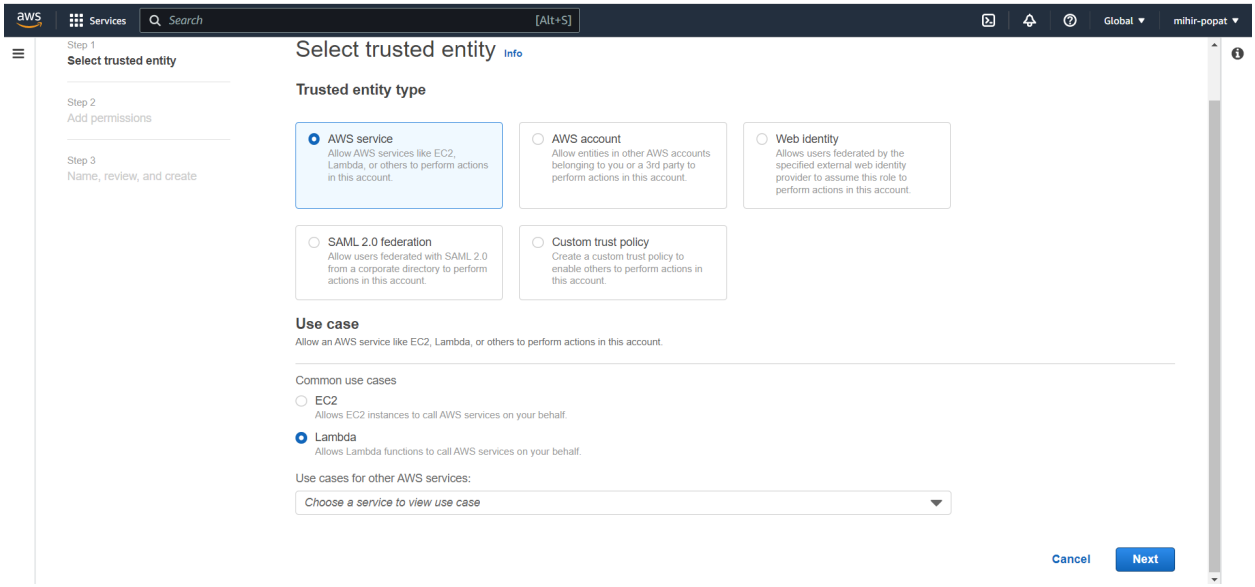
- In the IAM service on the left side click on Roles In that click on Create Role button.



- Select the type of trusted entity as an AWS service by default.

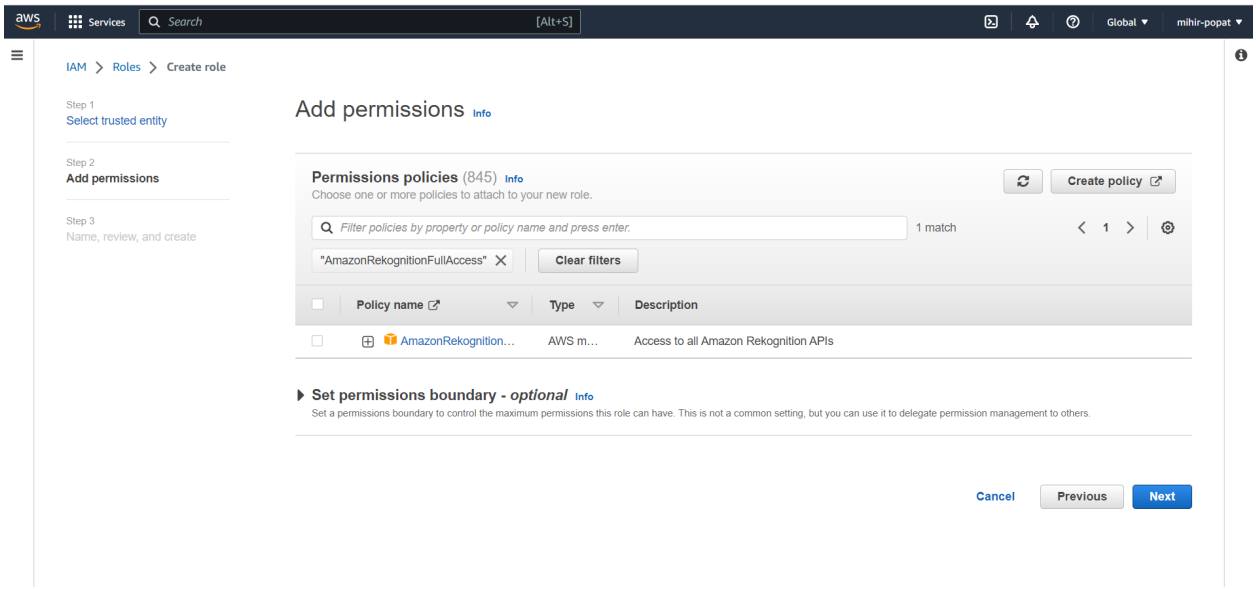


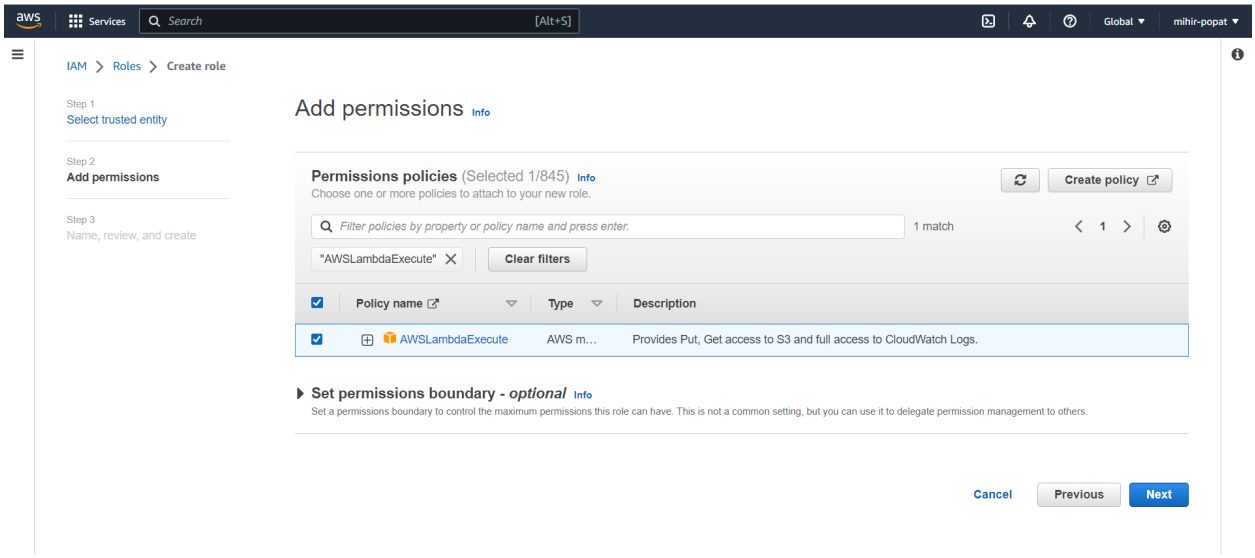
- In Choose a use case select Lambda and then click on Next: Permission button.



