Lambda Assignment

Problem Statement:

You work for XYZ Corporation. Your corporation wants to launch a new web-based application and they do not want their servers to be running all the time. It should also be managed by AWS. Implement suitable solutions.

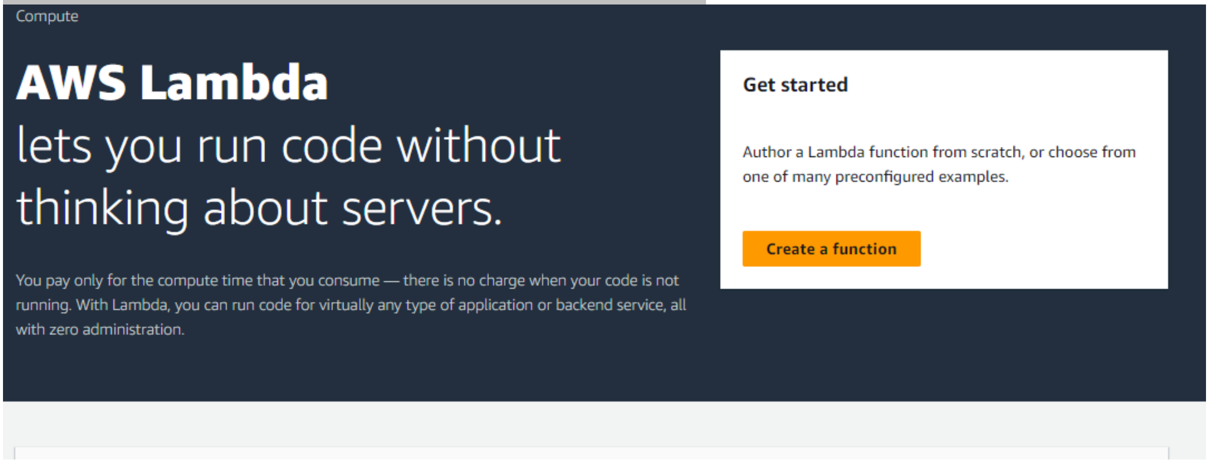
Tasks to be Performed:

1. Create a sample Python Lambda function.

2. Set the Lambda Trigger as SQS and send a message to test invocations.

Solutions:

Go to AWS > AWS Lambda > Create a function.



Compute

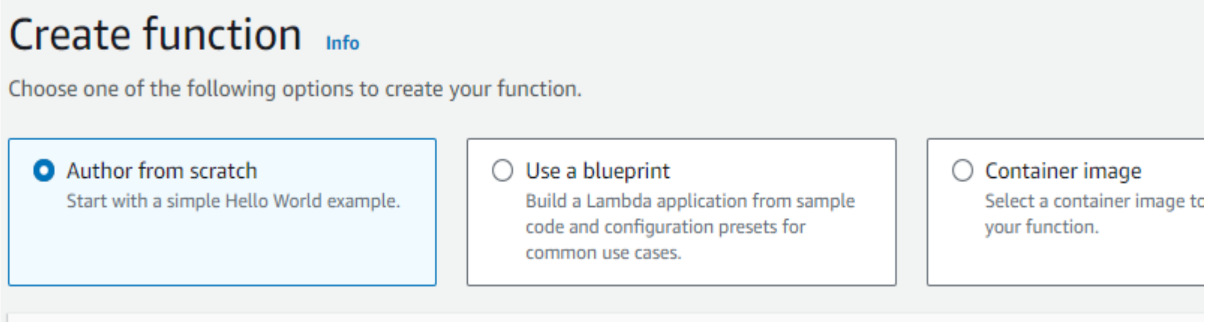
AWS Lambda
lets you run code without thinking about servers.

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.

Select author from Scratch



Create function Info

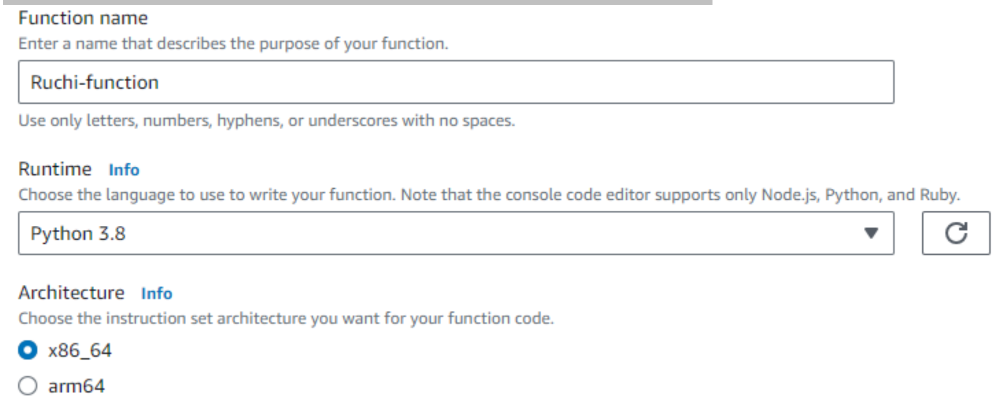
Choose one of the following options to create your function.

● 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 tc your function.

Give a function name and choose the language



Function name

Enter a name that describes the purpose of your function.

