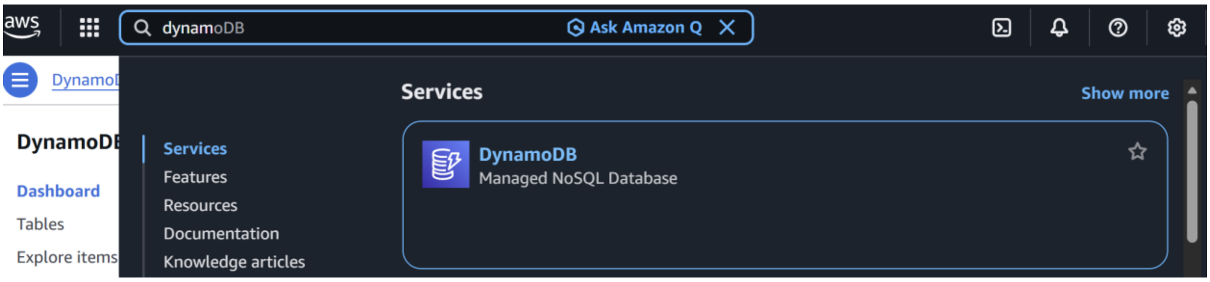
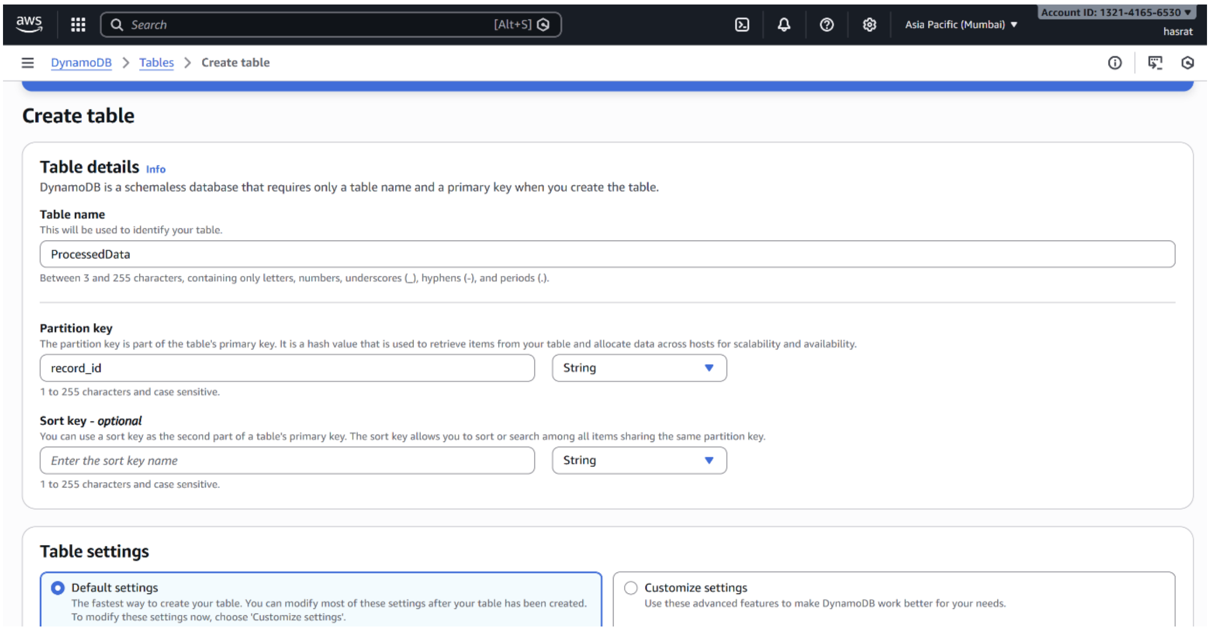
Event-Driven Data Processing Pipeline on AWS

1. Go to dynamoDB



2. Create a table named ProcessedData


aws
Search
[Alt+S]
Asia Pacific (Mumbai) ▼
Account ID: 1321-4165-6530 ▼
hasrat
DynamoDB > Tables > Create table

Create table

Table details Info

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name

This will be used to identify your table.

ProcessedData

Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.).

Partition key

The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.

record_id | String ▼

1 to 255 characters and case sensitive.

Sort key - *optional*

You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.

Enter the sort key name | String ▼

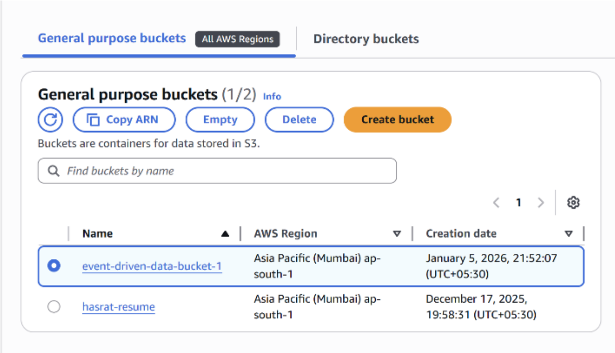
1 to 255 characters and case sensitive.