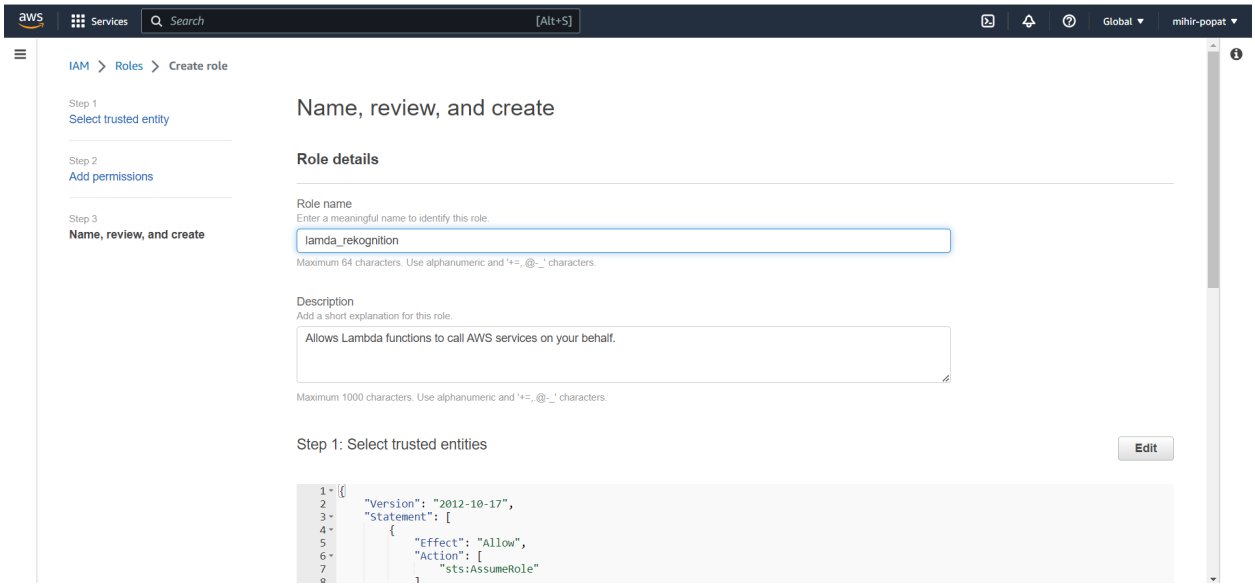
- In the Attach permissions policies select two policies :

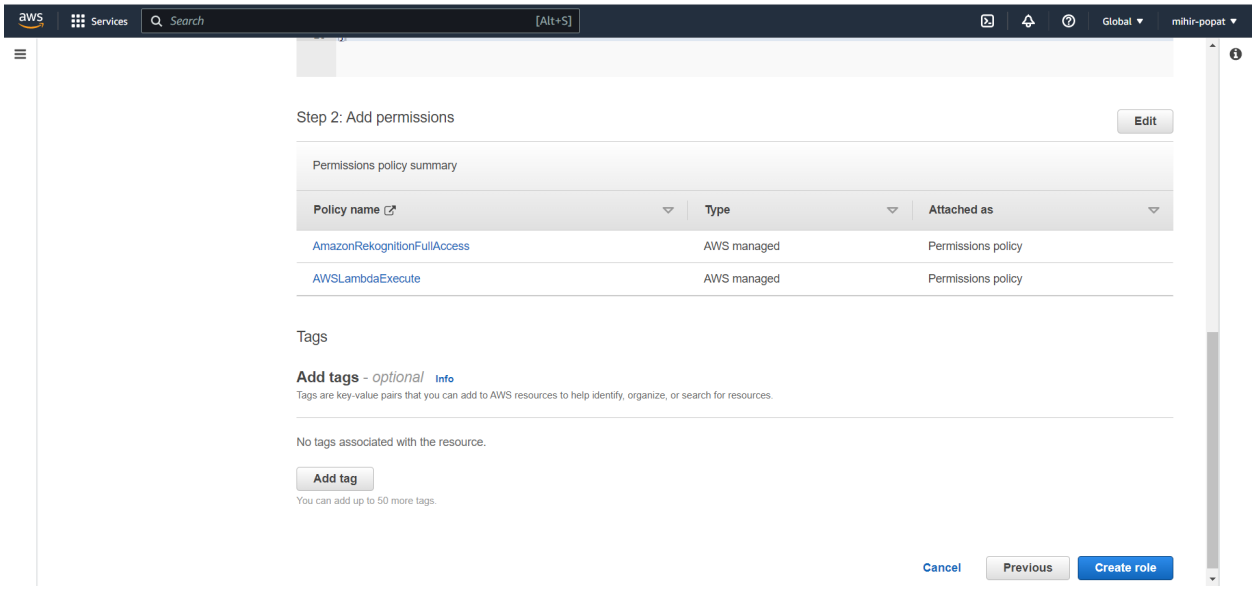
- **AmazonRekognitionFullAccess**
- **AWSLambdaExecute**



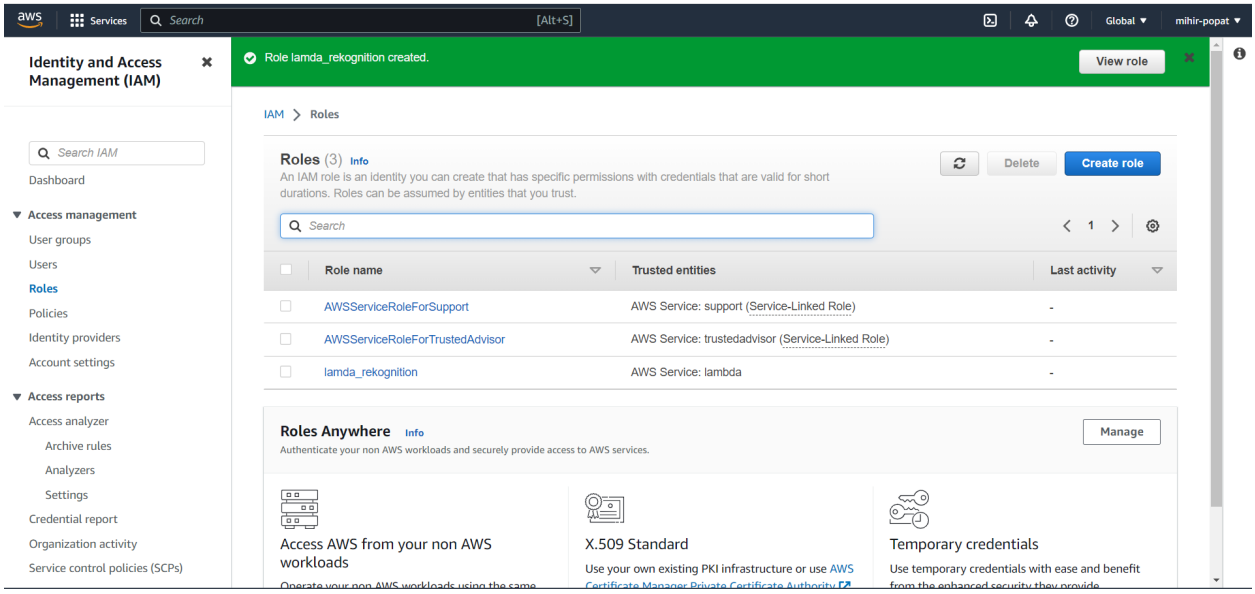


- Click on Next: Tags button.
- Add tags part is optional so click on Next: Review button.
- Give a name to your role. You can give any name to your role [for eg.lamda_rekognition] and click on the Create role button.



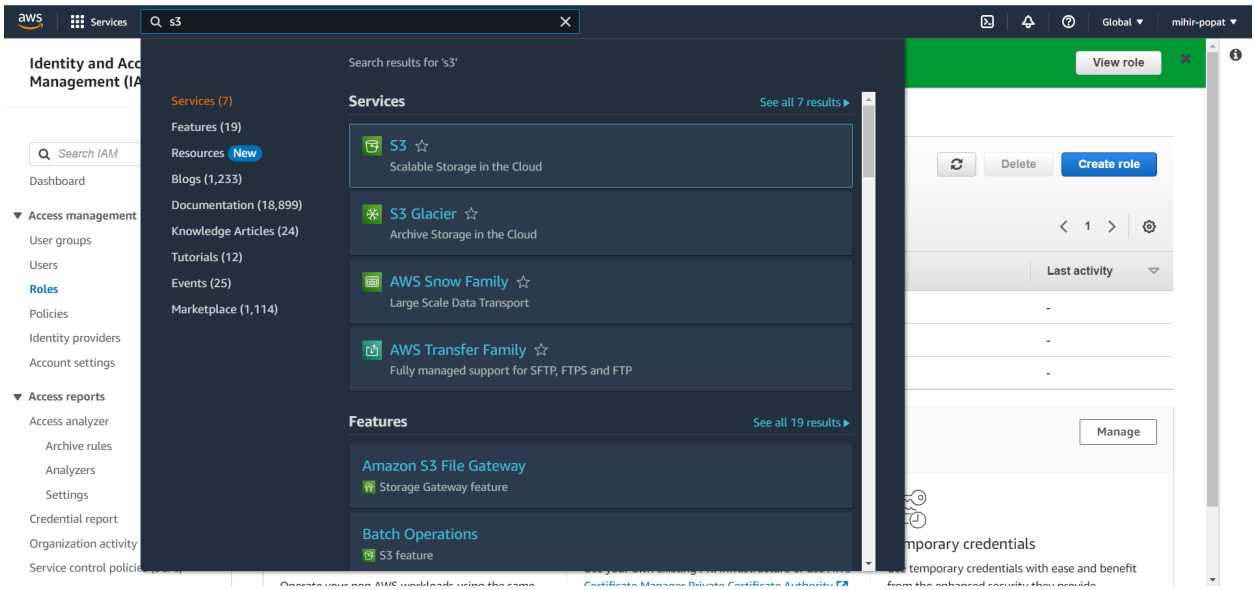


- Your role is ready.