Ruchi-function

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 3.8

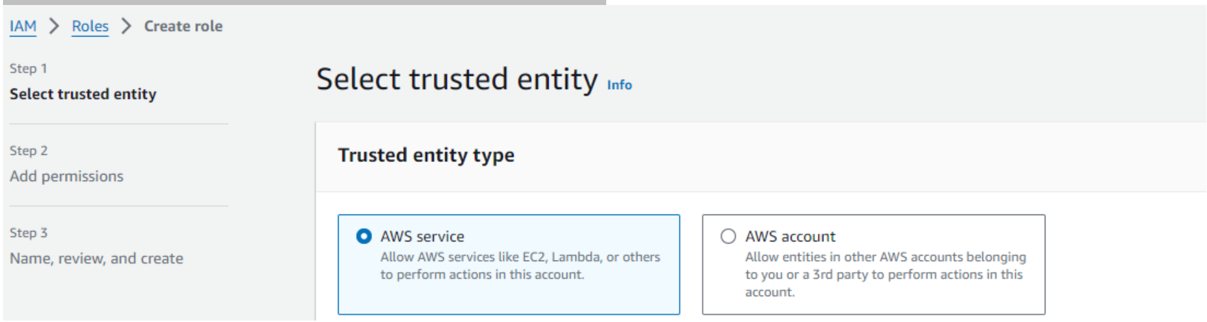
Architecture Info

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

● x86_64

○ arm64

Select the Trusted entity as AWS service



IAM > Roles > Create role

Step 1

Select trusted entity

Step 2

Add permissions

Step 3

Name, review, and create

Select trusted entity Info

Trusted entity type

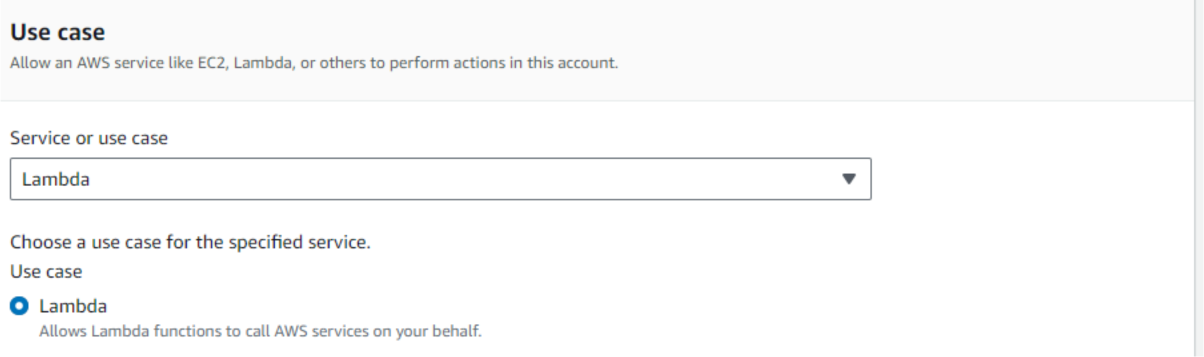
● AWS service

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.

Select the Use case as Lambda



Use case

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

Service or use case

Lambda

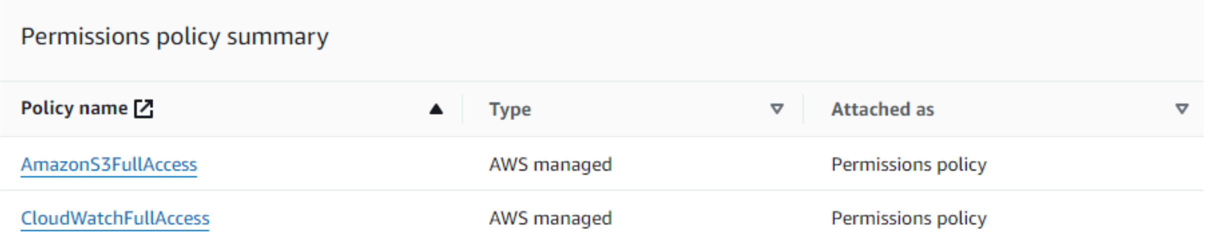
Choose a use case for the specified service.

Use case

● Lambda

Allows Lambda functions to call AWS services on your behalf.

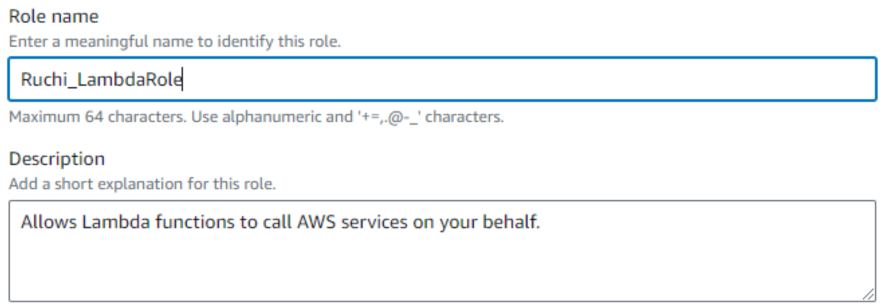
Select these 2 permissions policy



Permissions policy summary

Policy name ↗ ▲	Type ▽	Attached as ▽
AmazonS3FullAccess	AWS managed	Permissions policy
CloudWatchFullAccess	AWS managed	Permissions policy

Provide the Role Name



Role name
Enter a meaningful name to identify this role.

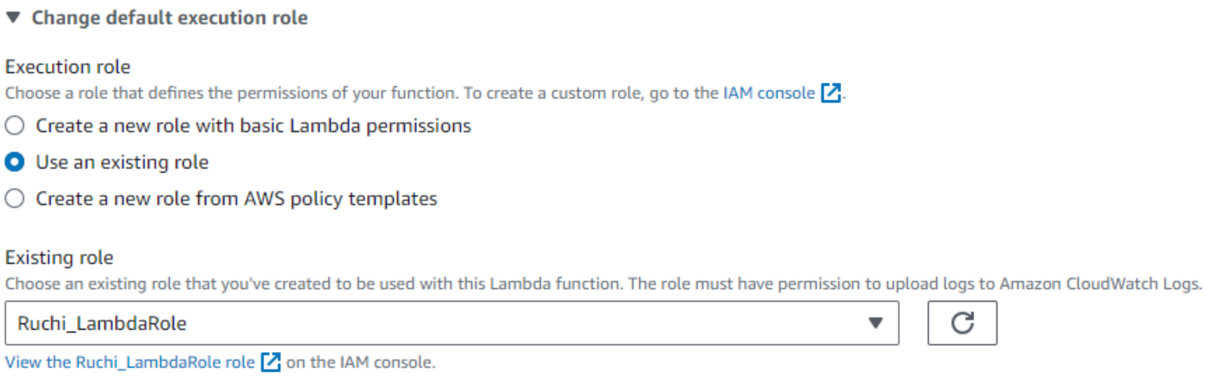
Ruchi_LambdaRole

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.

