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

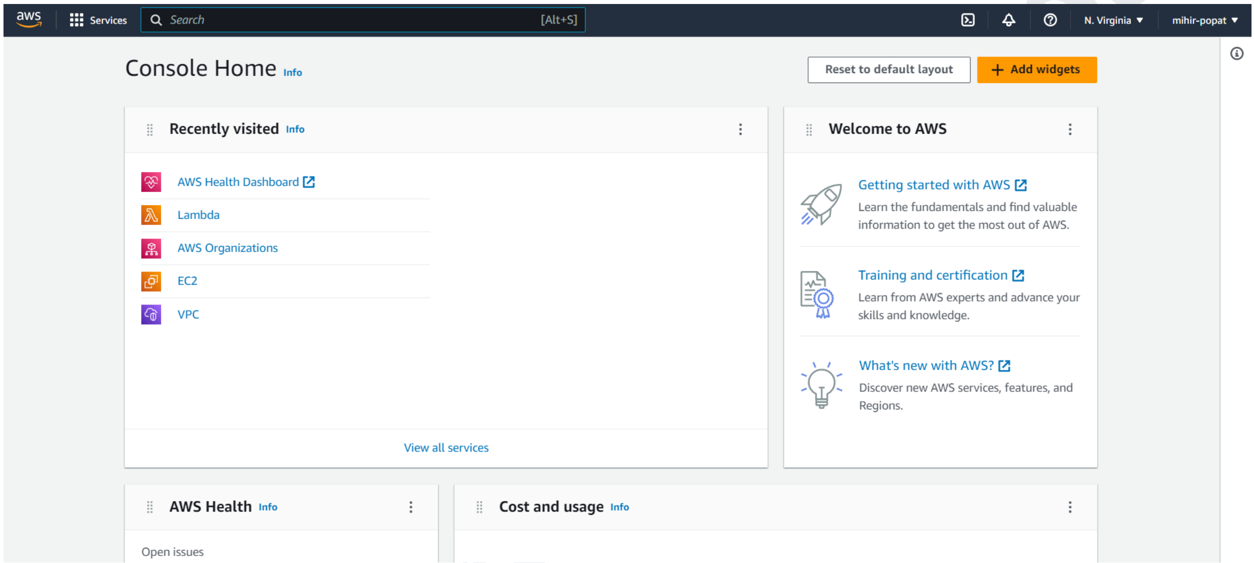
Flow for image analysis will be

- 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.



aws Services Search [Alt+S] N. Virginia ▾ mihir-popat ▾

Console Home Info

Reset to default layout

+ Add widgets

Recently visited Info

AWS Health Dashboard

Lambda

AWS Organizations

EC2

VPC

View all services

Welcome to AWS

Getting started with AWS

Learn the fundamentals and find valuable information to get the most out of AWS.

Training and certification

Learn from AWS experts and advance your skills and knowledge.

What's new with AWS?

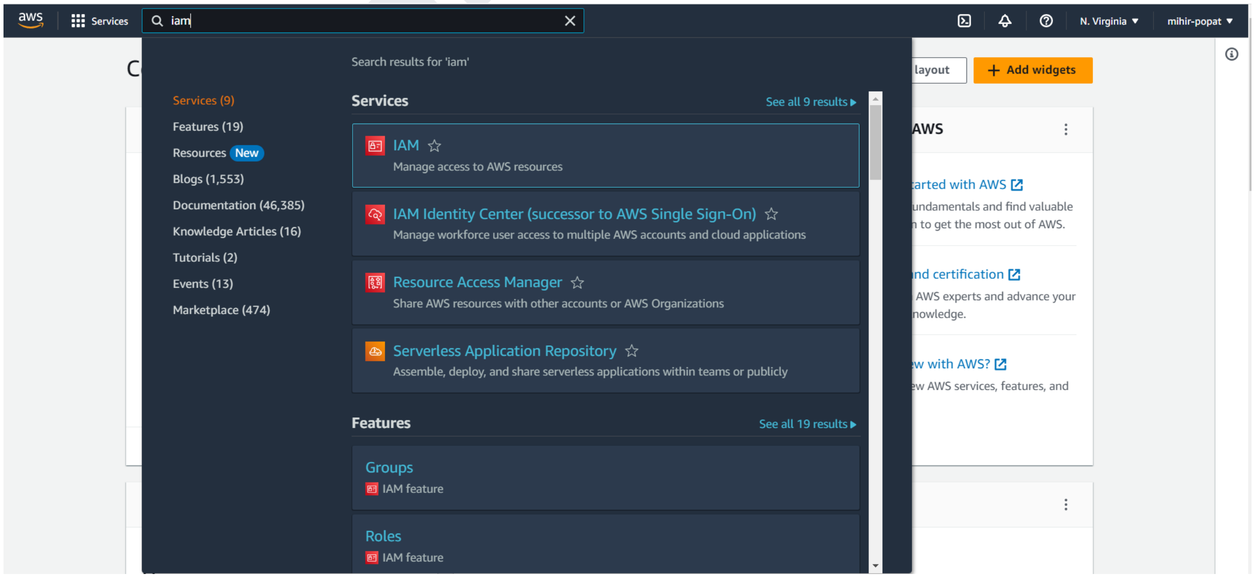
Discover new AWS services, features, and Regions.

AWS Health Info

Open issues

Cost and usage Info

- Search for the IAM service and enter.



aws Services iam N. Virginia ▾ mihir-popat ▾

Search results for 'iam'

Services (9)

Features (19)

Resources New

Blogs (1,553)

Documentation (46,385)

Knowledge Articles (16)

Tutorials (2)

Events (13)

Marketplace (474)

Services See all 9 results ▸

IAM ☆

Manage access to AWS resources

IAM Identity Center (successor to AWS Single Sign-On) ☆

Manage workforce user access to multiple AWS accounts and cloud applications

Resource Access Manager ☆

Share AWS resources with other accounts or AWS Organizations

Serverless Application Repository ☆

Assemble, deploy, and share serverless applications within teams or publicly

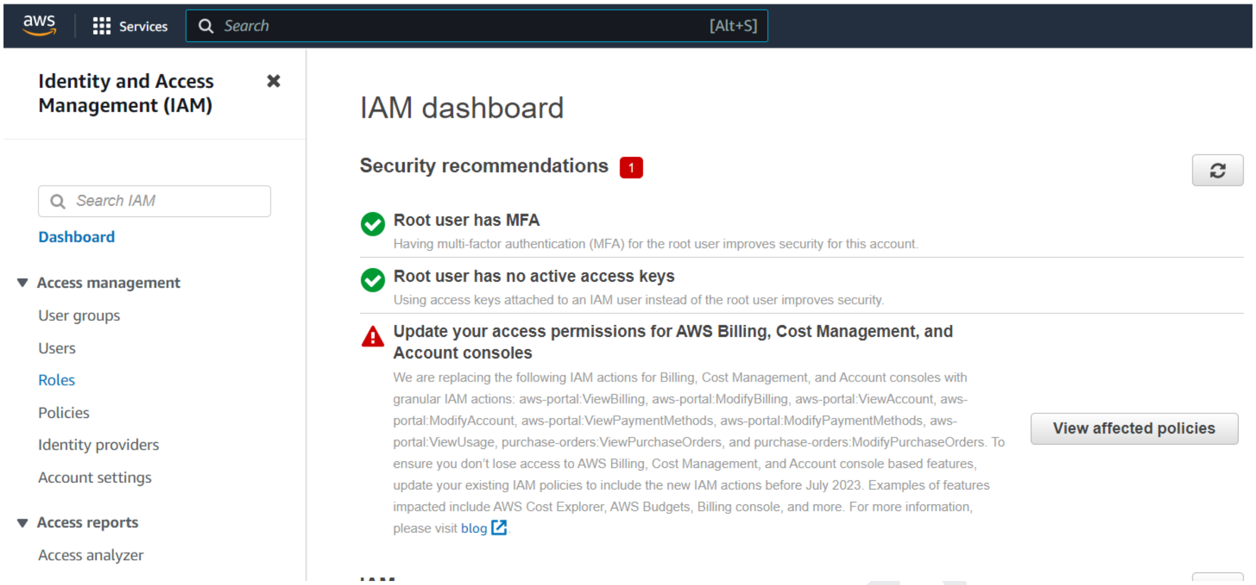
Features See all 19 results ▸

Groups

IAM feature

Roles

IAM feature



IAM dashboard

Security recommendations 1

Root user has MFA

Having multi-factor authentication (MFA) for the root user improves security for this account.

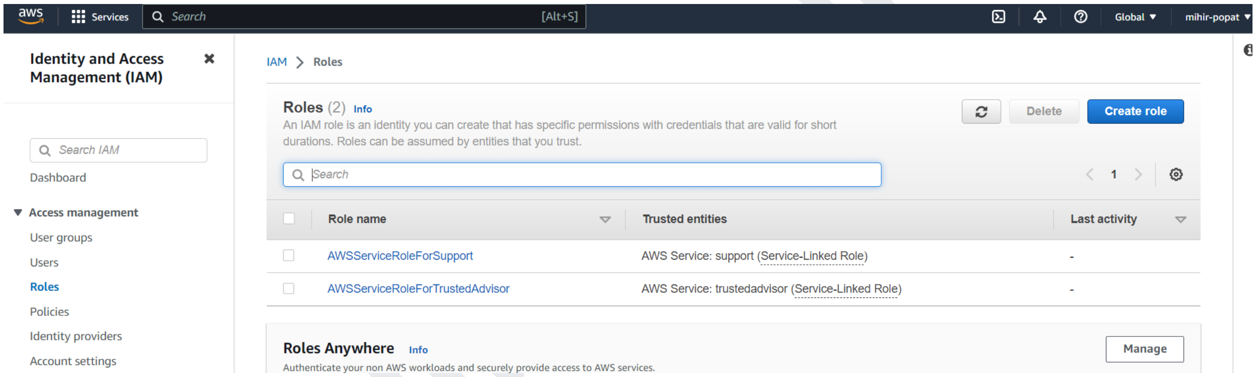
Root user has no active access keys

Using access keys attached to an IAM user instead of the root user improves security.