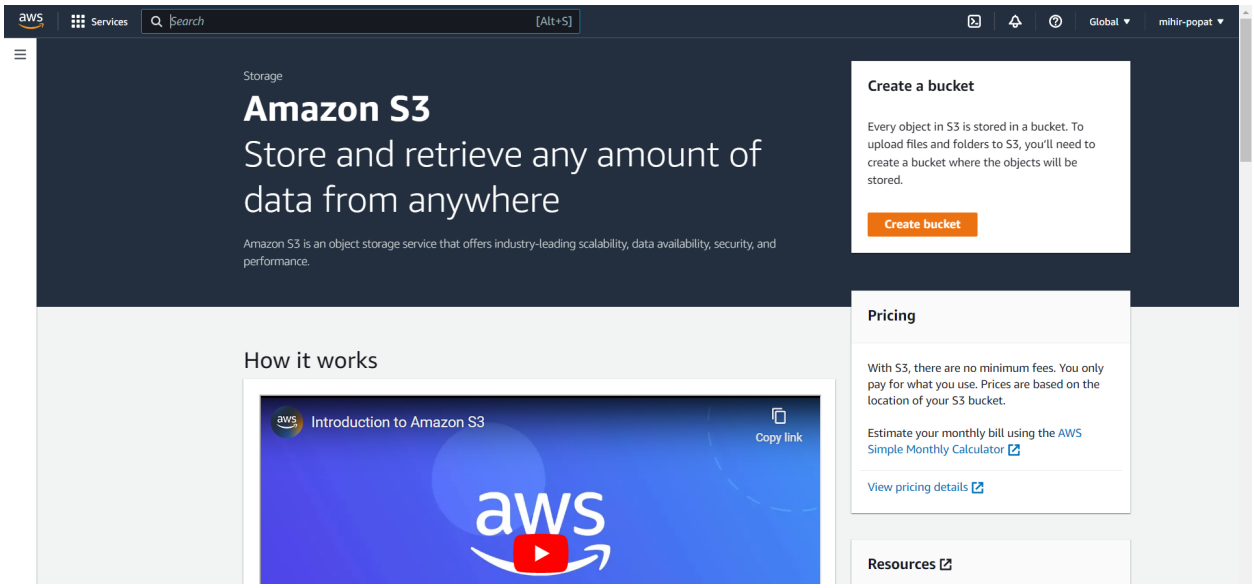


Step 2: Create an S3 bucket to store images:

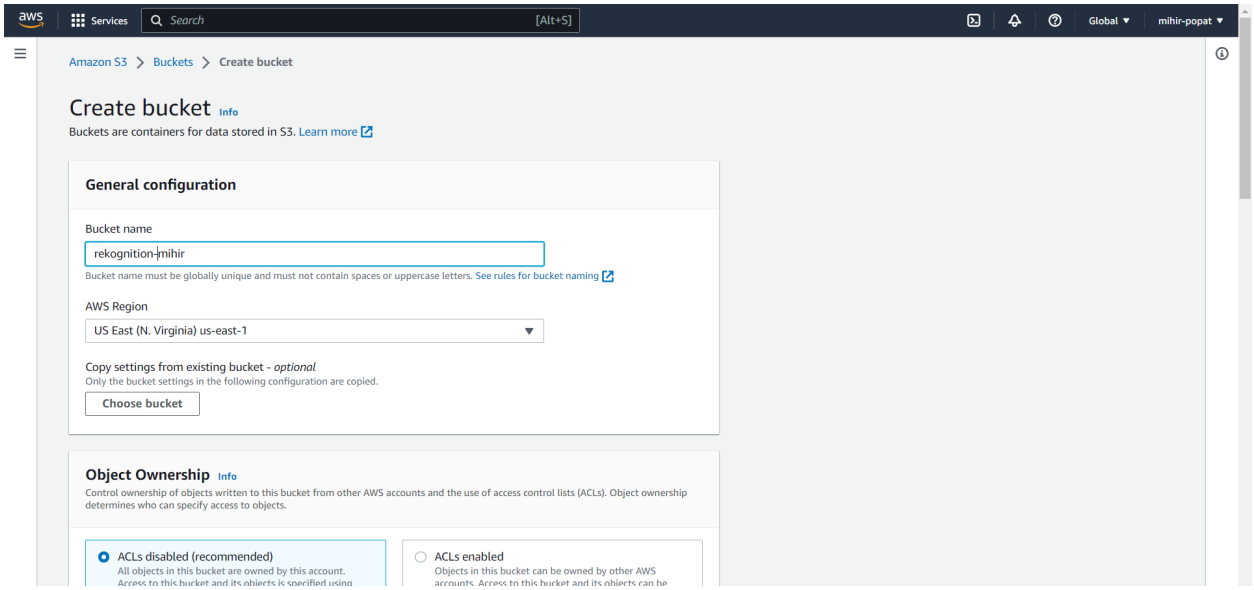
- Go to the AWS Management console.
- Search for the S3 service and enter.

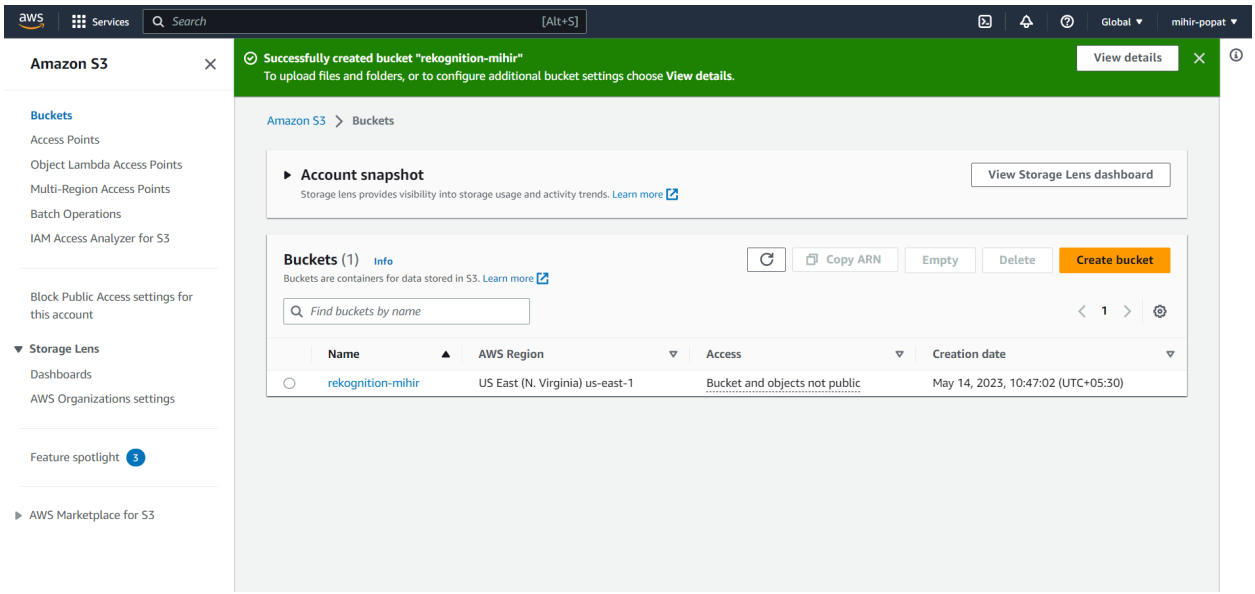


- Click on the Create bucket button.