Create a new role and select here



▼ 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 ↗.

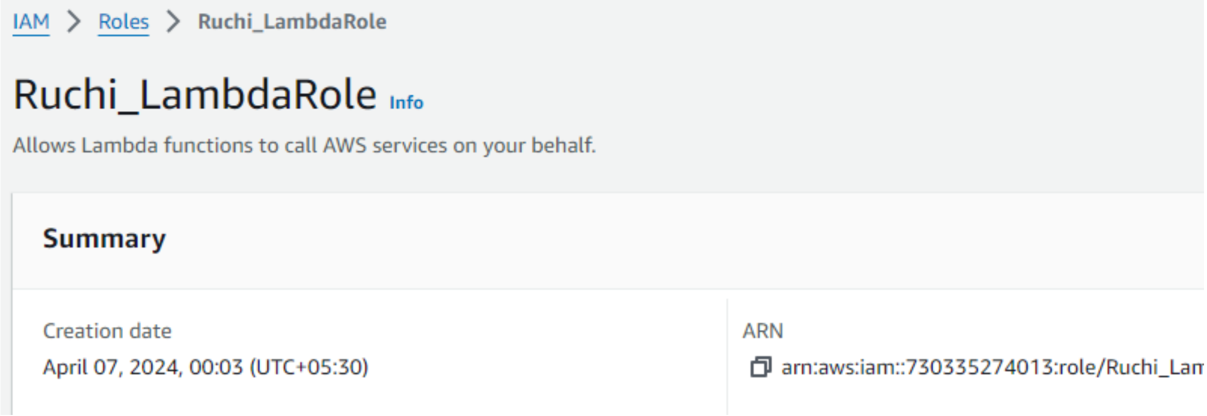
○ Create a new role with basic Lambda permissions
● Use an existing role
○ Create a new role from AWS policy templates

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.

Ruchi_LambdaRole ▼

View the Ruchi_LambdaRole role ↗ on the IAM console.

New Role which is created



Ruchi_LambdaRole Info

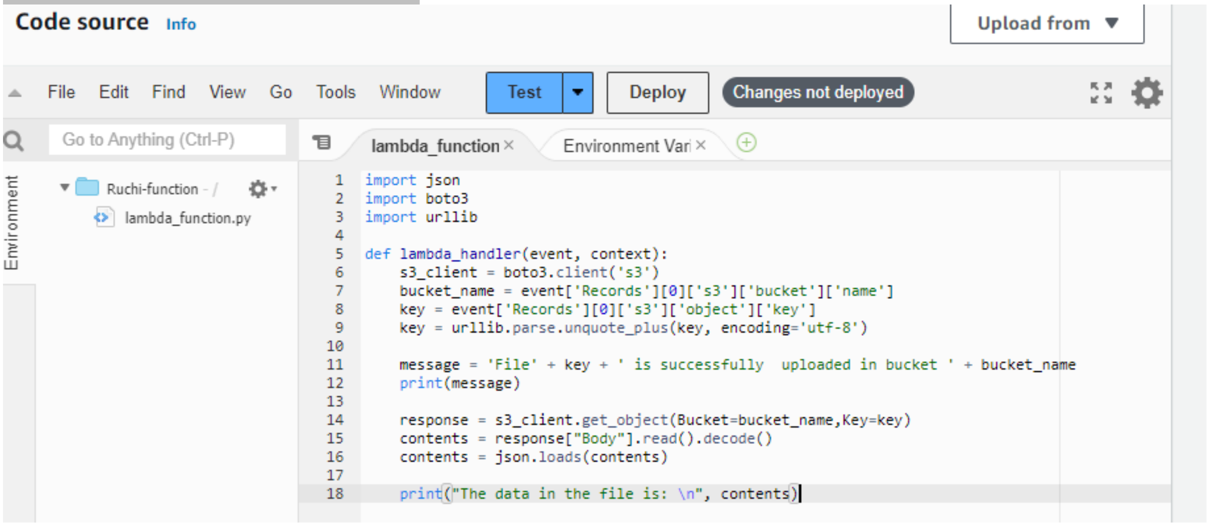
Allows Lambda functions to call AWS services on your behalf.

Summary

Creation date: April 07, 2024, 00:03 (UTC+05:30)

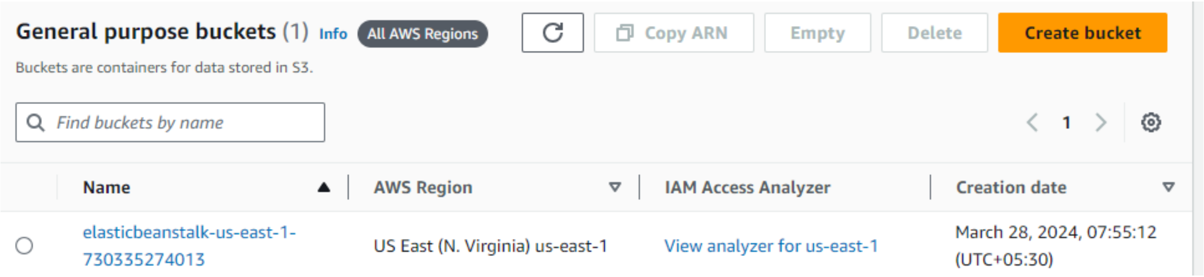
ARN: arn:aws:iam::730335274013:role/Ruchi_Lam

Go to code source > Deploy



```python
import json
import boto3
import urllib

def lambda_handler(event, context):
    s3_client = boto3.client('s3')
    bucket_name = event['Records'][0]['s3']['bucket']['name']
    key = event['Records'][0]['s3']['object']['key']
    key = urllib.parse.unquote_plus(key, encoding='utf-8')

    message = 'File' + key + ' is successfully  uploaded in bucket ' + bucket_name
    print(message)

    response = s3_client.get_object(Bucket=bucket_name,Key=key)
    contents = response["Body"].read().decode()
    contents = json.loads(contents)

    print("The data in the file is: \n", contents)
```

Create a S3 bucket to trigger the Lambda function.