Table settings

● Default settings
The fastest way to create your table. You can modify most of these settings after your table has been created. To modify these settings now, choose 'Customize settings'.

○ Customize settings
Use these advanced features to make DynamoDB work better for your needs.

3. Create a bucket

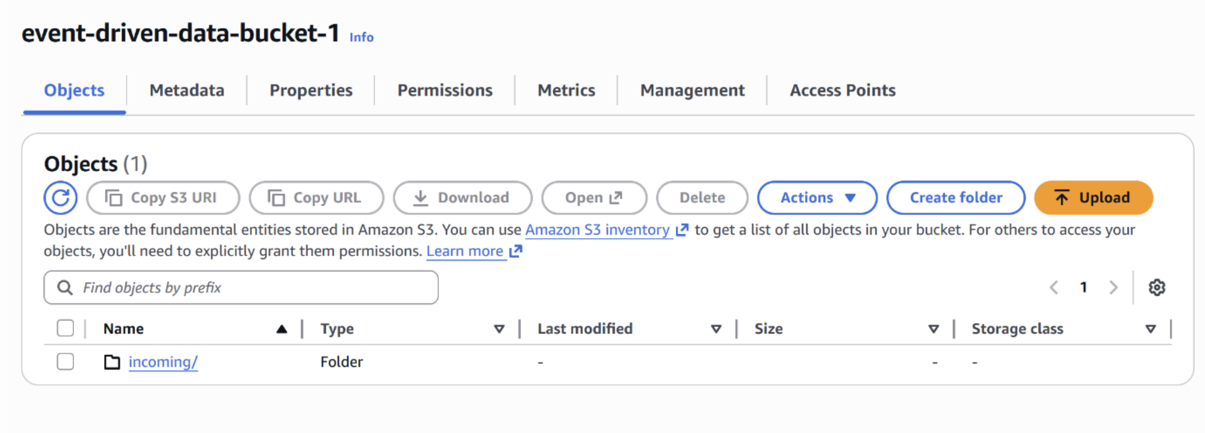


General purpose buckets (1/2) Info

Buckets are containers for data stored in S3.

	Name ▲	AWS Region	Creation date
●	event-driven-data-bucket-1	Asia Pacific (Mumbai) ap-south-1	January 5, 2026, 21:52:07 (UTC+05:30)
○	hasrat-resume	Asia Pacific (Mumbai) ap-south-1	December 17, 2025, 19:58:31 (UTC+05:30)

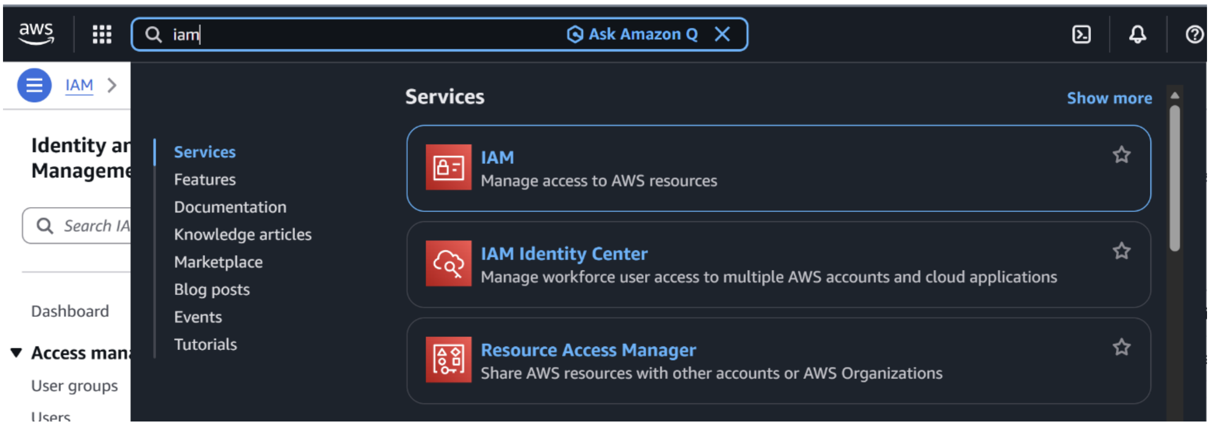
4. Inside the bucket create a folder named incoming (optional)



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
□	incoming/	Folder	-	-	-

5. Go to IAM

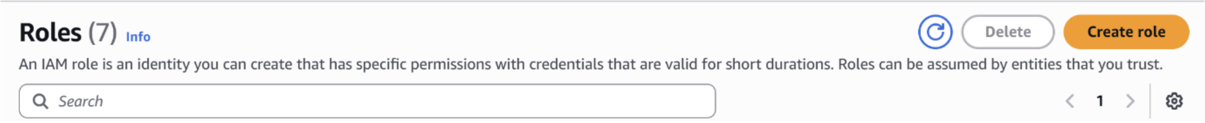


IAM
Manage access to AWS resources

IAM Identity Center
Manage workforce user access to multiple AWS accounts and cloud applications