Update your access permissions for AWS Billing, Cost Management, and Account consoles

We are replacing the following IAM actions for Billing, Cost Management, and Account consoles with granular IAM actions: aws-portal:ViewBilling, aws-portal:ModifyBilling, aws-portal:ViewAccount, aws-portal:ModifyAccount, aws-portal:ViewPaymentMethods, aws-portal:ModifyPaymentMethods, aws-portal:ViewUsage, purchase-orders:ViewPurchaseOrders, and purchase-orders:ModifyPurchaseOrders. To ensure you don't lose access to AWS Billing, Cost Management, and Account console based features, update your existing IAM policies to include the new IAM actions before July 2023. Examples of features impacted include AWS Cost Explorer, AWS Budgets, Billing console, and more. For more information, please visit blog

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



Roles (2) Info

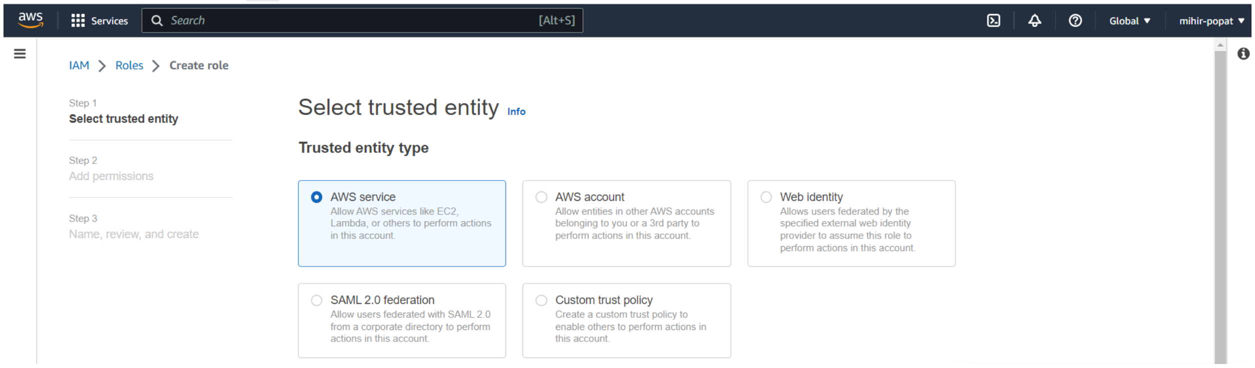
An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Role name	Trusted entities	Last activity
AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)	-

Roles Anywhere Info

Authenticate your non AWS workloads and securely provide access to AWS services.

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



Select trusted entity Info

Trusted entity type

● AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

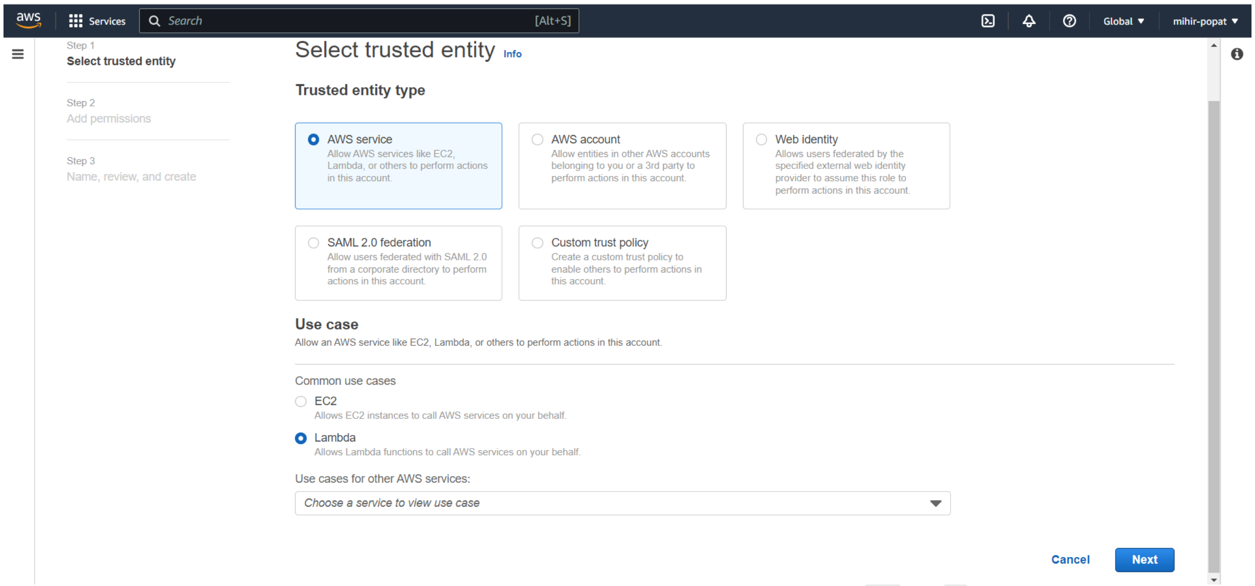
○ AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

○ Web identity
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

○ SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

○ Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.

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



Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Select trusted entity Info

Trusted entity type

● AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

○ AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

○ Web identity
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

○ SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

○ Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.

Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Common use cases

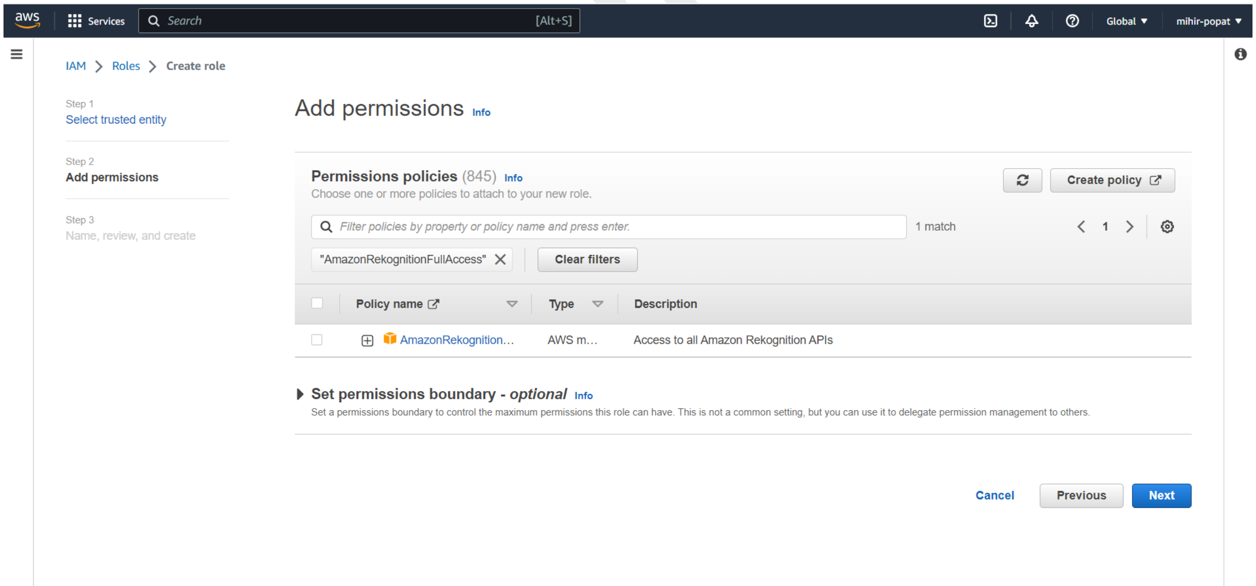
○ EC2
Allows EC2 instances to call AWS services on your behalf.

● Lambda
Allows Lambda functions to call AWS services on your behalf.

Use cases for other AWS services:
Choose a service to view use case

- In the Attach permissions policies select two policies :

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



IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Add permissions Info

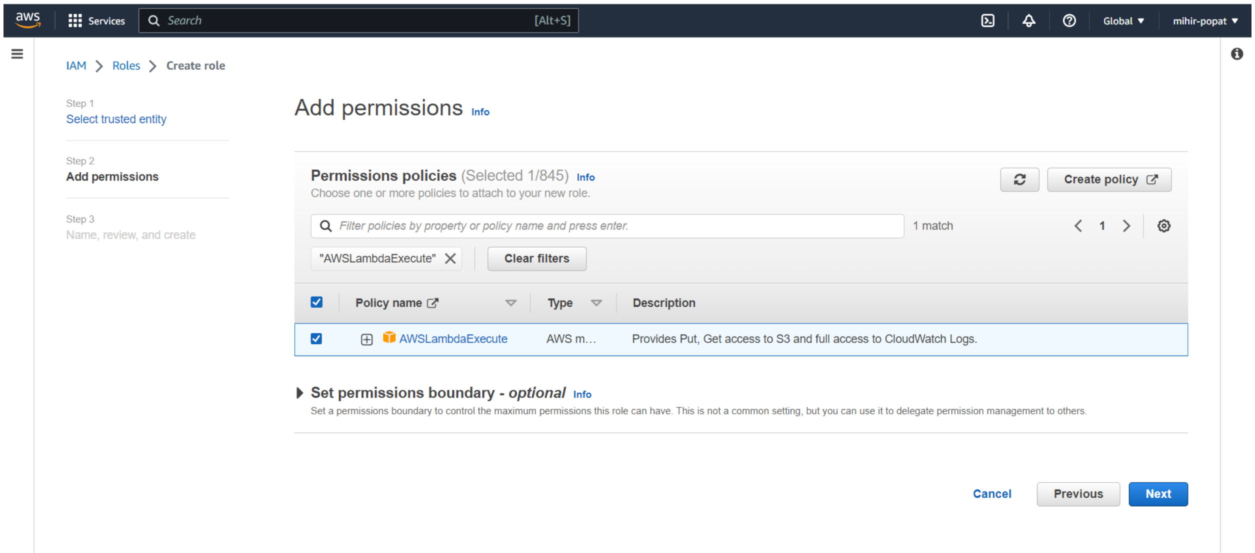
Permissions policies (845) Info
Choose one or more policies to attach to your new role.

Filter policies by property or policy name and press enter. 1 match

"AmazonRekognitionFullAccess" ✕ Clear filters

□	Policy name	Type	Description
□	AmazonRekognition...	AWS m...	Access to all Amazon Rekognition APIs

Set permissions boundary - *optional* Info
Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.



Add permissions Info

Permissions policies (Selected 1/845) Info

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

Filter policies by property or policy name and press enter. — 1 match

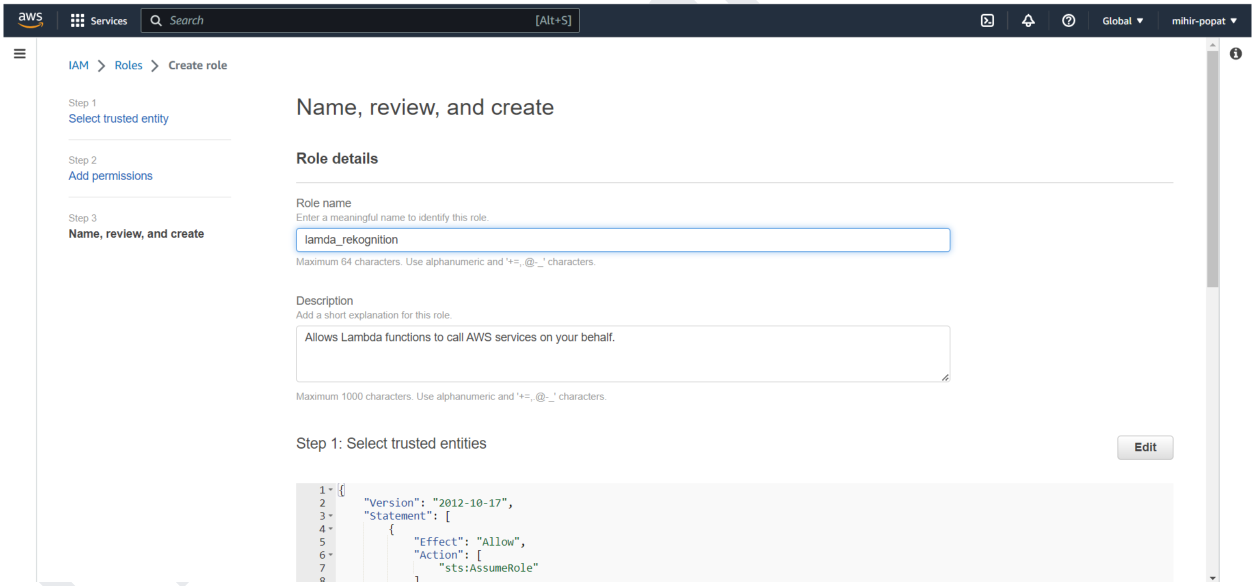
"AWSLambdaExecute" ✕ — Clear filters

☑	Policy name	Type	Description
☑	AWSLambdaExecute	AWS m…	Provides Put, Get access to S3 and full access to CloudWatch Logs.