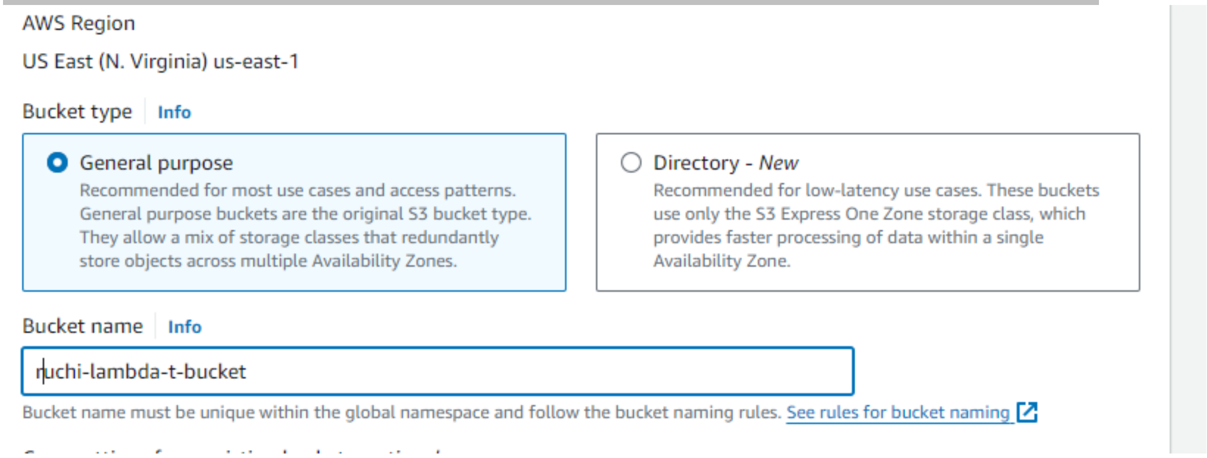


General purpose buckets (1) Info All AWS Regions

Buckets are containers for data stored in S3.

	Name	AWS Region	IAM Access Analyzer	Creation date
○	elasticbeanstalk-us-east-1-730335274013	US East (N. Virginia) us-east-1	View analyzer for us-east-1	March 28, 2024, 07:55:12 (UTC+05:30)

Go to AmazonS3 > Buckets > create buckets, Give a proper bucket name.



AWS Region

US East (N. Virginia) us-east-1

Bucket type Info

● General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

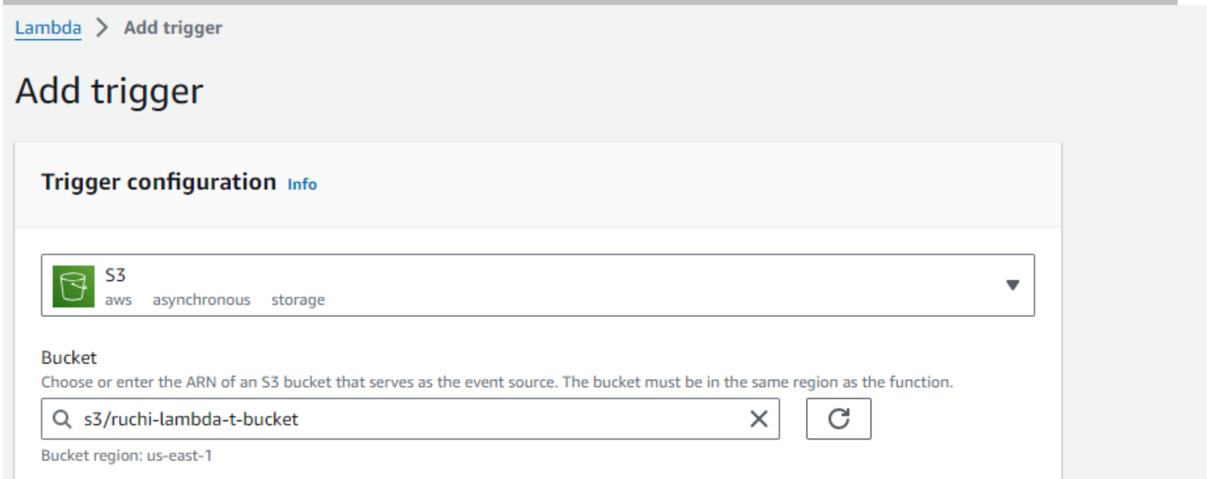
○ Directory - *New*
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name Info

ruchi-lambda-t-bucket

Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming

Add trigger go to Lambda > Add trigger select the created Bucket in Source S3



Add trigger

Trigger configuration Info

S3
aws asynchronous storage

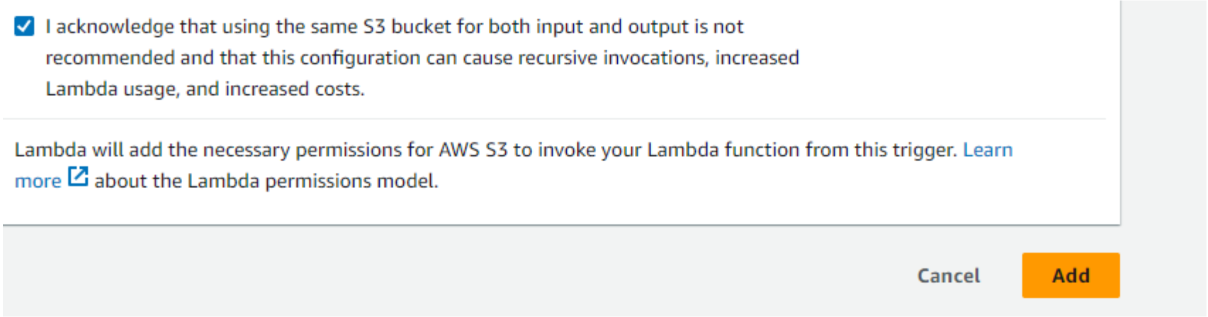
Bucket

Choose or enter the ARN of an S3 bucket that serves as the event source. The bucket must be in the same region as the function.