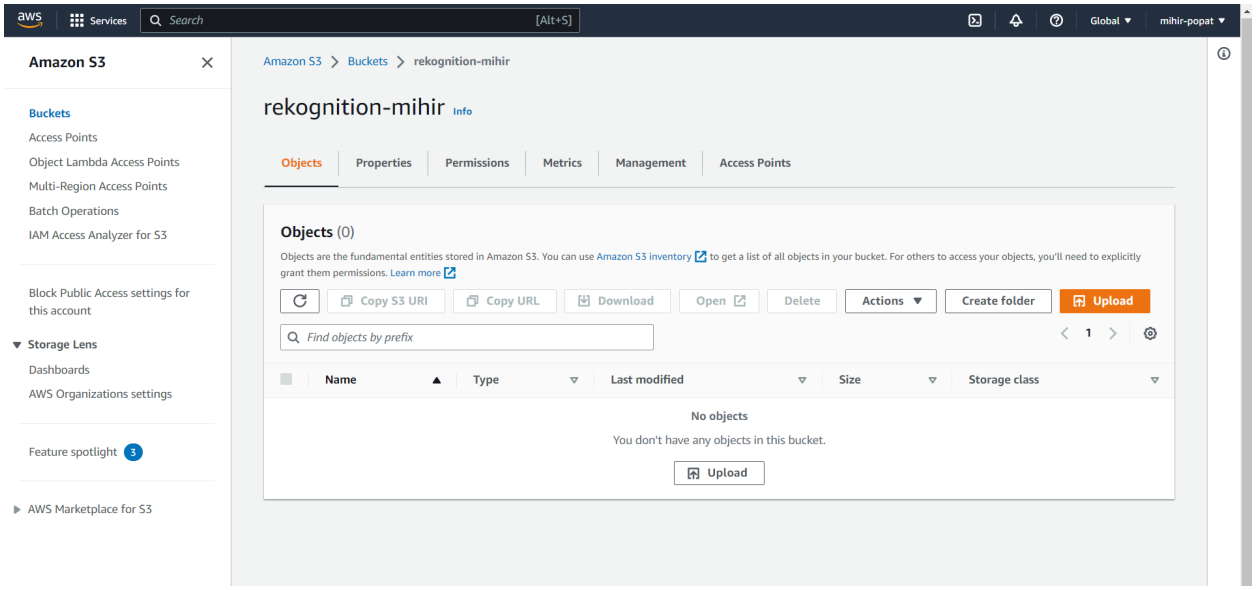


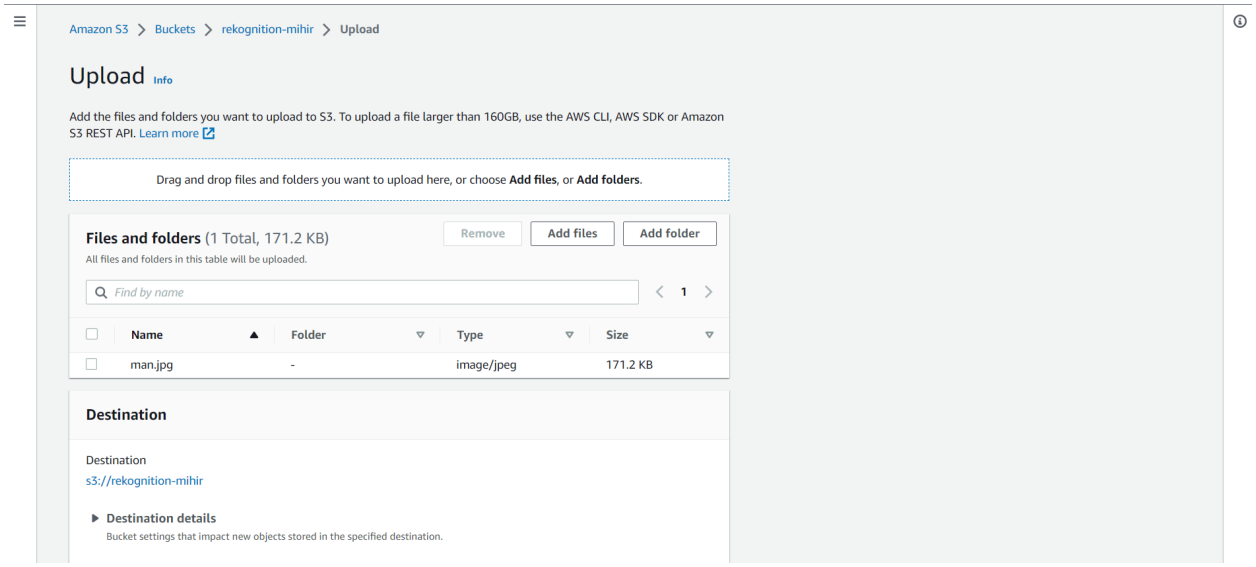
- Give any unique name to you bucket [for eg rekognition].

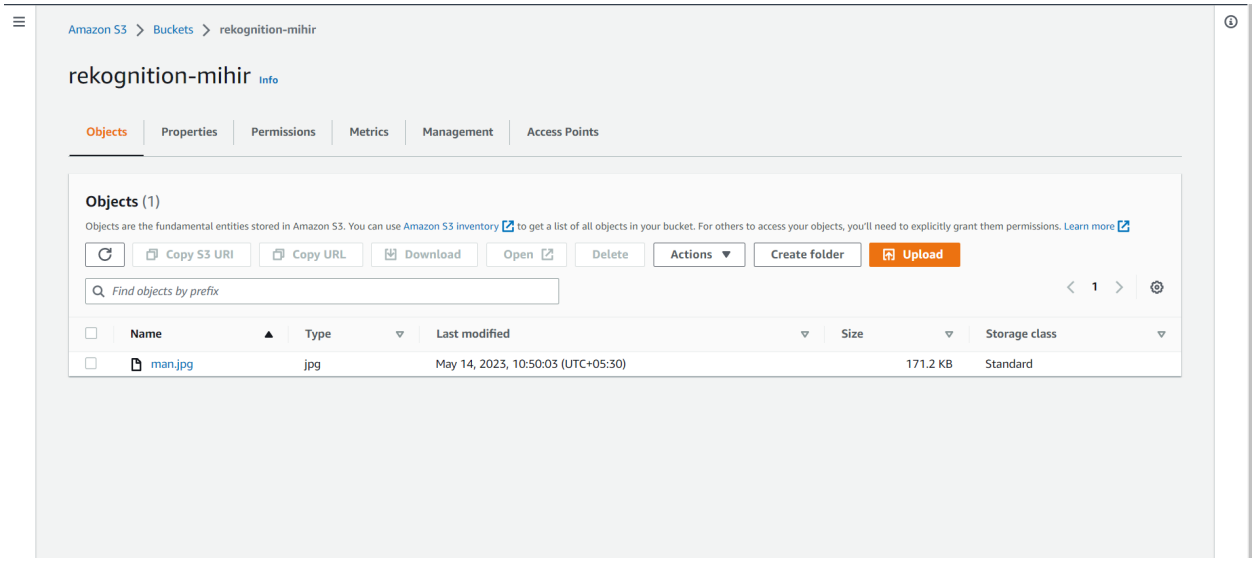




- Keep all default settings as it is and click on the Create bucket button.
- Once your bucket is created click on your bucket name. In that click on the upload button and drag and drop any image that you want and click on the upload button directly. Once the image is uploaded you can see the image as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