Set permissions boundary - *optional* Info

Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.

- 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.



Name, review, and create

Role details

Role name

Enter a meaningful name to identify this role.

lamda_rekognition

Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

Description

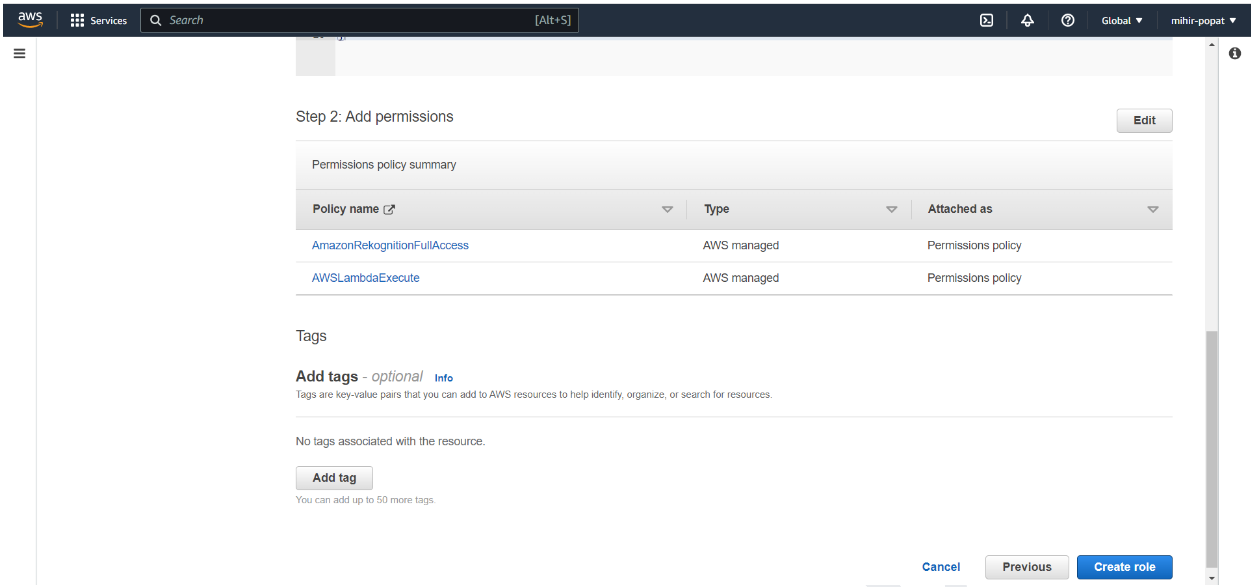
Add a short explanation for this role.