s3/ruchi-lambda-t-bucket

Bucket region: us-east-1

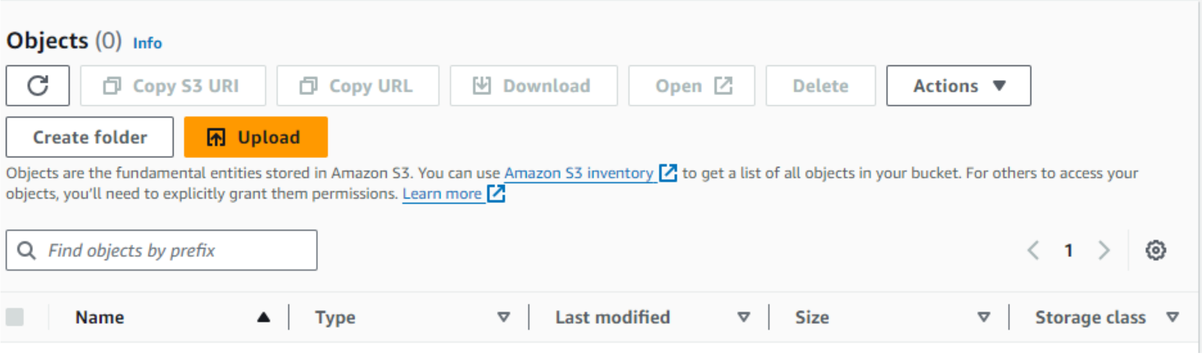
Acknowledge it and click Add.



☑ I acknowledge that using the same S3 bucket for both input and output is not recommended and that this configuration can cause recursive invocations, increased Lambda usage, and increased costs.

Lambda will add the necessary permissions for AWS S3 to invoke your Lambda function from this trigger. Learn more about the Lambda permissions model.

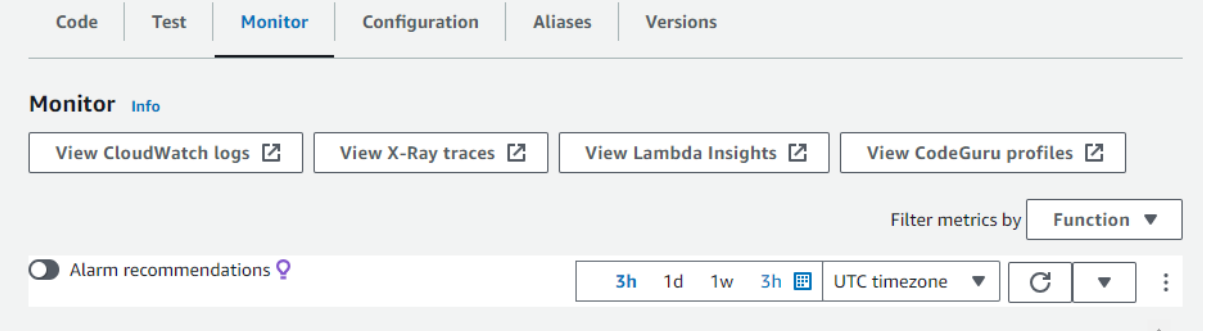
Upload a file into the bucket.



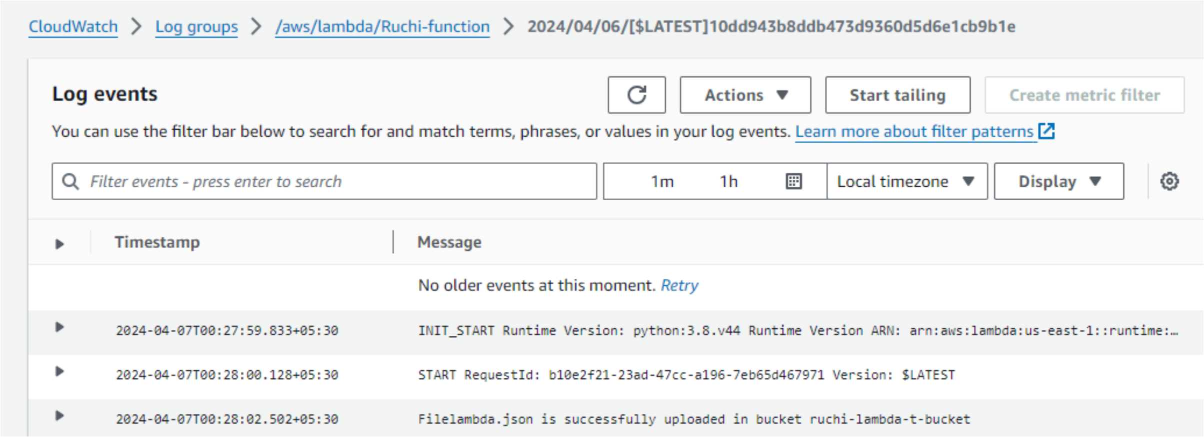
Objects (0) Info

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

Go to Monitor > View cloud watch logs > log groups > log stream



Watch out the comments.


CloudWatch > Log groups > /aws/lambda/Ruchi-function > 2024/04/06/[$LATEST]10dd943b8ddb473d9360d5d6e1cb9b1e
Log events
Actions
Start tailing
Create metric filter
Filter events - press enter to search
1m
1h
Local timezone
Display
Timestamp
Message