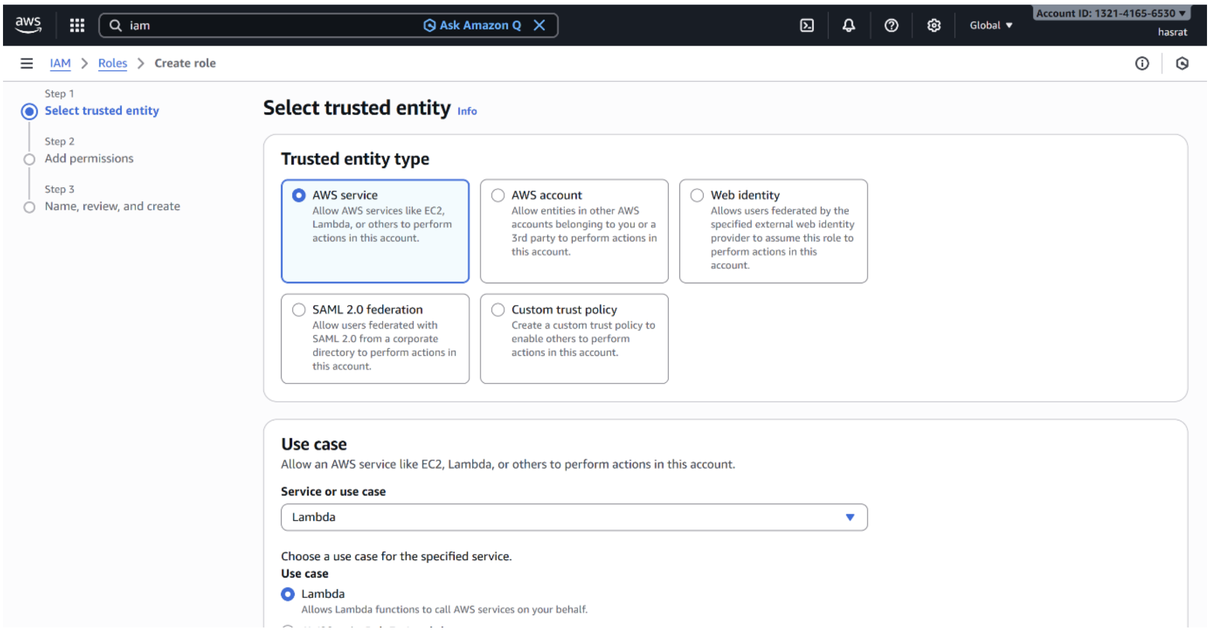
Resource Access Manager
Share AWS resources with other accounts or AWS Organizations

6. Click Create Role:



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

7. Select AWS service and Lambda in Use case:



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.

Service or use case

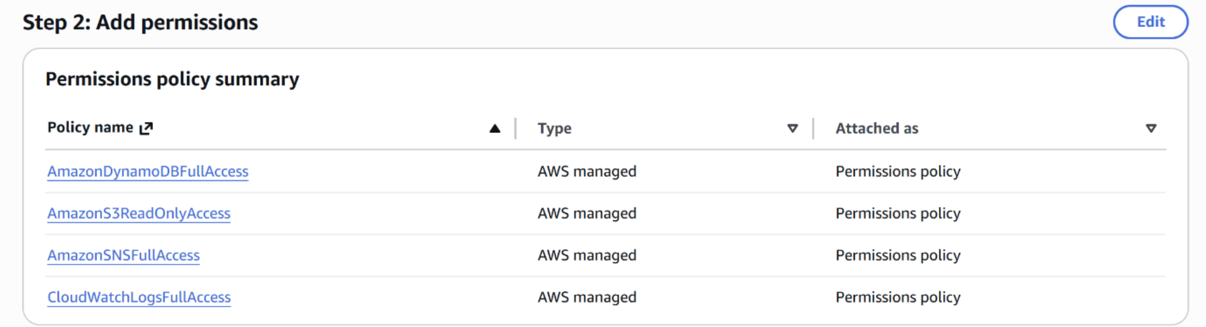
Lambda ▾

Choose a use case for the specified service.