Allows Lambda functions to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

Step 1: Select trusted entities

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole"
            ]
```



Step 2: Add permissions

Permissions policy summary

Policy name	Type	Attached as
AmazonRekognitionFullAccess	AWS managed	Permissions policy
AWSLambdaExecute	AWS managed	Permissions policy

Tags

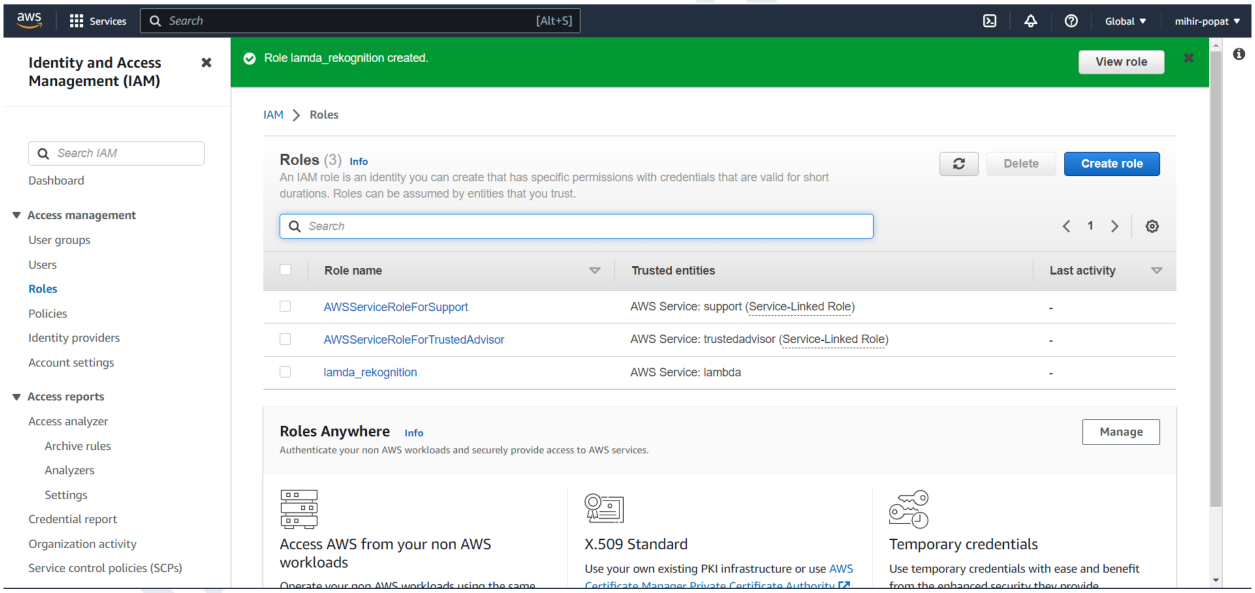
Add tags - *optional* Info

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

You can add up to 50 more tags.

- Your role is ready.



Role lamda_rekognition created.

Roles (3) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Role name	Trusted entities	Last activity
AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)	-
lamda_rekognition	AWS Service: lambda	-

Roles Anywhere Info

Authenticate your non AWS workloads and securely provide access to AWS services.

Access AWS from your non AWS workloads

Operate your non AWS workloads using the same

X.509 Standard

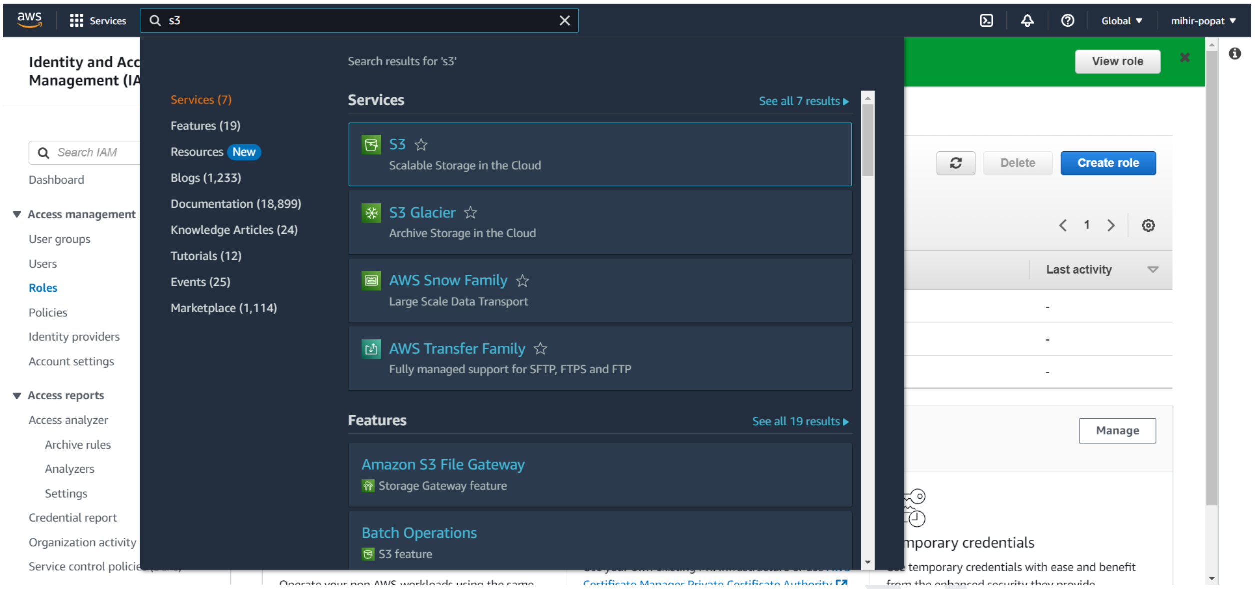
Use your own existing PKI infrastructure or use AWS Certificate Manager Private Certificate Authority

Temporary credentials

Use temporary credentials with ease and benefit from the enhanced security they provide

Step 2: Create an S3 bucket to store images:

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



s3

Search results for 's3'

Services (7)
Features (19)
Resources New
Blogs (1,233)
Documentation (18,899)
Knowledge Articles (24)
Tutorials (12)
Events (25)
Marketplace (1,114)

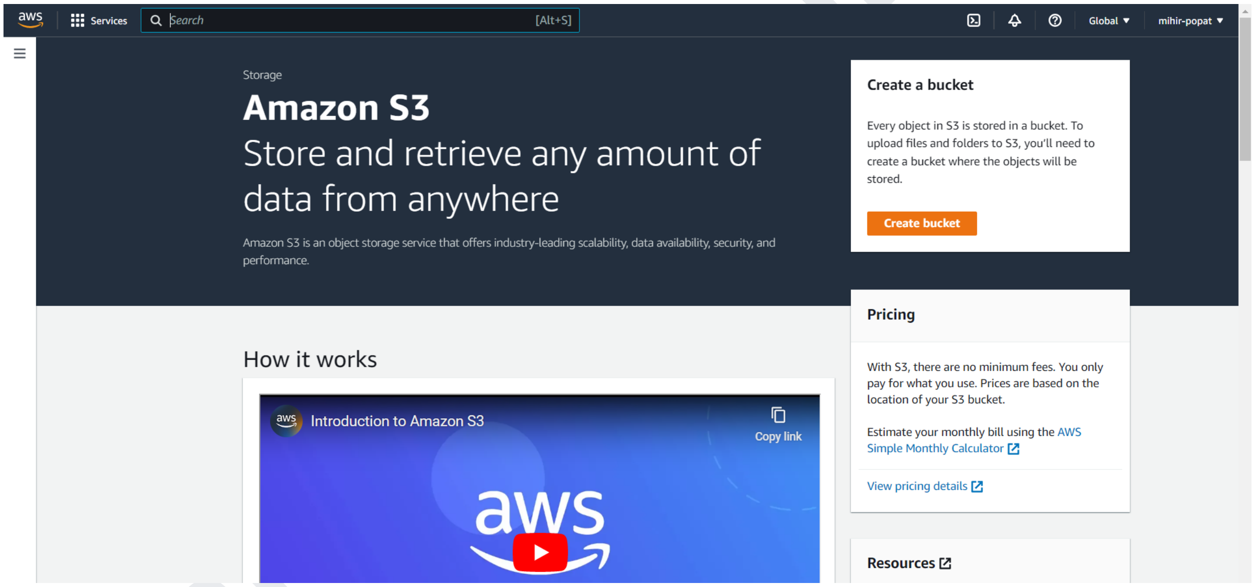
Services — See all 7 results

S3 — Scalable Storage in the Cloud
S3 Glacier — Archive Storage in the Cloud
AWS Snow Family — Large Scale Data Transport
AWS Transfer Family — Fully managed support for SFTP, FTPS and FTP

Features — See all 19 results

Amazon S3 File Gateway — Storage Gateway feature
Batch Operations — S3 feature

- Click on the Create bucket button.



Storage

Amazon S3

Store and retrieve any amount of data from anywhere

Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance.

Create a bucket

Every object in S3 is stored in a bucket. To upload files and folders to S3, you'll need to create a bucket where the objects will be stored.

Create bucket

Pricing

With S3, there are no minimum fees. You only pay for what you use. Prices are based on the location of your S3 bucket.

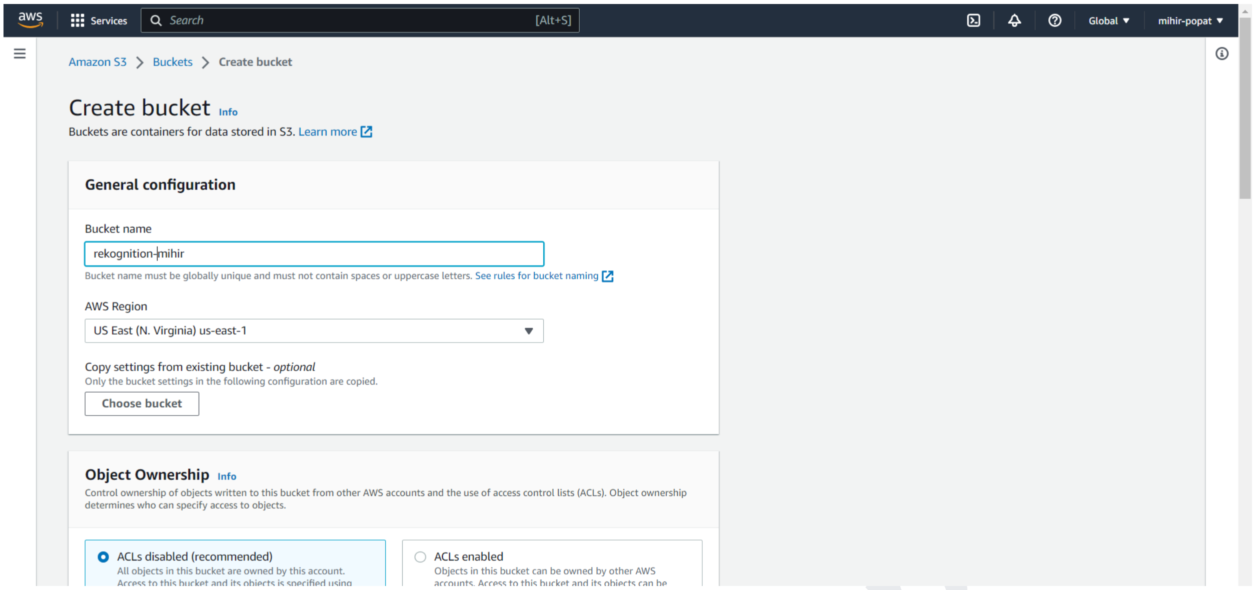
Estimate your monthly bill using the AWS Simple Monthly Calculator

View pricing details

Resources

How it works

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



Create bucket Info

Buckets are containers for data stored in S3. Learn more

General configuration

Bucket name: rekognition-mihir

Bucket name must be globally unique and must not contain spaces or uppercase letters. See rules for bucket naming

AWS Region: US East (N. Virginia) us-east-1

Copy settings from existing bucket - *optional*

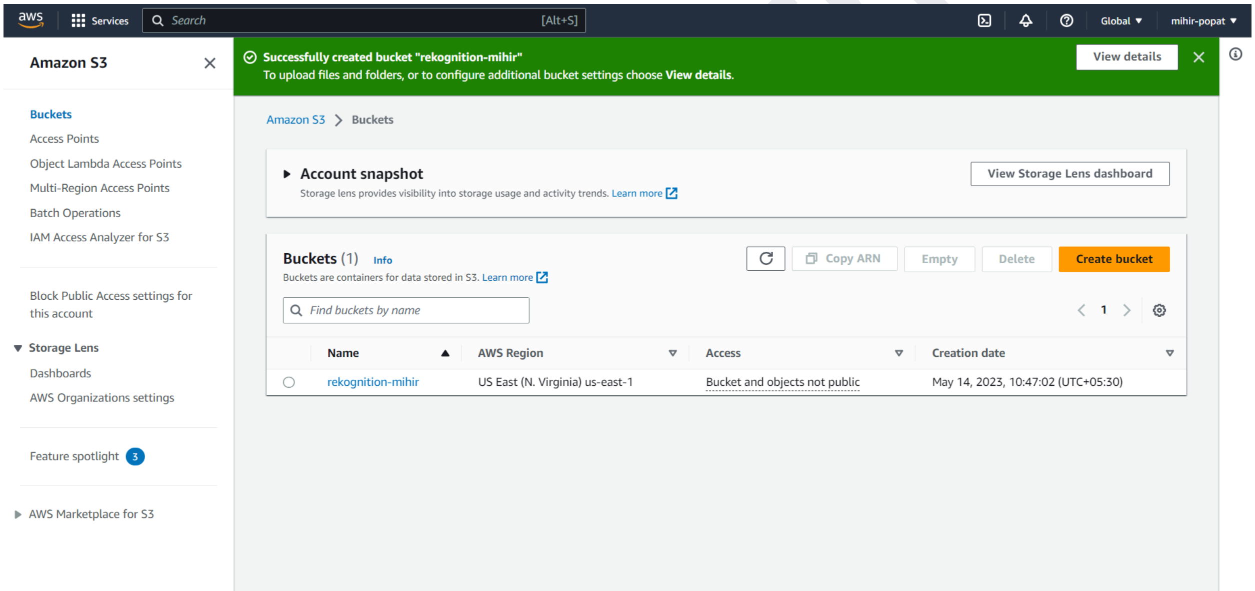
Only the bucket settings in the following configuration are copied.

Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

● ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using

○ ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be



Successfully created bucket "rekognition-mihir"

To upload files and folders, or to configure additional bucket settings choose **View details.**

Account snapshot

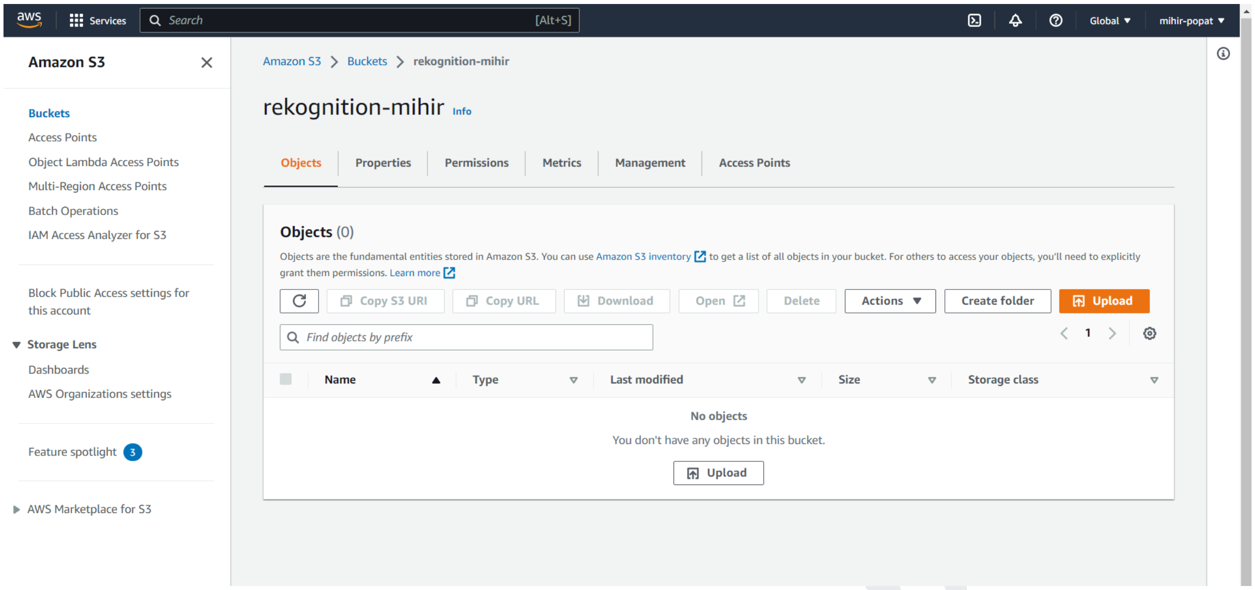
Storage lens provides visibility into storage usage and activity trends. Learn more

Buckets (1) Info

Buckets are containers for data stored in S3. Learn more

	Name	AWS Region	Access	Creation date
○	rekognition-mihir	US East (N. Virginia) us-east-1	Bucket and objects not public	May 14, 2023, 10:47:02 (UTC+05:30)

- 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



rekognition-mihir

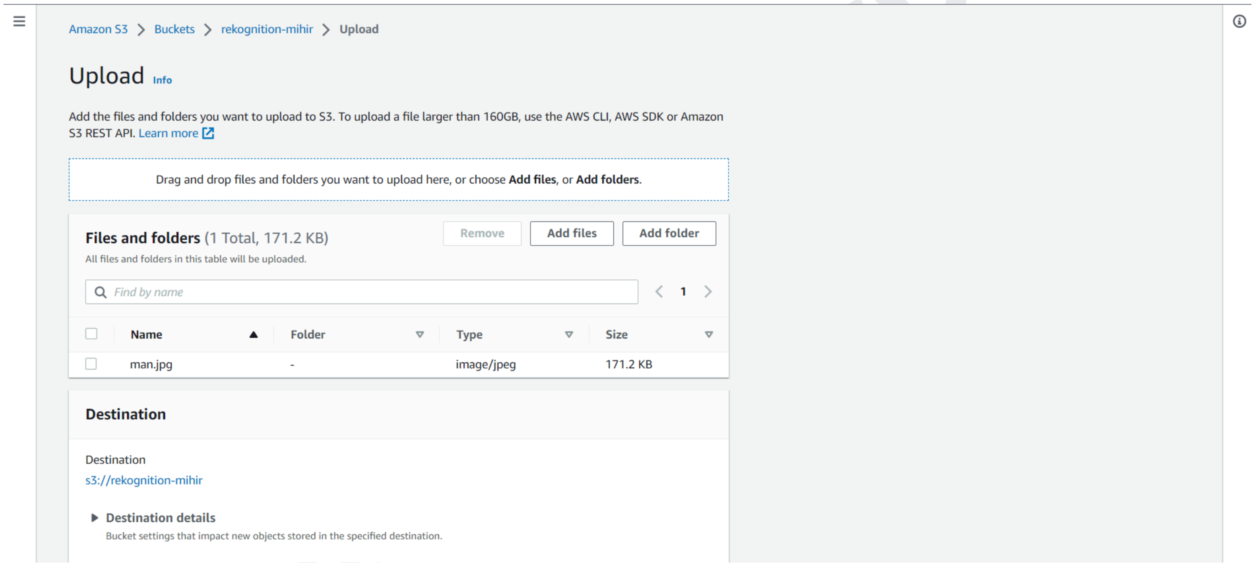
Objects (0)

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

Name	Type	Last modified	Size	Storage class

No objects

You don't have any objects in this bucket.



Upload

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)

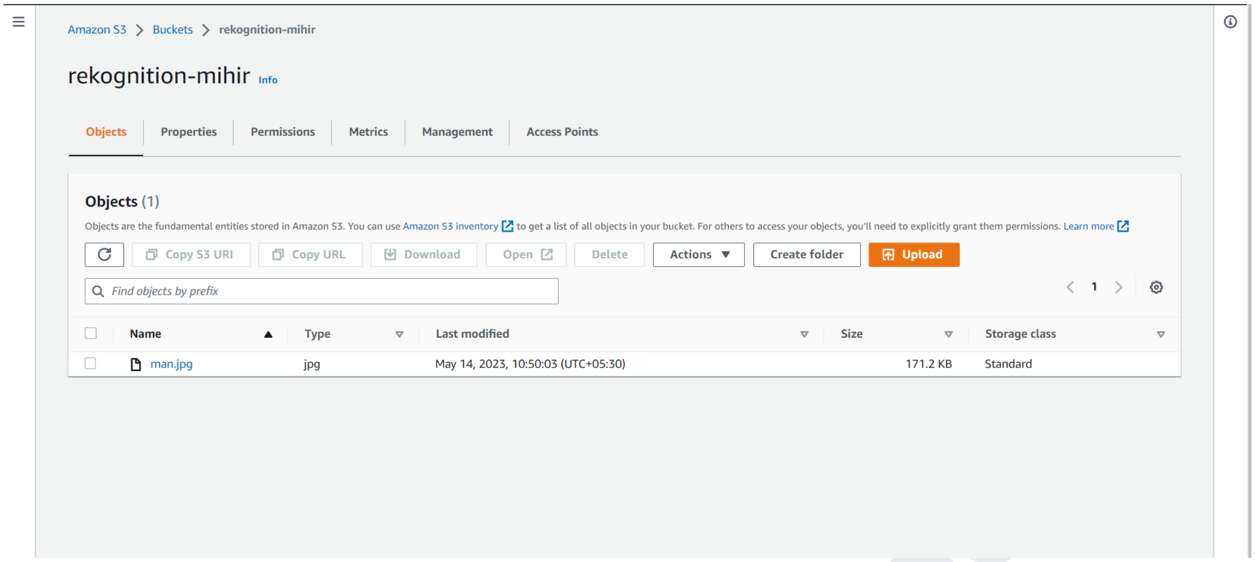
All files and folders in this table will be uploaded.

	Name	Folder	Type	Size
□	man.jpg	-	image/jpeg	171.2 KB

Destination

Destination
s3://rekognition-mihir

Bucket settings that impact new objects stored in the specified destination.

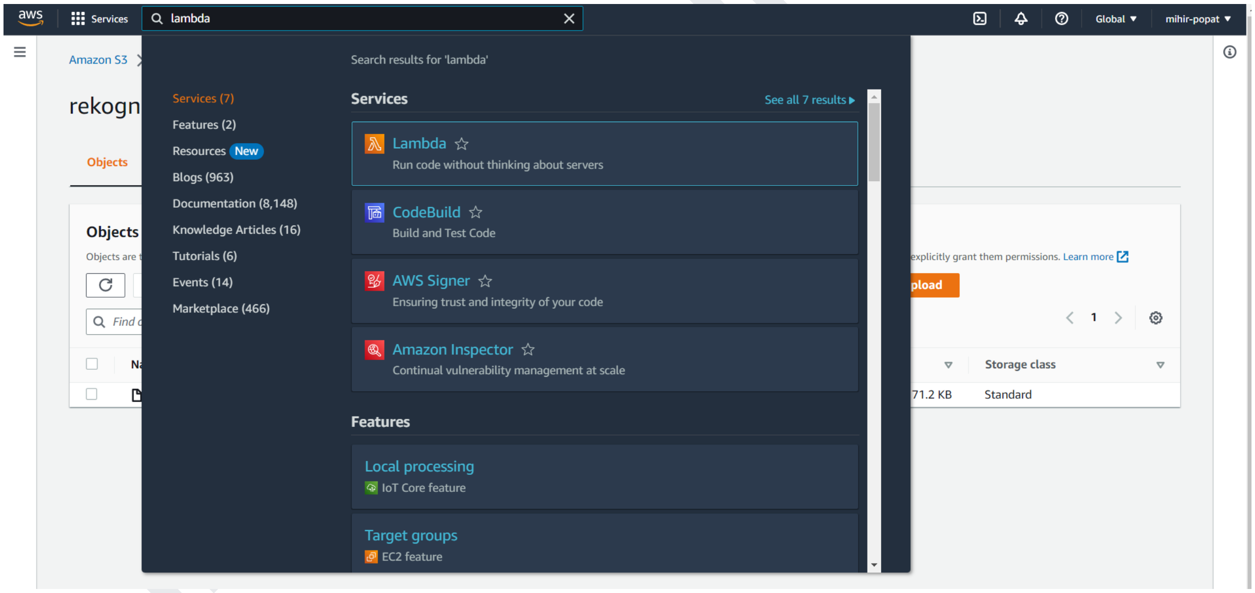


Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

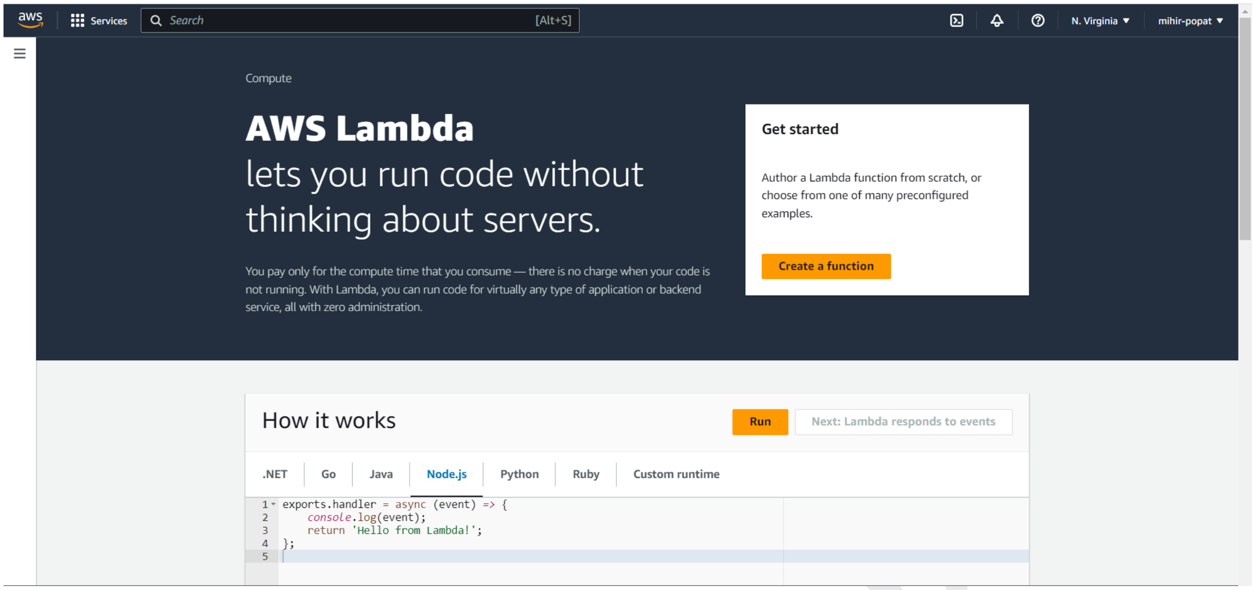
	Name	Type	Last modified	Size	Storage class
□	man.jpg	jpg	May 14, 2023, 10:50:03 (UTC+05:30)	171.2 KB	Standard

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.



Compute

AWS Lambda
lets you run code without thinking about servers.

You pay only for the compute time that you consume — there is no charge when your code is not running. With Lambda, you can run code for virtually any type of application or backend service, all with zero administration.

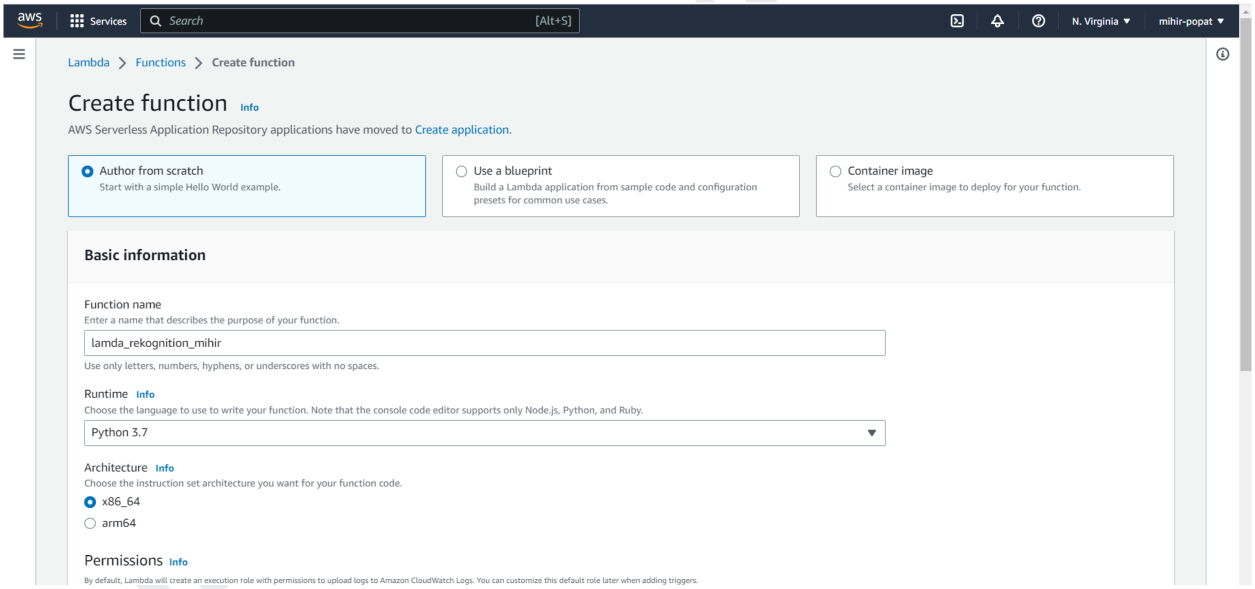
Get started

Author a Lambda function from scratch, or choose from one of many preconfigured examples.

How it works