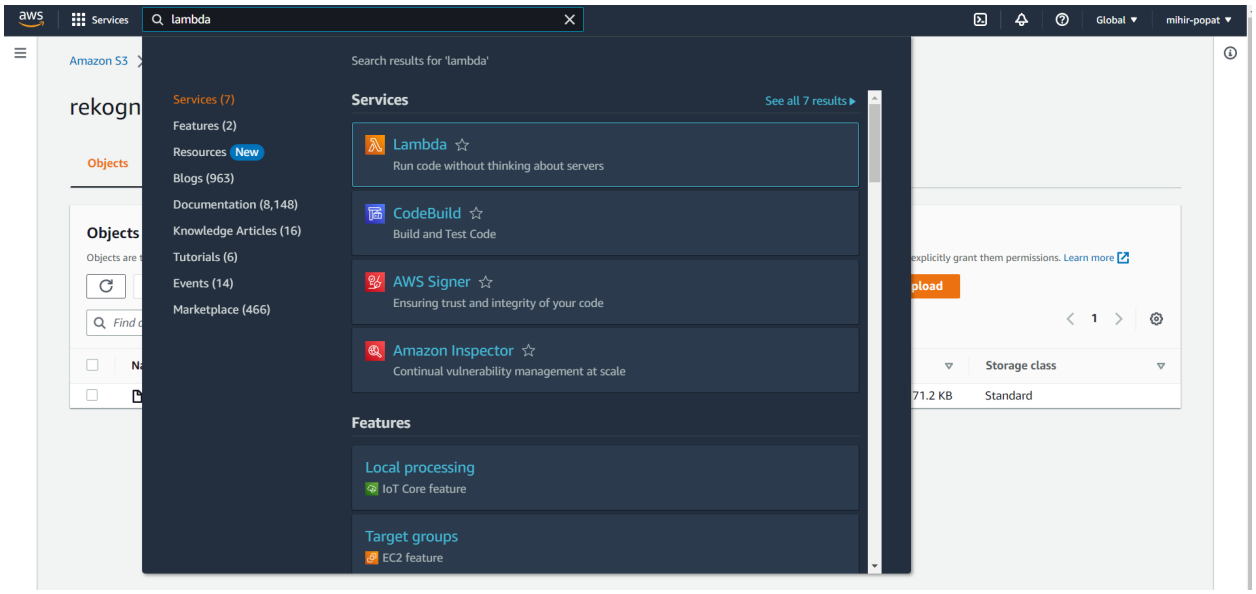

aws
Services
Search
[Alt+S]
Global
mihir-popat
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight 3
AWS Marketplace for S3
Amazon S3 > Buckets > rekognition-mihir
rekognition-mihir Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Objects (0)
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
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Find objects by prefix
1
Name
Type
Last modified
Size
Storage class
No objects
You don't have any objects in this bucket.
Upload


Amazon S3 > Buckets > rekognition-mihir > Upload
Upload Info
Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more
Drag and drop files and folders you want to upload here, or choose Add files, or Add folders.
Files and folders (1 Total, 171.2 KB)
Remove
Add files
Add folder
All files and folders in this table will be uploaded.
Find by name
1
Name
Folder
Type
Size
man.jpg
-
image/jpeg
171.2 KB
Destination
Destination
s3://rekognition-mihir
Destination details
Bucket settings that impact new objects stored in the specified destination.

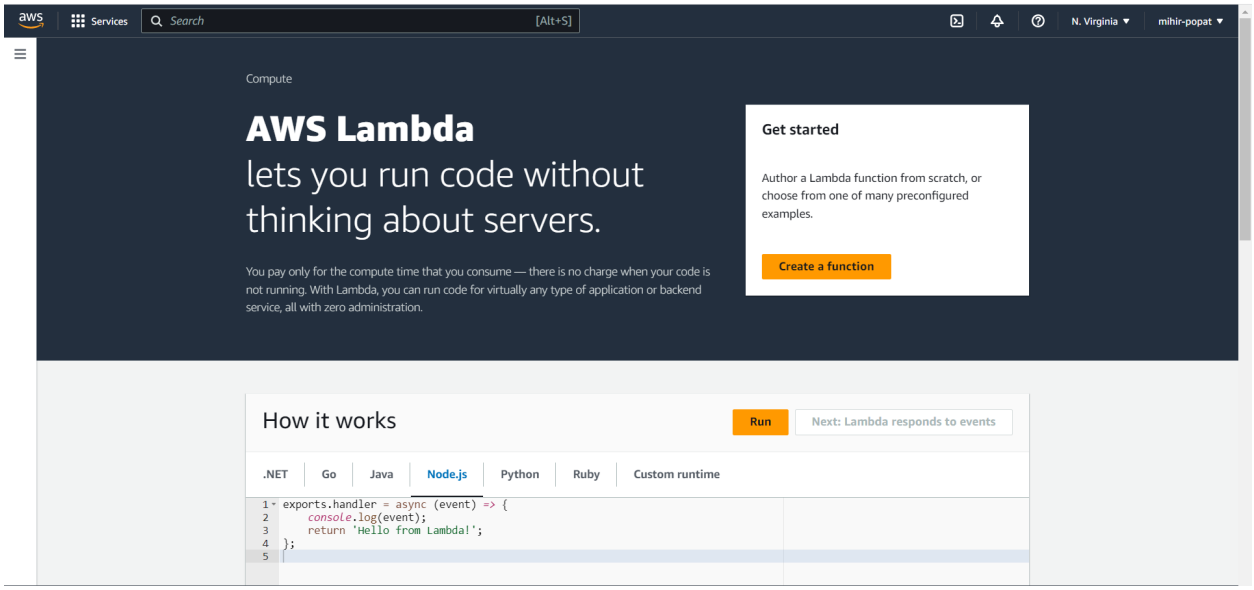


Step 3: Create a Lambda Function:

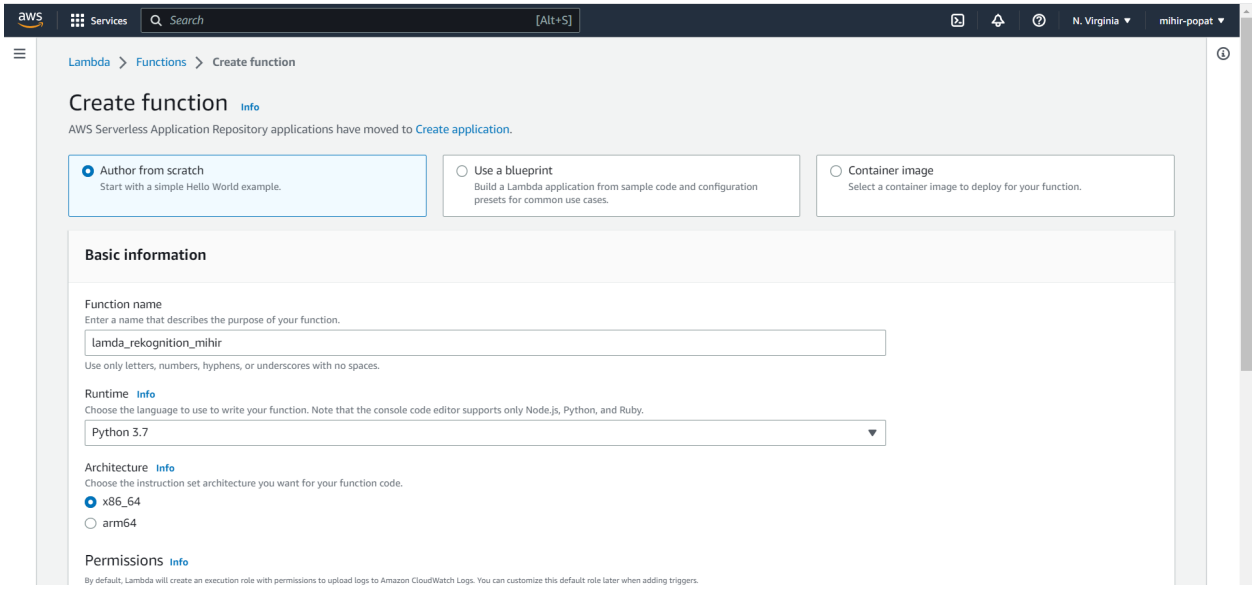
- Go to the AWS Management console.
- Search for the Lambda service and enter.