Use case

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

8. Give name and Give Following Permissions:

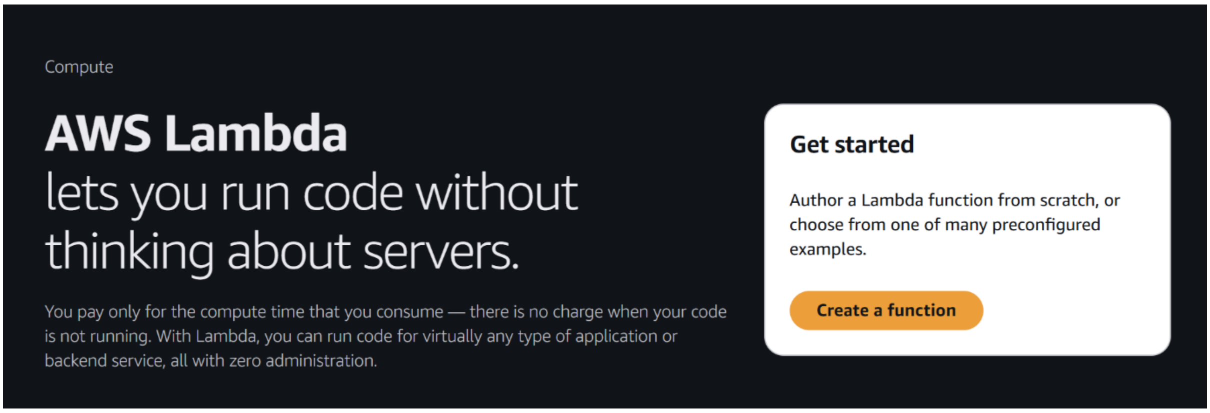


Step 2: Add permissions

Permissions policy summary

Policy name	Type	Attached as
AmazonDynamoDBFullAccess	AWS managed	Permissions policy
AmazonS3ReadOnlyAccess	AWS managed	Permissions policy
AmazonSNSFullAccess	AWS managed	Permissions policy
CloudWatchLogsFullAccess	AWS managed	Permissions policy

9. Now, Go to AWS Lambda and Create function For S3 EVENT TRIGGER:



Compute

AWS Lambda

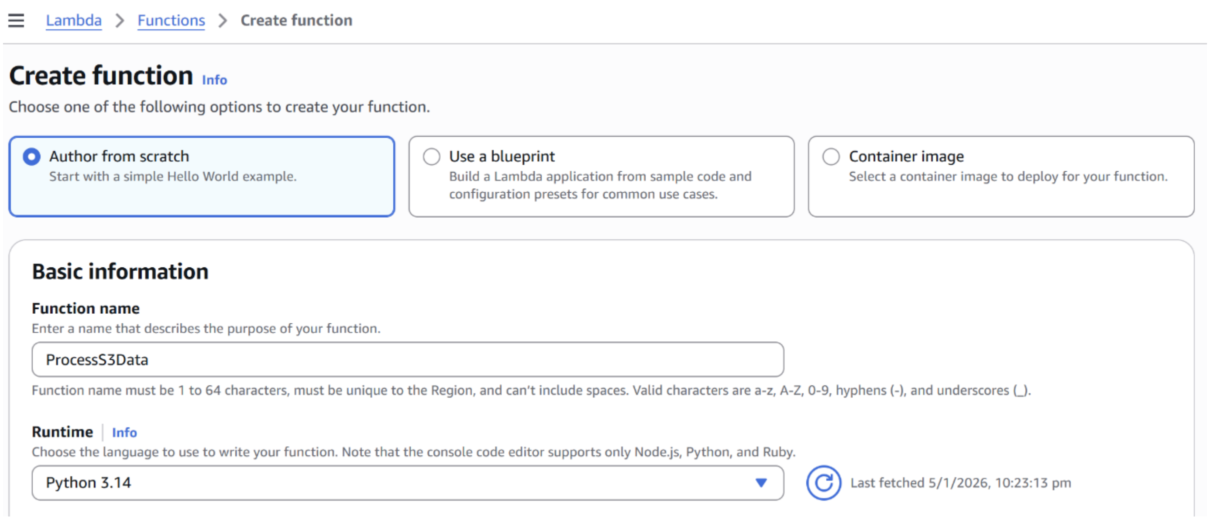
lets you run code without thinking about servers.

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.

10. Give Name to the function and Runtime as Python:



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 to deploy for your function.

Basic information

Function name
Enter a name that describes the purpose of your function.

ProcessS3Data

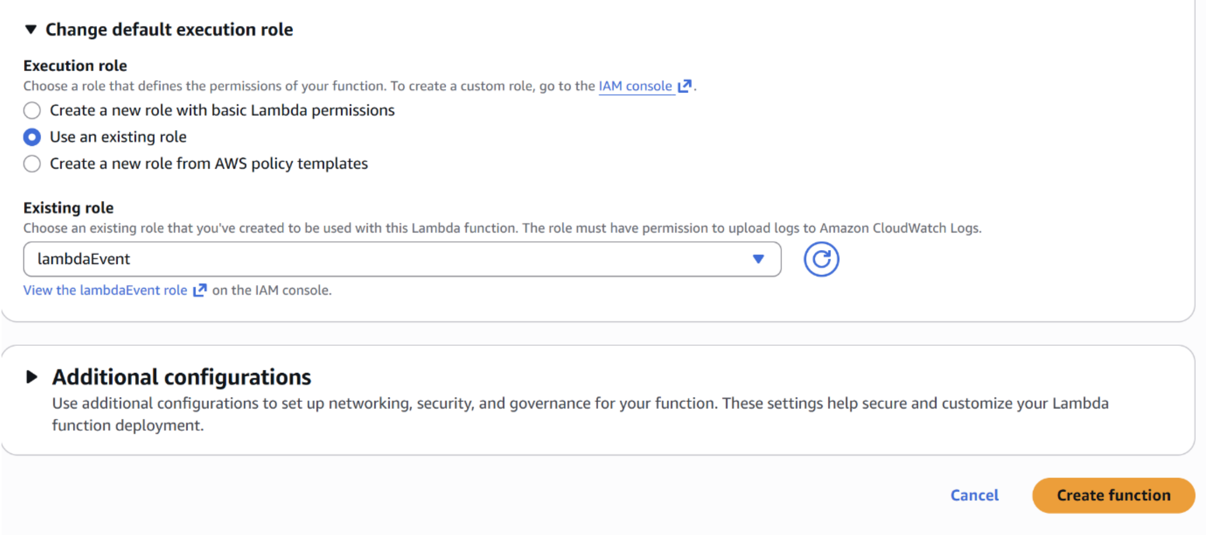
Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