```
exports.handler = async (event) => {
    console.log(event);
    return 'Hello from Lambda!';
};
```

- 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



Create function Info

AWS Serverless Application Repository applications have moved to Create application.

● Author from scratch
Start with a simple Hello World example.

○ Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.

○ Container image
Select a container image to deploy for your function.

Basic information

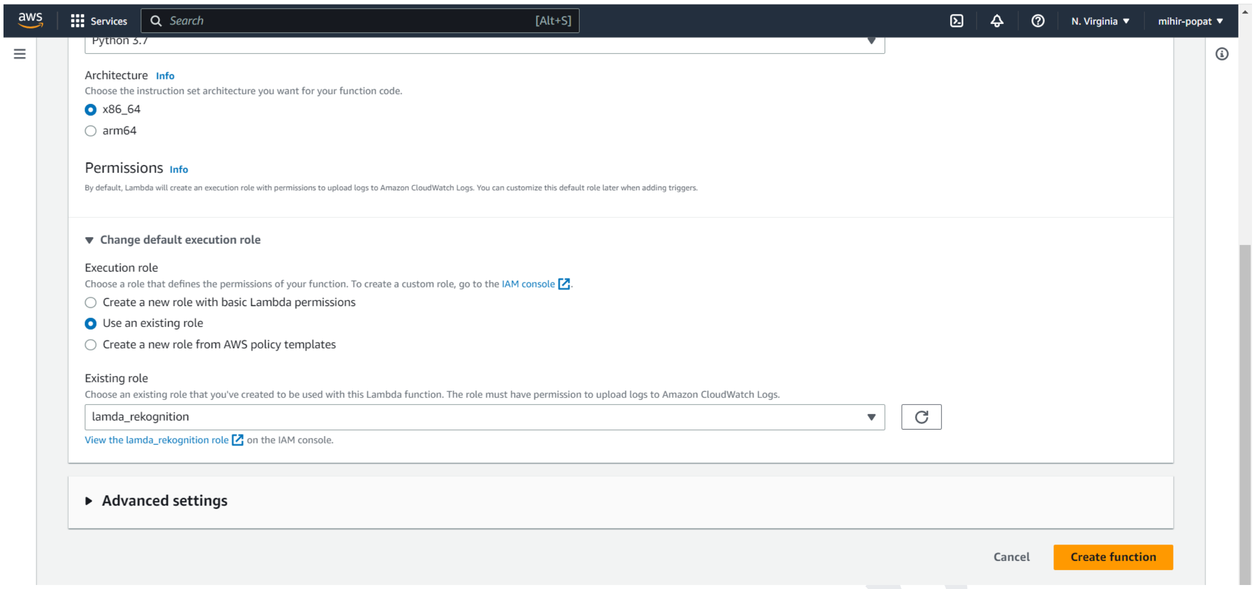
Function name
Enter a name that describes the purpose of your function.
lamda_rekognition_mihir
Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.7

Architecture Info
Choose the instruction set architecture you want for your function code.
● x86_64
○ arm64

Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

- 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].



Python 3.7

Architecture Info

Choose the instruction set architecture you want for your function code.

● x86_64

○ arm64

Permissions Info

By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ Change default execution role

Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.

○ Create a new role with basic Lambda permissions

● Use an existing role

○ Create a new role from AWS policy templates

Existing role

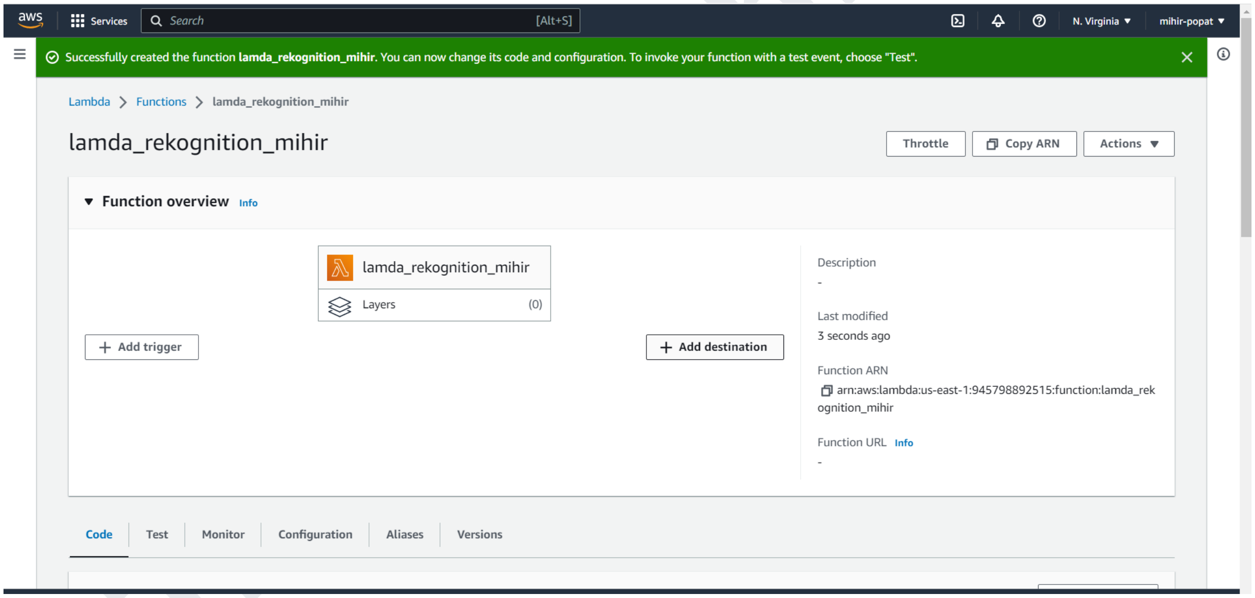
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

lamda_rekognition ▼

View the lamda_rekognition role on the IAM console.

▶ Advanced settings

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



Successfully created the function **lamda_rekognition_mihir**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Lambda > Functions > lamda_rekognition_mihir

lamda_rekognition_mihir

▼ Function overview Info

lamda_rekognition_mihir

Layers (0)

Description

-

Last modified

3 seconds ago

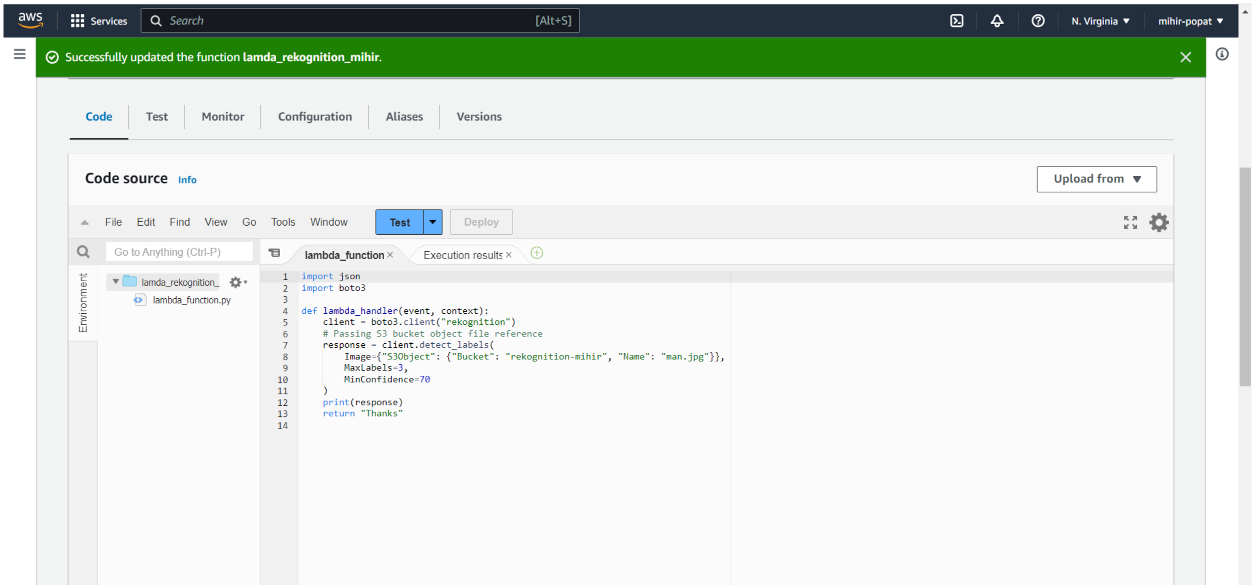
Function ARN

arn:aws:lambda:us-east-1:945798892515:function:lamda_rekognition_mihir

Function URL Info

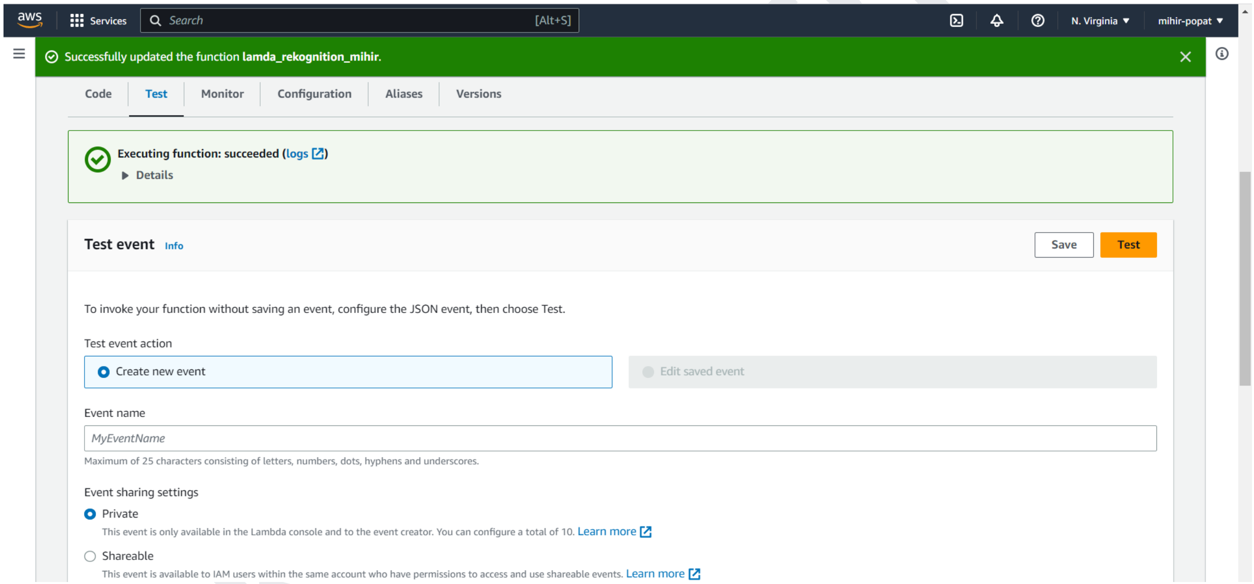
-

- 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



Successfully updated the function **lamda_rekognition_mihir.**