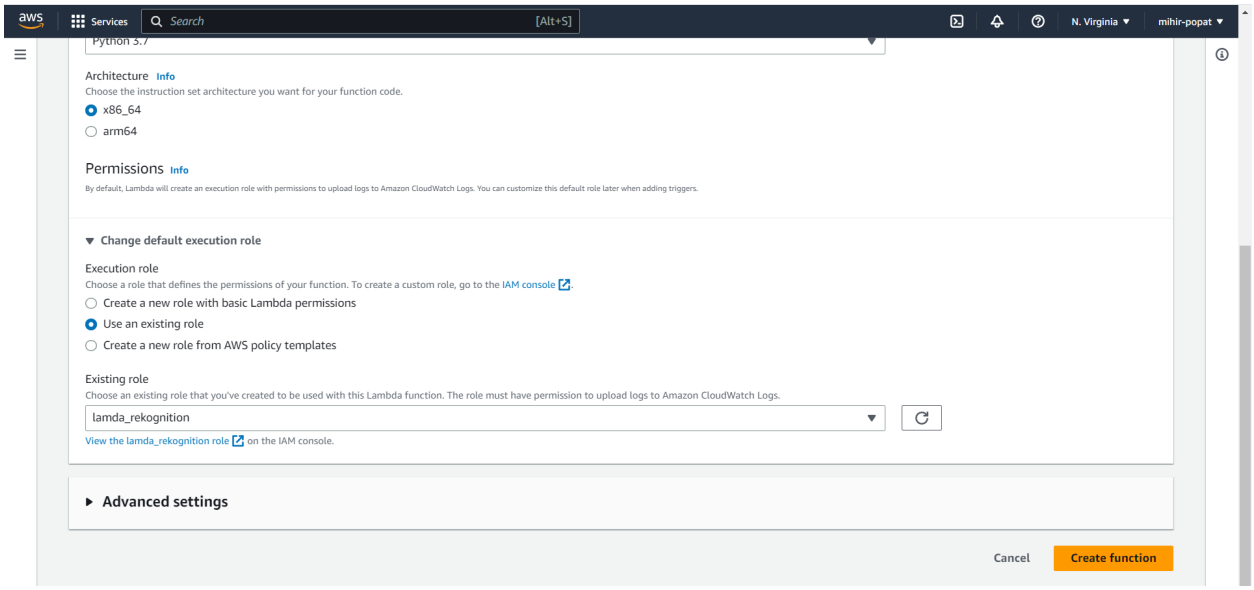
- After coming onto the lambda service page click on the Create function button.



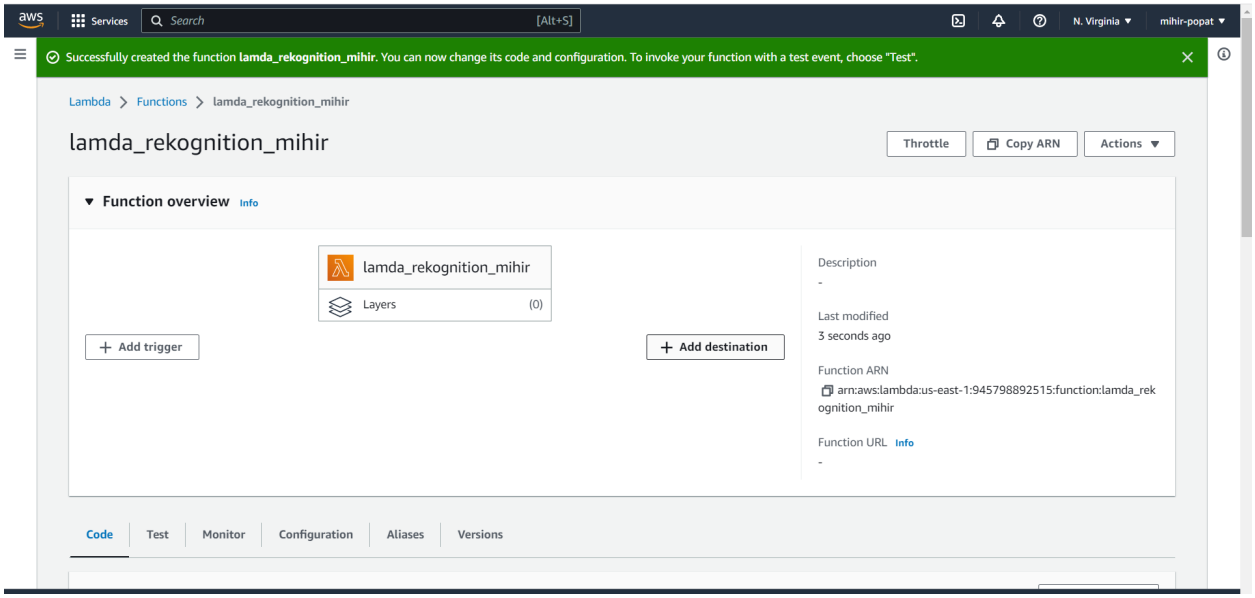
- Select the Author from Scratch default option.
- For function name give any name of your choice[for eg lamda_rekognition].
- In the Runtime select python 3.7



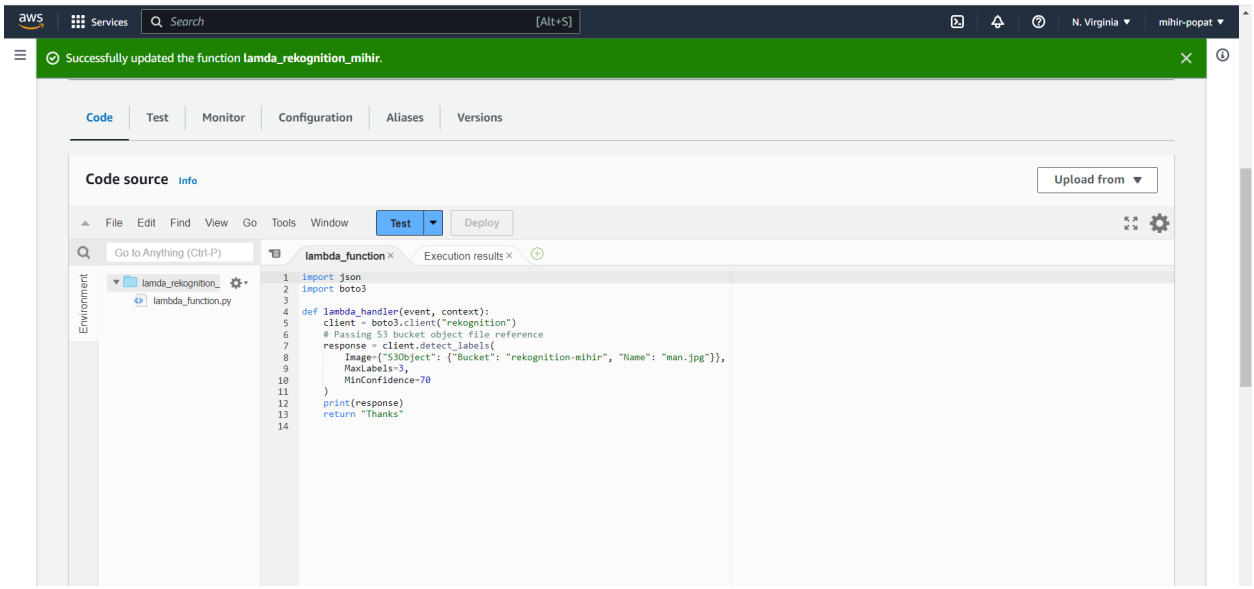
- Expand the Choose or create an execution role.
- In that select Use an existing role. And in the existing role select the role that we created in our first step[I have given the name for a role is lamda_rekognition].