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 3.14 ▼

Last fetched 5/1/2026, 10:23:13 pm

11. In exexution role select existing role and use LambdaEvent role we created and click Create function:


▼ 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

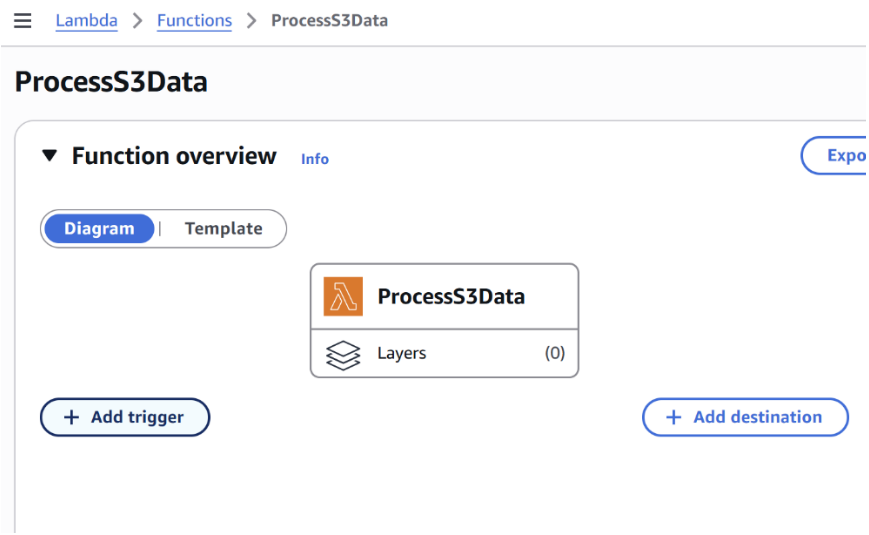
Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

lambdaEvent ▼

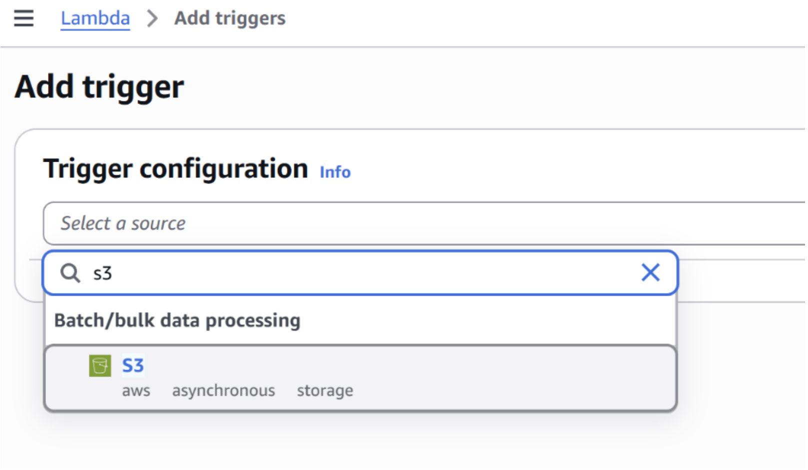
View the lambdaEvent role on the IAM console.

▶ Additional configurations
Use additional configurations to set up networking, security, and governance for your function. These settings help secure and customize your Lambda function deployment.