```python
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"

```



Successfully updated the function **lamda_rekognition_mihir.**

Executing function: succeeded (logs)

▶ Details

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.

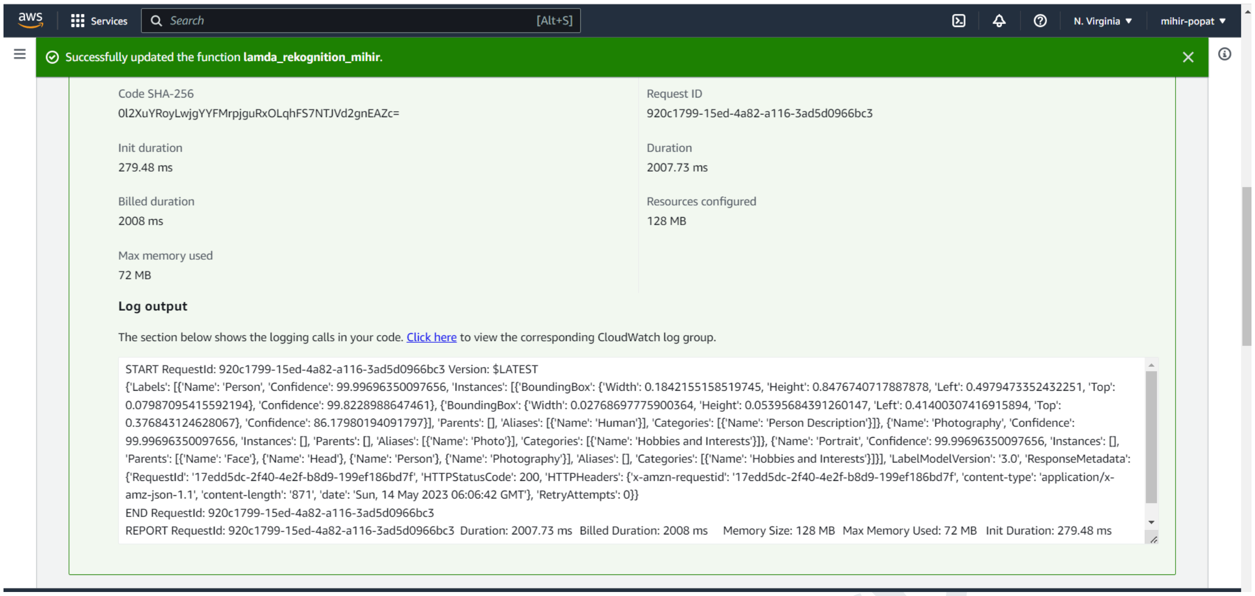
Event sharing settings

● 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



Successfully updated the function **lamda_rekognition_mihir**.

Code SHA-256
0l2XuYRoyLwjgYYFMrpjguRxOLqhFS7NTJVd2gnEAZc=

Request ID
920c1799-15ed-4a82-a116-3ad5d0966bc3

Init duration
279.48 ms

Duration
2007.73 ms

Billed duration
2008 ms

Resources configured
128 MB

Max memory used
72 MB

Log output

The section below shows the logging calls in your code. Click here to view the corresponding CloudWatch log group.