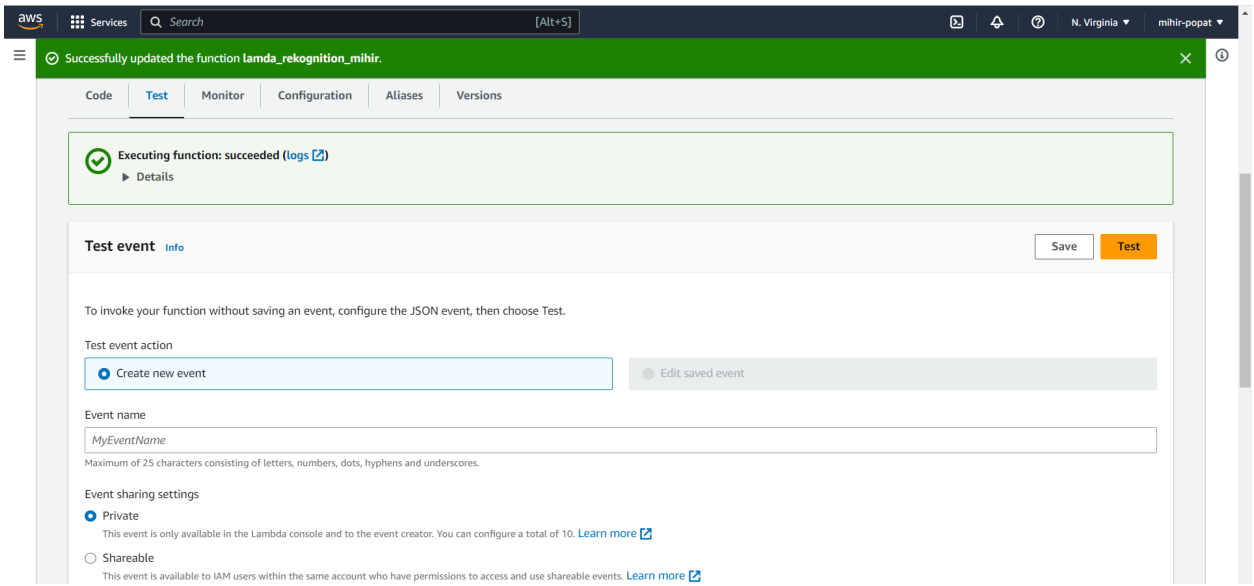


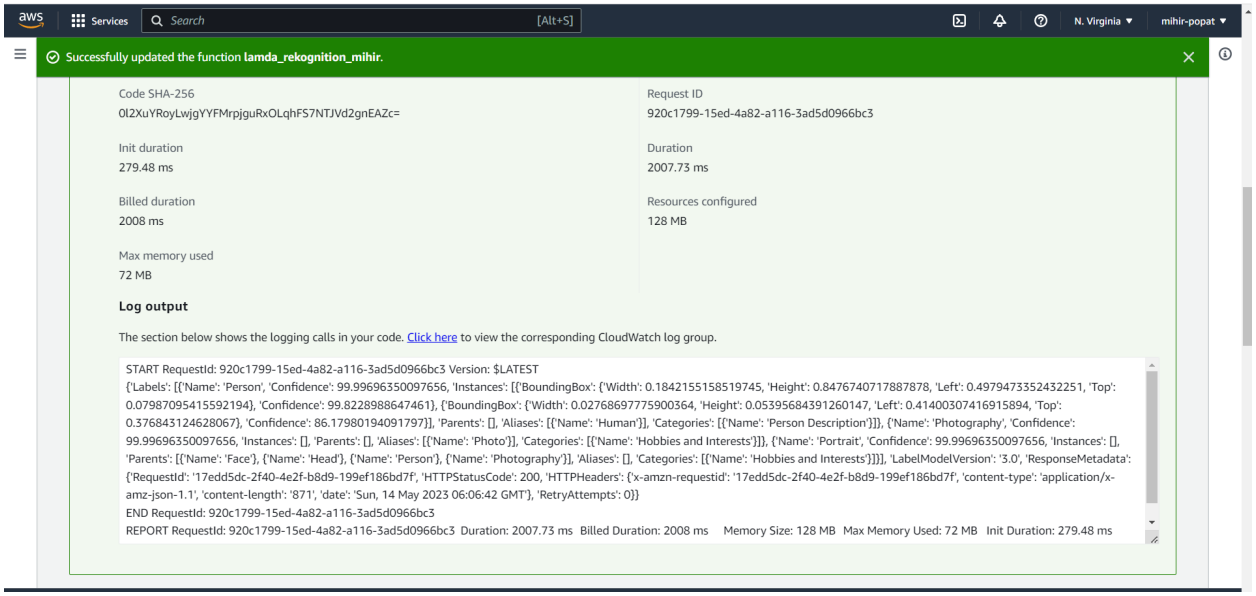
- Finally, click on the create function button.
- Once you created a lambda function then click on your function name.



- 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


Services
Search
[Alt+S]
N. Virginia
mihir-popat
Successfully updated the function lamda_rekognition_mihir.
Code
Test
Monitor
Configuration
Aliases
Versions
Code source Info
Upload from
File Edit Find View Go Tools Window
Test
Deploy
Go to Anything (Ctrl-P)
lambda_function
Execution results
Environment
lamda_rekognition_
lambda_function.py
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": "rekognition-mihir", "Name": "man.jpg"}},
MaxLabels=3,
MinConfidence=70
)
print(response)
return "Thanks"


Services
Search
[Alt+S]
N. Virginia
mihir-popat
Successfully updated the function lamda_rekognition_mihir.
Code
Test
Monitor
Configuration
Aliases
Versions
Executing function: succeeded (logs)
Details
Test event Info
Save
Test
To invoke your function without saving an event, configure the JSON event, then choose Test.
Test event action
Create new event
Edit saved event
Event name
MyEventName
Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.
Event sharing settings
Private
This event is only available in the Lambda console and to the event creator. You can configure a total of 10. Learn more
Shareable
This event is available to IAM users within the same account who have permissions to access and use shareable events. Learn more



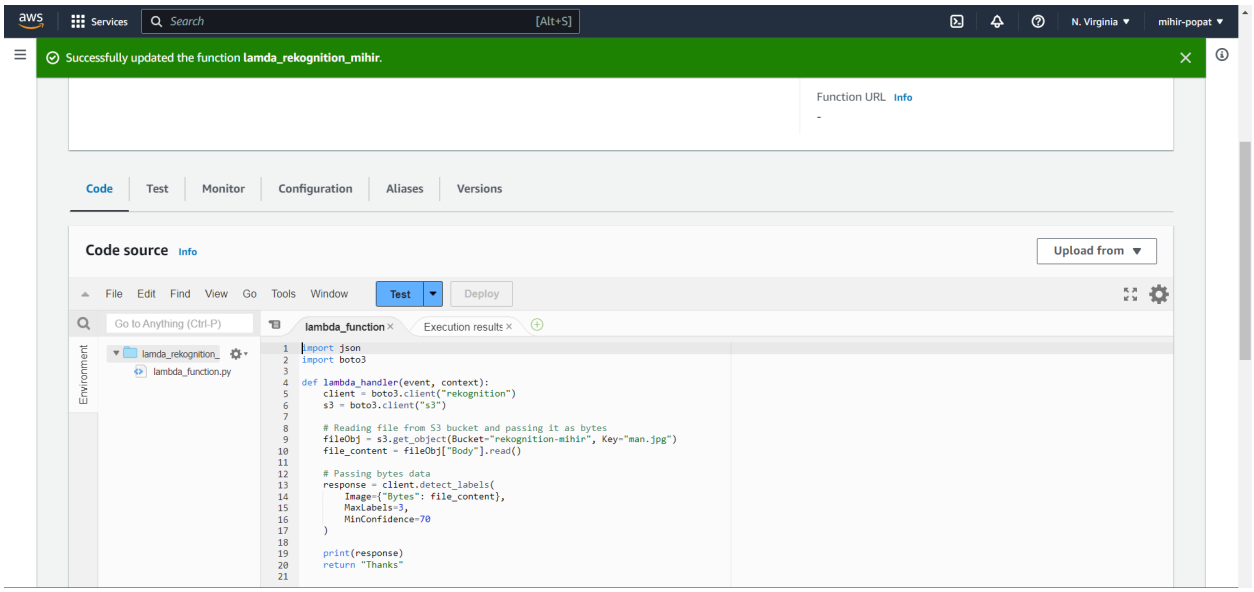
- Note: In place of buket_name and image_name please mention your S3 bucket name and uploaded image name.