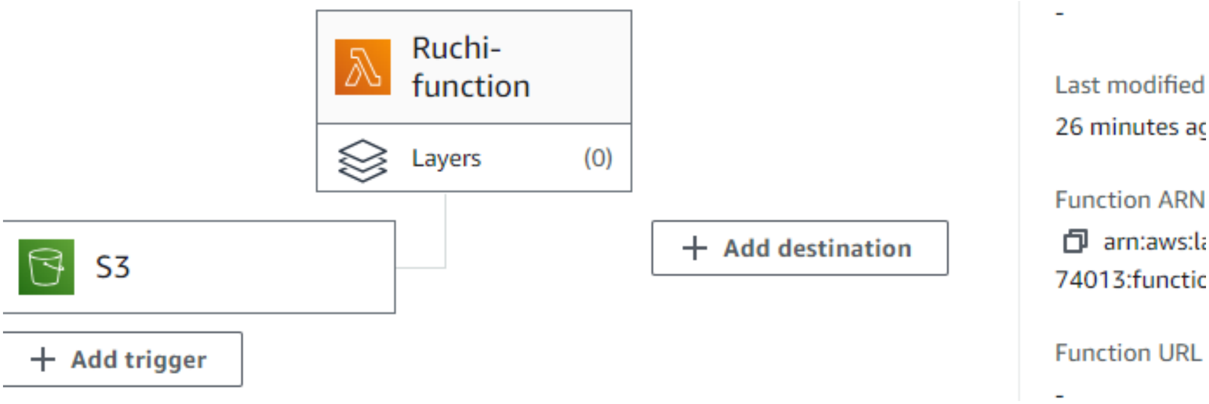
You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns

No older events at this moment. Retry

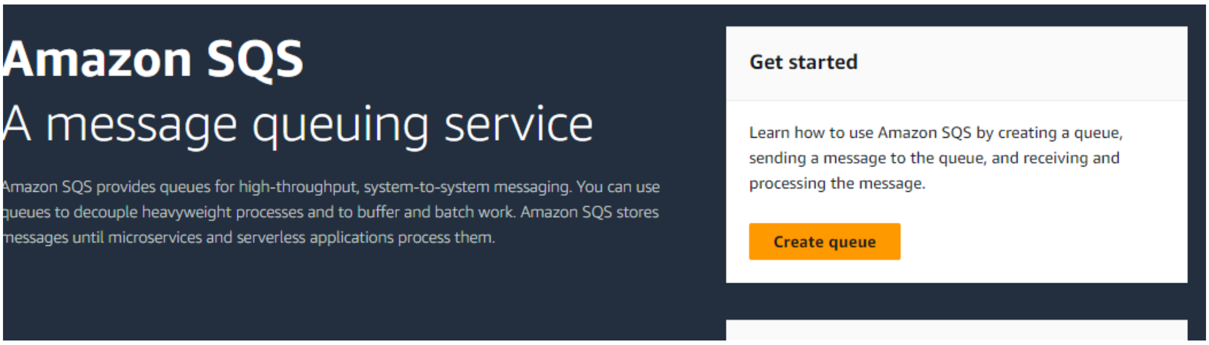
```
2024-04-07T00:27:59.833+05:30    INIT_START Runtime Version: python:3.8.v44 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:…
2024-04-07T00:28:00.128+05:30    START RequestId: b10e2f21-23ad-47cc-a196-7eb65d467971 Version: $LATEST
2024-04-07T00:28:02.502+05:30    Filelambda.json is successfully uploaded in bucket ruchi-lambda-t-bucket
```

Now Add Destination

Destination is added to send the message to SQS after the Lambda function triggered successfully.



Go to AWS > SQS Service > Create queue



Amazon SQS

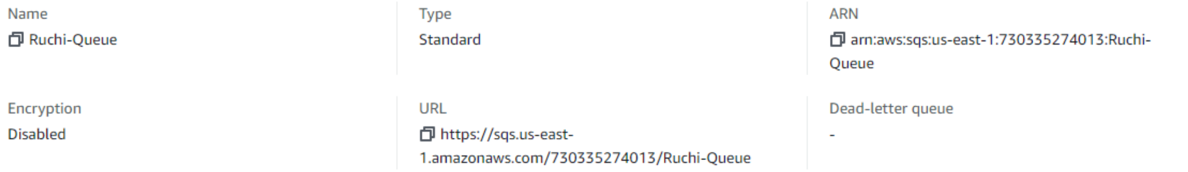
A message queuing service

Amazon SQS provides queues for high-throughput, system-to-system messaging. You can use queues to decouple heavyweight processes and to buffer and batch work. Amazon SQS stores messages until microservices and serverless applications process them.

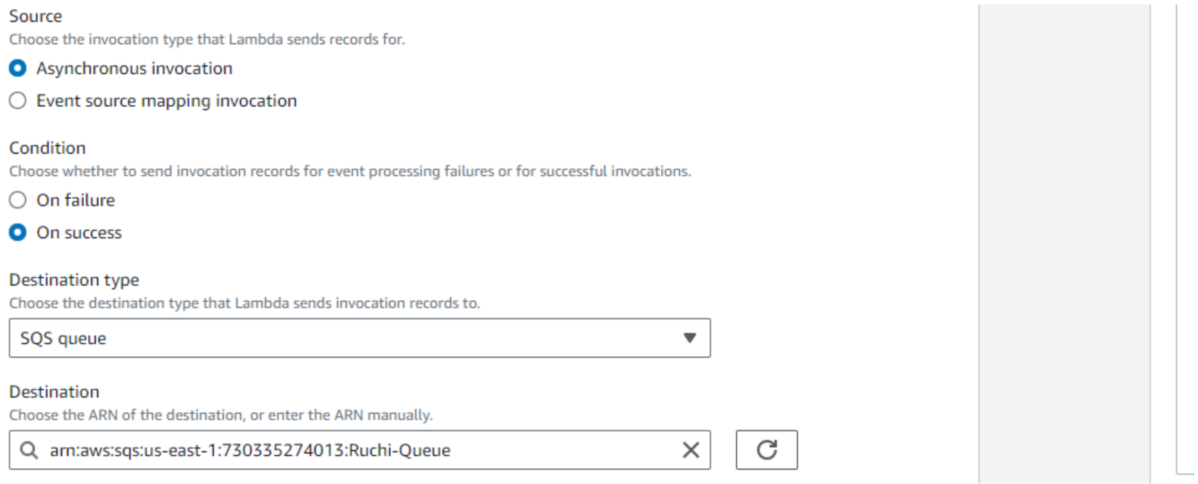
Get started

Learn how to use Amazon SQS by creating a queue, sending a message to the queue, and receiving and processing the message.