```
START RequestId: 920c1799-15ed-4a82-a116-3ad5d0966bc3 Version: $LATEST
{'Labels': [{'Name': 'Person', 'Confidence': 99.99696350097656, 'Instances': [{'BoundingBox': {'Width': 0.1842155158519745, 'Height': 0.8476740717887878, 'Left': 0.4979473352432251, 'Top': 0.07987095415592194}, 'Confidence': 99.8228988647461}, {'BoundingBox': {'Width': 0.02768697775900364, 'Height': 0.05395684391260147, 'Left': 0.41400307416915894, 'Top': 0.376843124628067}, 'Confidence': 86.17980194091797}], 'Parents': [], 'Aliases': [{'Name': 'Human'}], 'Categories': [{'Name': 'Person Description'}]}, {'Name': 'Photography', 'Confidence': 99.99696350097656, 'Instances': [], 'Parents': [], 'Aliases': [{'Name': 'Photo'}], 'Categories': [{'Name': 'Hobbies and Interests'}]}, {'Name': 'Portrait', 'Confidence': 99.99696350097656, 'Instances': [], 'Parents': [{'Name': 'Face'}, {'Name': 'Head'}, {'Name': 'Person'}, {'Name': 'Photography'}], 'Aliases': [], 'Categories': [{'Name': 'Hobbies and Interests'}]}], 'LabelModelVersion': '3.0', 'ResponseMetadata': {'RequestId': '17edd5dc-2f40-4e2f-b8d9-199ef186bd7f', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amzn-requestid': '17edd5dc-2f40-4e2f-b8d9-199ef186bd7f', 'content-type': 'application/x-amz-json-1.1', 'content-length': '871', 'date': 'Sun, 14 May 2023 06:06:42 GMT'}, 'RetryAttempts': 0}}
END RequestId: 920c1799-15ed-4a82-a116-3ad5d0966bc3
REPORT RequestId: 920c1799-15ed-4a82-a116-3ad5d0966bc3  Duration: 2007.73 ms  Billed Duration: 2008 ms    Memory Size: 128 MB  Max Memory Used: 72 MB  Init Duration: 279.48 ms
```

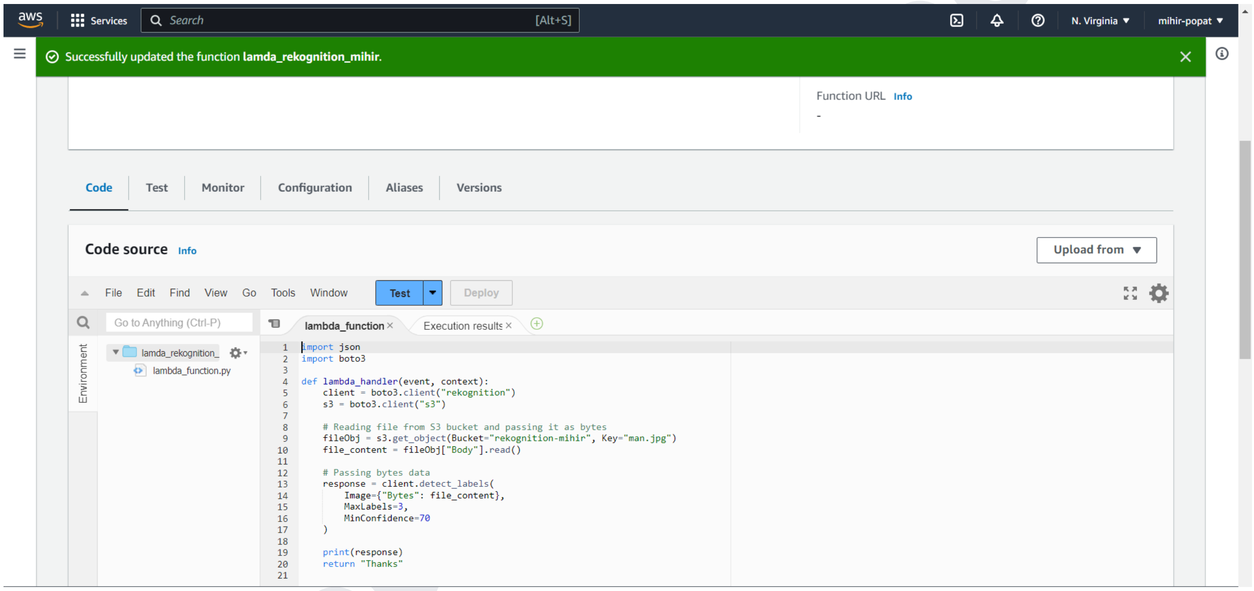
- Note: In place of buket_name and image_name please mention your S3 bucket name and uploaded image name.

```
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.

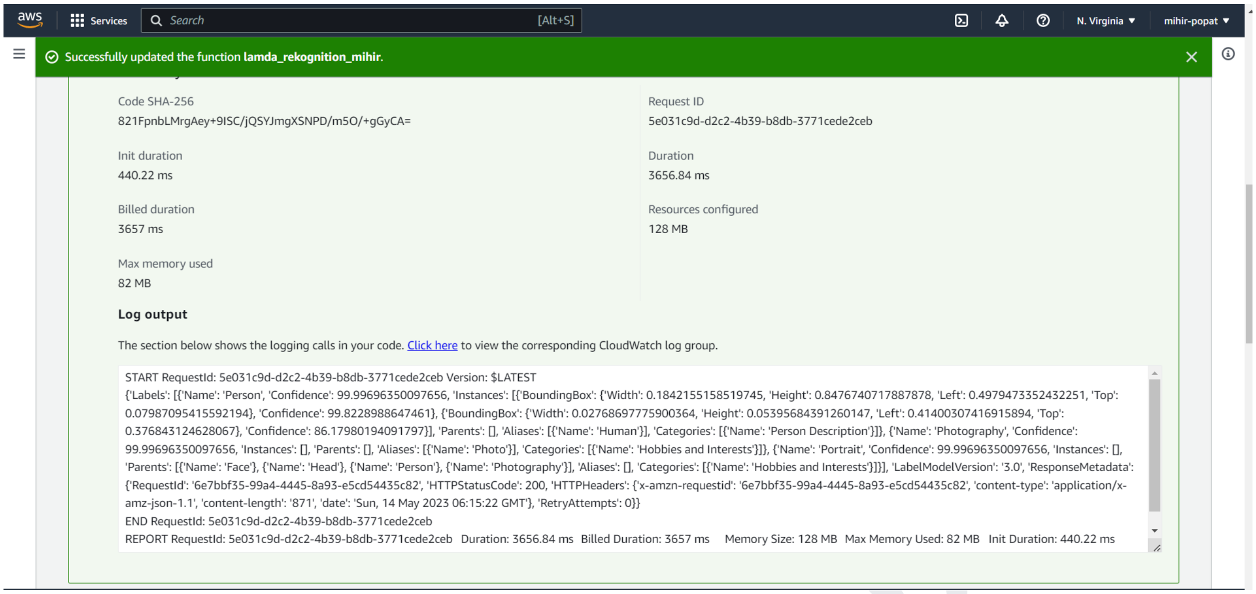
```
import json
import boto3

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"
```



Successfully updated the function **lamda_rekognition_mihir**.

```
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="rekognition-mihir", Key="man.jpg")
    file_content = fileObj["Body"].read()

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

    print(response)
    return "Thanks"

```



Successfully updated the function **lamda_rekognition_mihir**.