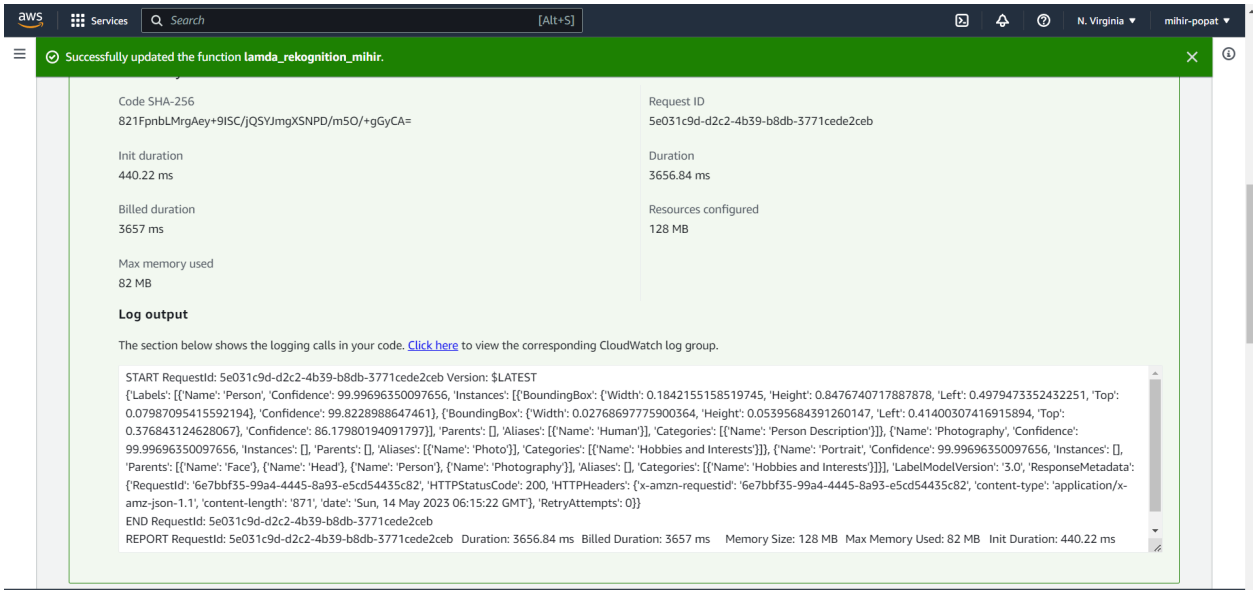
```
import json
import boto3

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")
```

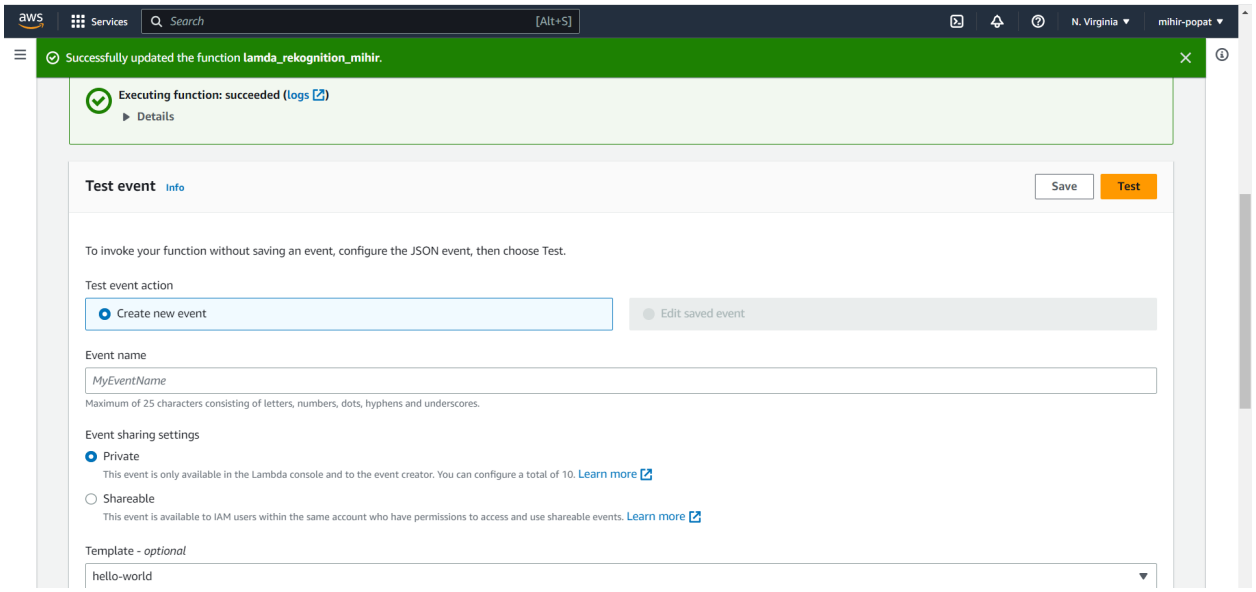
```
# Reading file from S3 bucket and passing it as bytes
fileObj = s3.get_object(Bucket="bucket_name", Key="image_name")
file_content = fileObj["Body"].read()

# Passing bytes data
response = client.detect_labels(
    Image={"Bytes": file_content},
    MaxLabels=3,
    MinConfidence=70
)

print(response)
return "Thanks"
```



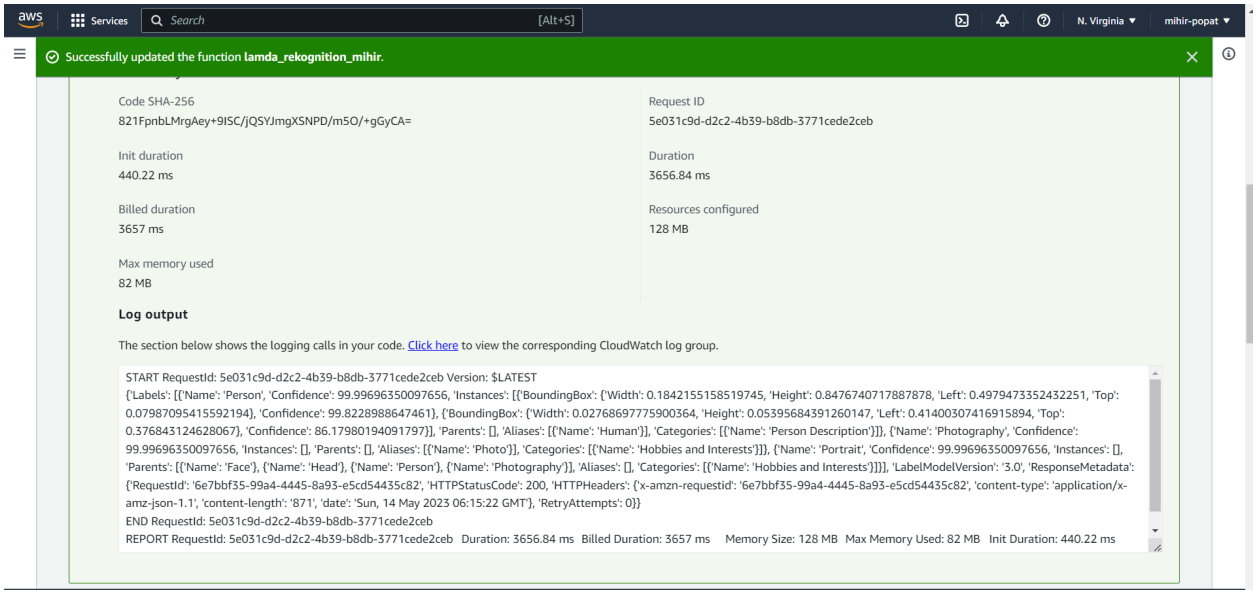


You can add any one of the codes that I have given above and you will get the same response. I have just shown you two different kinds of code for the lambda function.

- 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.



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



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 picture of a person 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 e.g height and width of persons face).

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