Give a name and select the below configurations



Name Ruchi-Queue	Type Standard	ARN arn:aws:sqs:us-east-1:730335274013:Ruchi-Queue
Encryption Disabled	URL https://sqs.us-east-1.amazonaws.com/730335274013/Ruchi-Queue	Dead-letter queue -



Source

Choose the invocation type that Lambda sends records for.

● Asynchronous invocation

○ Event source mapping invocation

Condition

Choose whether to send invocation records for event processing failures or for successful invocations.

○ On failure

● On success

Destination type

Choose the destination type that Lambda sends invocation records to.

SQS queue ▼

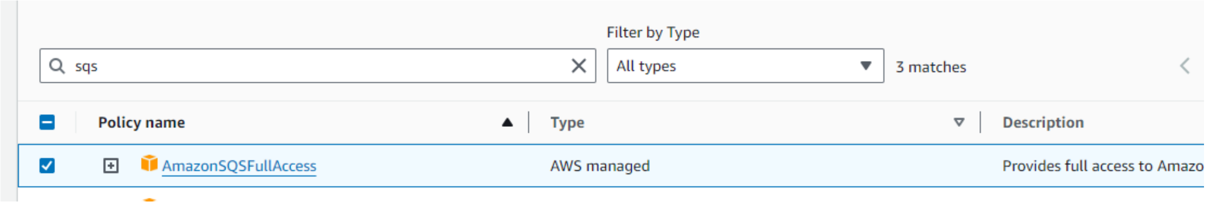
Destination

Choose the ARN of the destination, or enter the ARN manually.

arn:aws:sqs:us-east-1:730335274013:Ruchi-Queue

Go to IAM Roles > Roles

Add Permission - Attach Policy - AmazonSQSFullAccess



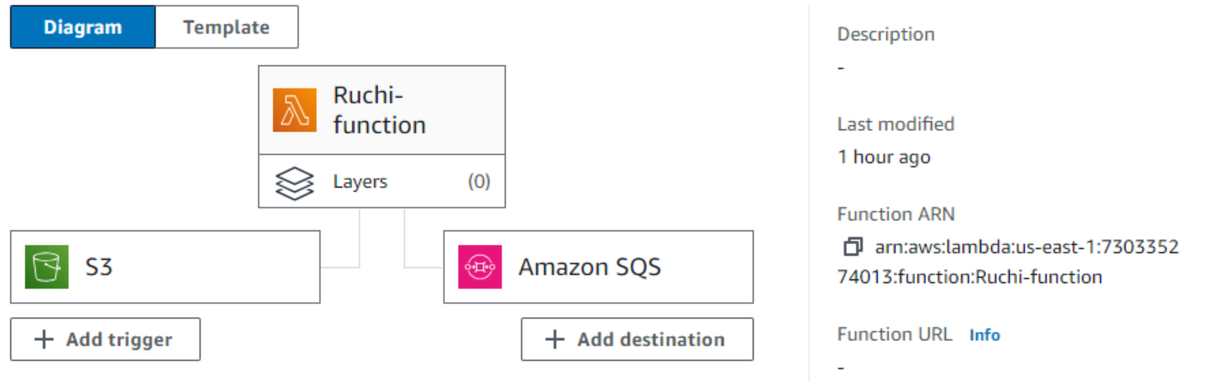
sqs

Filter by Type: All types

3 matches

☐	Policy name ▲	Type	Description
☑	⊞ AmazonSQSFullAccess	AWS managed	Provides full access to Amazo

Trigger and Destination is added



Description

-

Last modified

1 hour ago

Function ARN

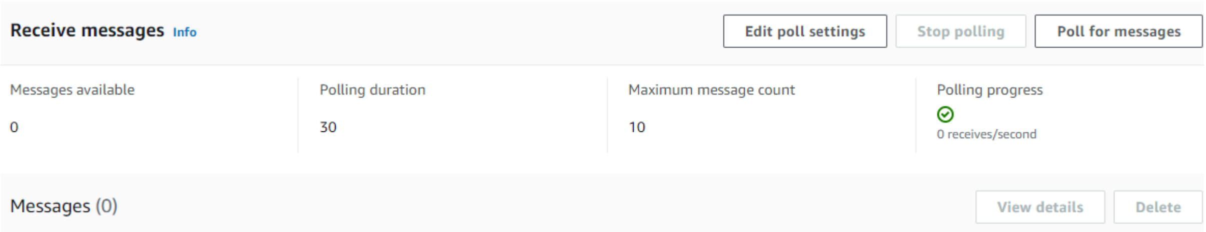
arn:aws:lambda:us-east-1:730335274013:function:Ruchi-function

Function URL Info

-

Check the Message in SQA, Click Poll for messages

Messages are not available

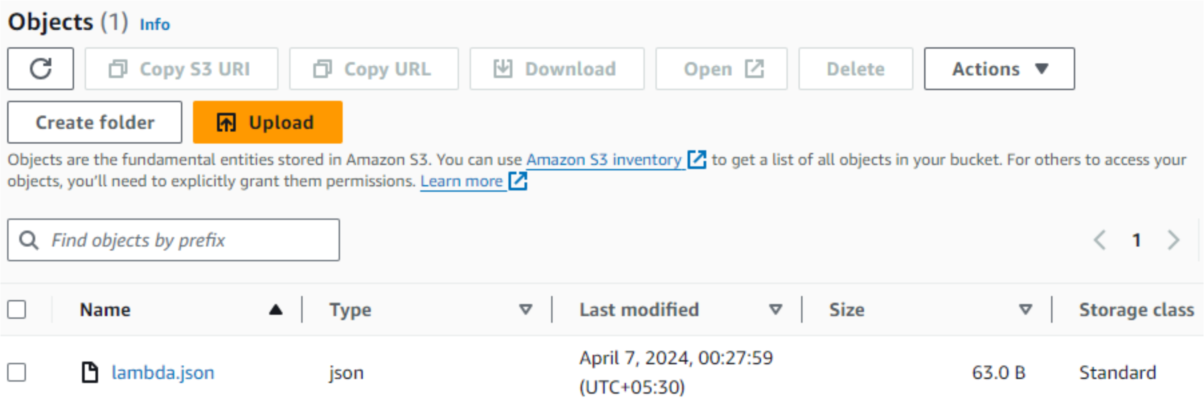


Messages available	Polling duration	Maximum message count	Polling progress
0	30	10	✓ 0 receives/second