Code SHA-256
821FpnbLMrgAey+9ISC/jQSYJmgXSNPD/m5O/+gGyCA=

Request ID
5e031c9d-d2c2-4b39-b8db-3771cede2ceb

Init duration
440.22 ms

Duration
3656.84 ms

Billed duration
3657 ms

Resources configured
128 MB

Max memory used
82 MB

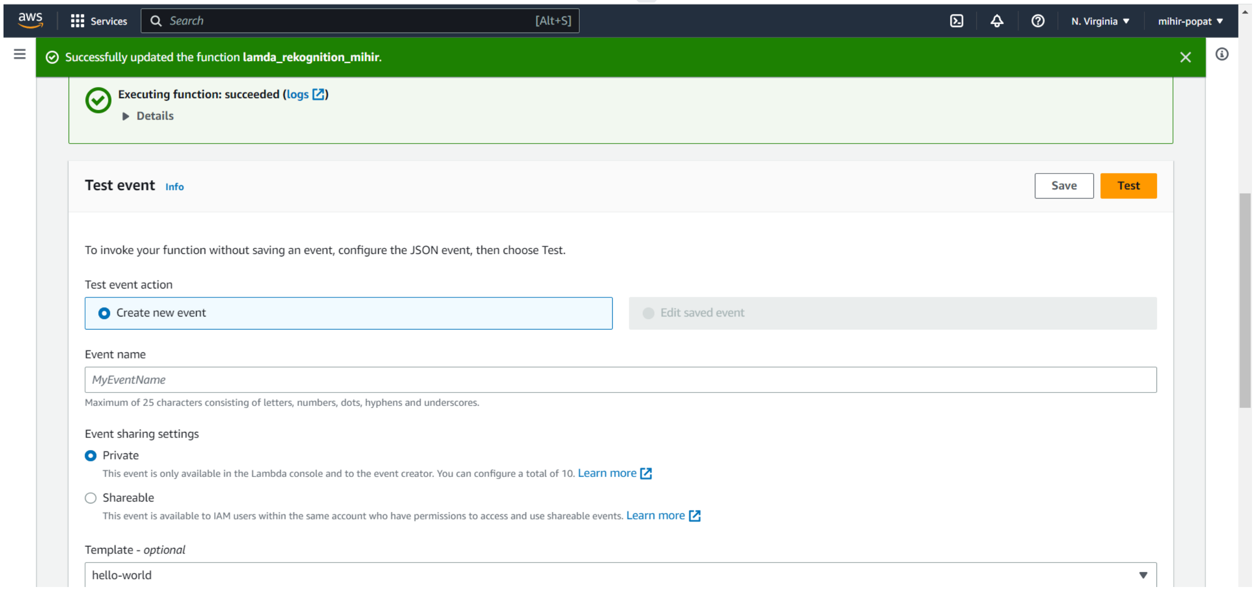
Log output

The section below shows the logging calls in your code. Click here to view the corresponding CloudWatch log group.

```
START RequestId: 5e031c9d-d2c2-4b39-b8db-3771cede2ceb Version: $LATEST
{'Labels': [{'Name': 'Person', 'Confidence': 99.99696350097656, 'Instances': [{'BoundingBox': {'Width': 0.1842155158519745, 'Height': 0.8476740717887878, 'Left': 0.4979473352432251, 'Top': 0.07987095415592194}, 'Confidence': 99.8228988647461}, {'BoundingBox': {'Width': 0.02768697775900364, 'Height': 0.05395684391260147, 'Left': 0.41400307416915894, 'Top': 0.376843124628067}, 'Confidence': 86.17980194091797}], 'Parents': [], 'Aliases': [{'Name': 'Human'}], 'Categories': [{'Name': 'Person Description'}]}, {'Name': 'Photography', 'Confidence': 99.99696350097656, 'Instances': [], 'Parents': [], 'Aliases': [{'Name': 'Photo'}], 'Categories': [{'Name': 'Hobbies and Interests'}]}, {'Name': 'Portrait', 'Confidence': 99.99696350097656, 'Instances': [], 'Parents': [{'Name': 'Face'}, {'Name': 'Head'}, {'Name': 'Person'}, {'Name': 'Photography'}], 'Aliases': [], 'Categories': [{'Name': 'Hobbies and Interests'}]}], 'LabelModelVersion': '3.0', 'ResponseMetadata': {'RequestId': '6e7bbf35-99a4-4445-8a93-e5cd54435c82', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amzn-requestid': '6e7bbf35-99a4-4445-8a93-e5cd54435c82', 'content-type': 'application/x-amz-json-1.1', 'content-length': '871', 'date': 'Sun, 14 May 2023 06:15:22 GMT'}, 'RetryAttempts': 0}}
END RequestId: 5e031c9d-d2c2-4b39-b8db-3771cede2ceb
REPORT RequestId: 5e031c9d-d2c2-4b39-b8db-3771cede2ceb  Duration: 3656.84 ms  Billed Duration: 3657 ms    Memory Size: 128 MB  Max Memory Used: 82 MB  Init Duration: 440.22 ms
```

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.



Successfully updated the function **lamda_rekognition_mihir**.

Executing function: succeeded (logs)

▸ Details

Test event Info

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

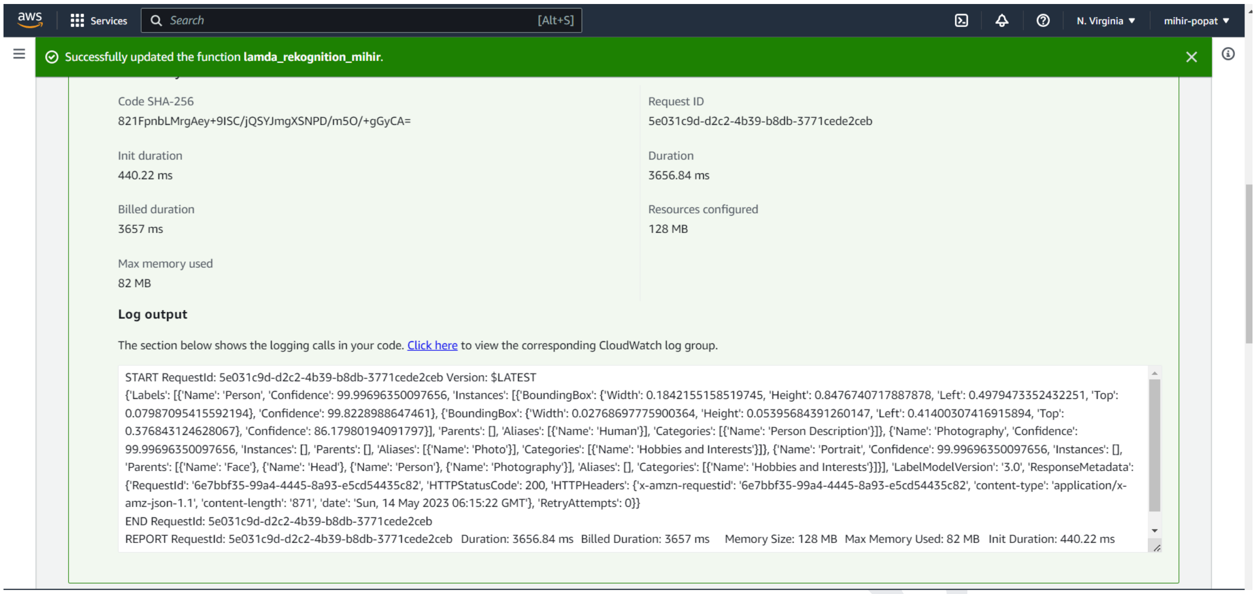
○ Shareable

This event is available to IAM users within the same account who have permissions to access and use shareable events. Learn more

Template - *optional*

hello-world ▾

- 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:



Successfully updated the function **lamda_rekognition_mihir**.

Code SHA-256
821FpnbLMrgAey+9ISC/jQSYJmgXSNPD/m5O/+gGyCA=

Request ID
5e031c9d-d2c2-4b39-b8db-3771cede2ceb

Init duration
440.22 ms

Duration
3656.84 ms

Billed duration
3657 ms

Resources configured
128 MB

Max memory used
82 MB

Log output

The section below shows the logging calls in your code. Click here to view the corresponding CloudWatch log group.

```
START RequestId: 5e031c9d-d2c2-4b39-b8db-3771cede2ceb Version: $LATEST
{'Labels': [{'Name': 'Person', 'Confidence': 99.99696350097656, 'Instances': [{'BoundingBox': {'Width': 0.1842155158519745, 'Height': 0.8476740717887878, 'Left': 0.4979473352432251, 'Top': 0.0798709541559219 4}, 'Confidence': 99.8228988647461}, {'BoundingBox': {'Width': 0.02768697775900364, 'Height': 0.05395684391260147, 'Left': 0.41400307416915894, 'Top': 0.376843124628067}, 'Confidence': 86.17980194091797}], 'Parents': [], 'Aliases': [{'Name': 'Human'}], 'Categories': [{'Name': 'Person Description'}]}, {'Name': 'Photography', 'Confidence': 99.99696350097656, 'Instances': [], 'Parents': [], 'Aliases': [{'Name': 'Photo'}], 'Categories': [{'Name': 'Hobbies and Interests'}]}, {'Name': 'Portrait', 'Confidence': 99.99696350097656, 'Instances': [], 'Parents': [{'Name': 'Face'}, {'Name': 'Head'}, {'Name': 'Person'}, {'Name': 'Photography'}], 'Aliases': [], 'Categories': [{'Name': 'Hobbies and Interests'}]}], 'LabelModelVersion': '3.0', 'ResponseMetadata': {'RequestId': '6e7bbf35-99a4-4445-8a93-e5cd54435c82', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amzn-requestid': '6e7bbf35-99a4-4445-8a93-e5cd54435c82', 'content-type': 'application/x-amz-json-1.1', 'content-length': '871', 'date': 'Sun, 14 May 2023 06:15:22 GMT'}, 'RetryAttempts': 0}}
END RequestId: 5e031c9d-d2c2-4b39-b8db-3771cede2ceb
REPORT RequestId: 5e031c9d-d2c2-4b39-b8db-3771cede2ceb  Duration: 3656.84 ms  Billed Duration: 3657 ms    Memory Size: 128 MB  Max Memory Used: 82 MB  Init Duration: 440.22 ms
```

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.