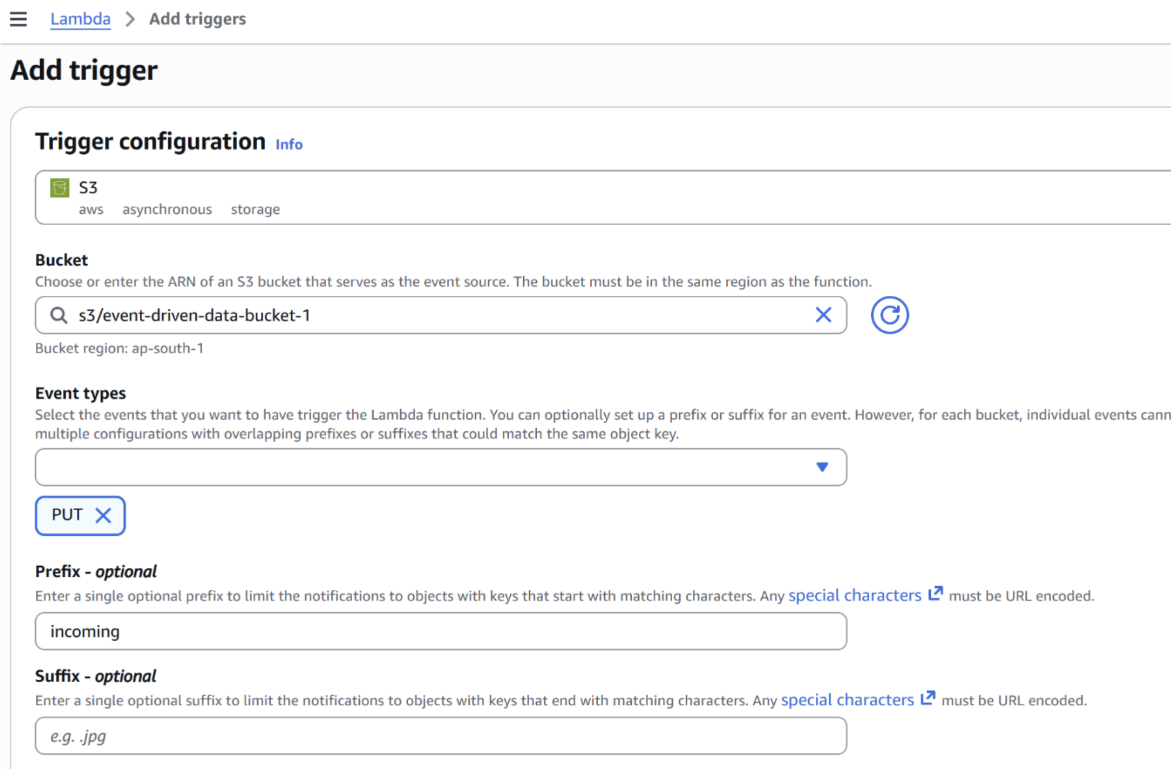
12. Go inside the created Function and Click Add Trigger:



13. Select S3:



14. Select our created Bucket and select Put in Event Types



Lambda > Add triggers

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/event-driven-data-bucket-1

Bucket region: ap-south-1

Event types
Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cann multiple configurations with overlapping prefixes or suffixes that could match the same object key.

PUT ✕

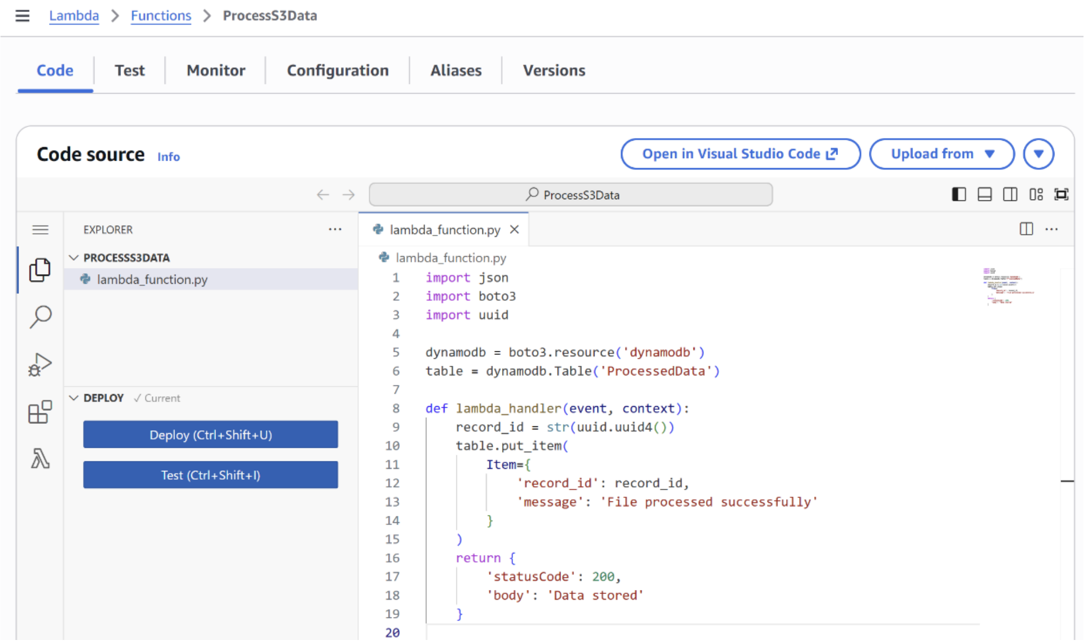
Prefix - *optional*
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters. Any special characters must be URL encoded.

incoming

Suffix - *optional*
Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters. Any special characters must be URL encoded.