Messages (0)

Hence Trigger the Lambda function once again

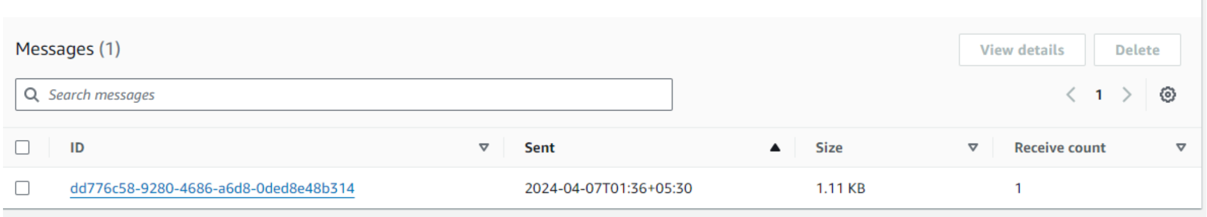
Go to Lambda Function > trigger > upload a Json file to S3 bucket



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

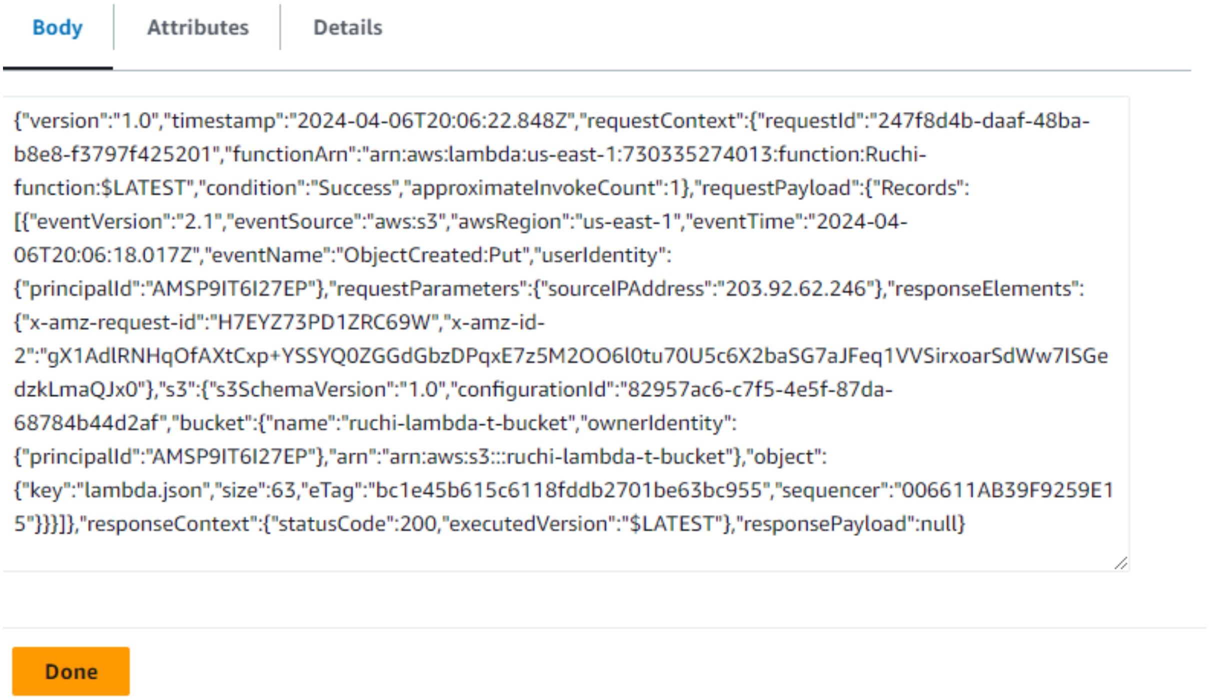
□	Name ▲	Type	Last modified	Size	Storage class
□	lambda.json	json	April 7, 2024, 00:27:59 (UTC+05:30)	63.0 B	Standard

Now if you poll the messages you will be able to find



□	ID	Sent ▲	Size	Receive count
□	dd776c58-9280-4686-a6d8-0ded8e48b314	2024-04-07T01:36+05:30	1.11 KB	1

Check the body, Attributes and Details for more information



```
{"version":"1.0","timestamp":"2024-04-06T20:06:22.848Z","requestContext":{"requestId":"247f8d4b-daaf-48ba-
b8e8-f3797f425201","functionArn":"arn:aws:lambda:us-east-1:730335274013:function:Ruchi-
function:$LATEST","condition":"Success","approximateInvokeCount":1},"requestPayload":{"Records":
[{"eventVersion":"2.1","eventSource":"aws:s3","awsRegion":"us-east-1","eventTime":"2024-04-
06T20:06:18.017Z","eventName":"ObjectCreated:Put","userIdentity":
{"principalId":"AMSP9IT6I27EP"},"requestParameters":{"sourceIPAddress":"203.92.62.246"},"responseElements":
{"x-amz-request-id":"H7EYZ73PD1ZRC69W","x-amz-id-
2":"gX1AdlRNHqOfAXtCxp+YSSYQ0ZGGdGbzDPqxE7z5M2OO6l0tu70U5c6X2baSG7aJFeq1VVSirxoarSdWw7ISGe
dzkLmaQJx0"},"s3":{"s3SchemaVersion":"1.0","configurationId":"82957ac6-c7f5-4e5f-87da-
68784b44d2af","bucket":{"name":"ruchi-lambda-t-bucket","ownerIdentity":
{"principalId":"AMSP9IT6I27EP"},"arn":"arn:aws:s3:::ruchi-lambda-t-bucket"},"object":
{"key":"lambda.json","size":63,"eTag":"bc1e45b615c6118fddb2701be63bc955","sequencer":"006611AB39F9259E1
5"}}}]},"responseContext":{"statusCode":200,"executedVersion":"$LATEST"},"responsePayload":null}
```

===