e.g. .jpg

15. Go inside the function and Scroll down , Inside the Code Tab write the Code we want to trigger in response to event:



Lambda > Functions > ProcessS3Data

Code | Test | Monitor | Configuration | Aliases | Versions

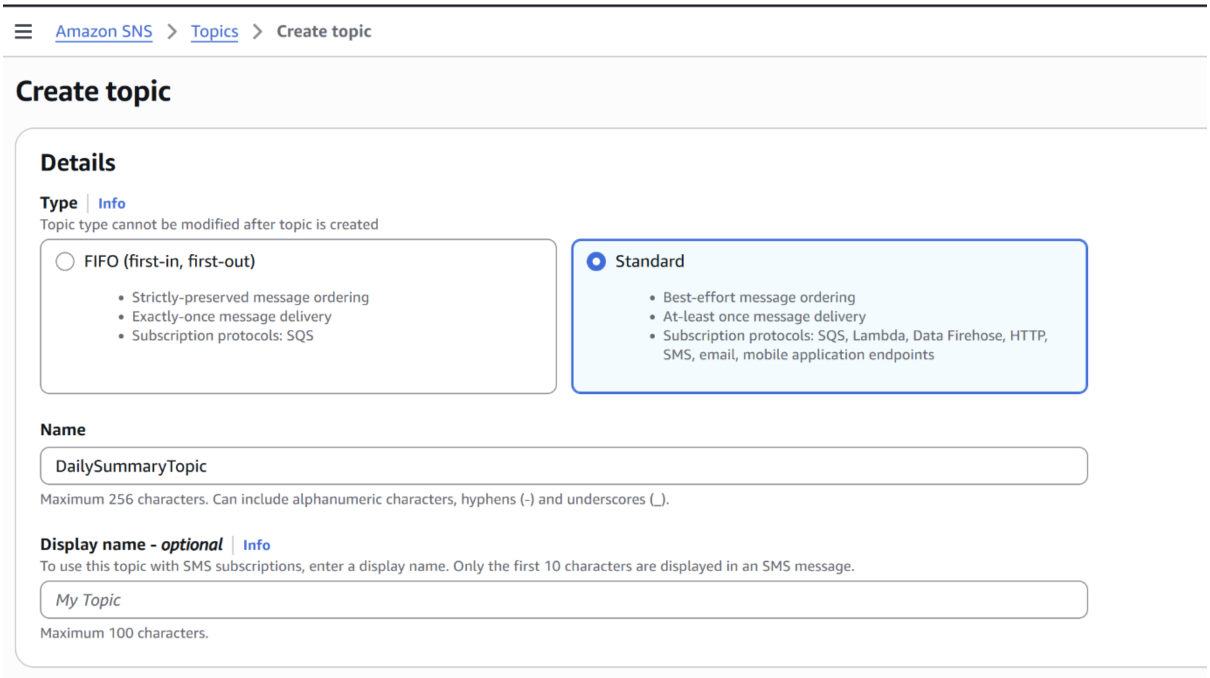
Code source Info

```python
import json
import boto3
import uuid

dynamodb = boto3.resource('dynamodb')
table = dynamodb.Table('ProcessedData')

def lambda_handler(event, context):
    record_id = str(uuid.uuid4())
    table.put_item(
        Item={
            'record_id': record_id,
            'message': 'File processed successfully'
        }
    )
    return {
        'statusCode': 200,
        'body': 'Data stored'
    }
```

16. Now go to SNS Topics and Create a New Standard Topic:



Create topic

Details

Type | Info

Topic type cannot be modified after topic is created

○ FIFO (first-in, first-out)

- Strictly-preserved message ordering
- Exactly-once message delivery
- Subscription protocols: SQS

● Standard

- Best-effort message ordering
- At-least once message delivery
- Subscription protocols: SQS, Lambda, Data Firehose, HTTP, SMS, email, mobile application endpoints

Name

DailySummaryTopic

Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (_).

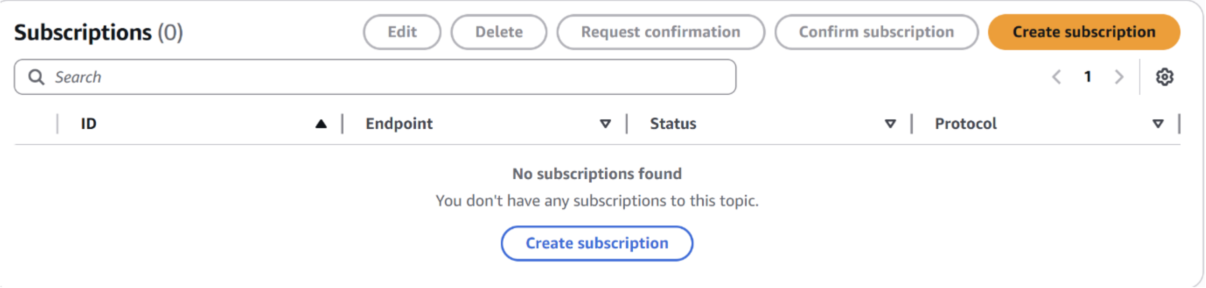
Display name - *optional* | Info

To use this topic with SMS subscriptions, enter a display name. Only the first 10 characters are displayed in an SMS message.

My Topic

Maximum 100 characters.

17. Create new Subscription inside SNS topics:



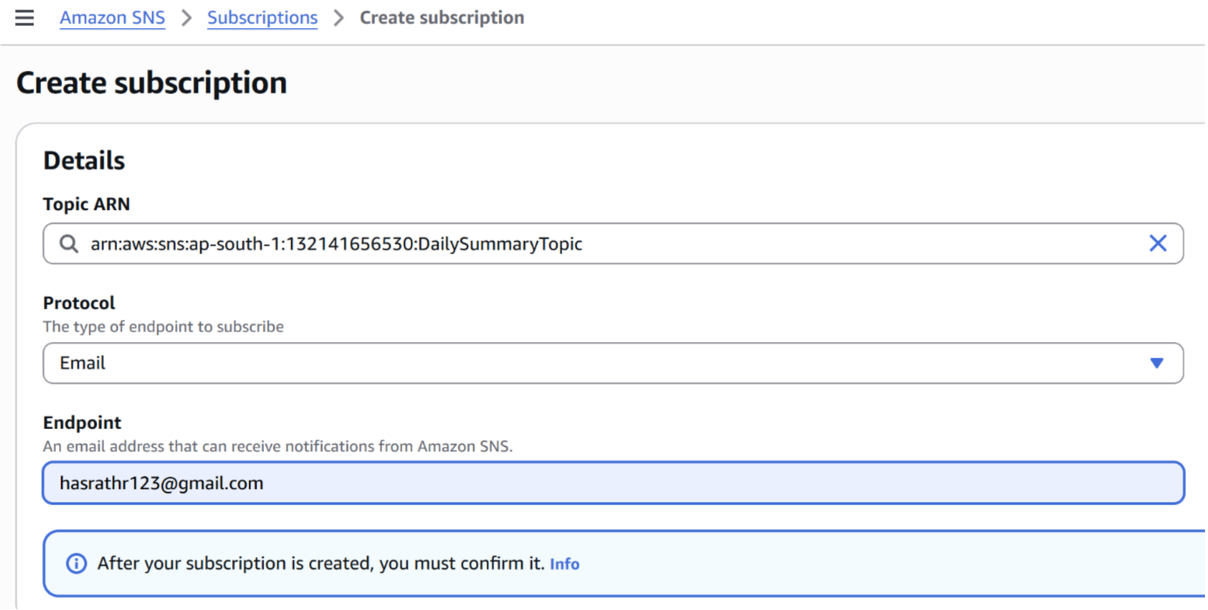
Subscriptions (0)

ID	Endpoint	Status	Protocol

No subscriptions found

You don't have any subscriptions to this topic.

18. Select ARN , Protocal as Email And enter email:



Create subscription

Details

Topic ARN

arn:aws:sns:ap-south-1:132141656530:DailySummaryTopic

Protocol

The type of endpoint to subscribe

Email

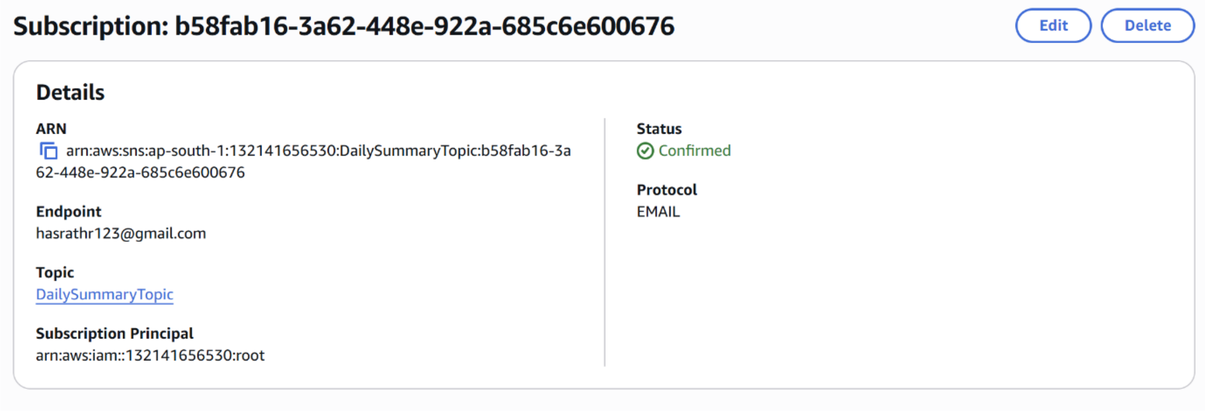
Endpoint

An email address that can receive notifications from Amazon SNS.

hasrathr123@gmail.com

ⓘ After your subscription is created, you must confirm it. Info

19. Now Give confirmation for the email through email:



Subscription: b58fab16-3a62-448e-922a-685c6e600676

Details

ARN

arn:aws:sns:ap-south-1:132141656530:DailySummaryTopic:b58fab16-3a62-448e-922a-685c6e600676

Endpoint

hasrathr123@gmail.com

Topic

DailySummaryTopic

Subscription Principal

arn:aws:iam::132141656530:root

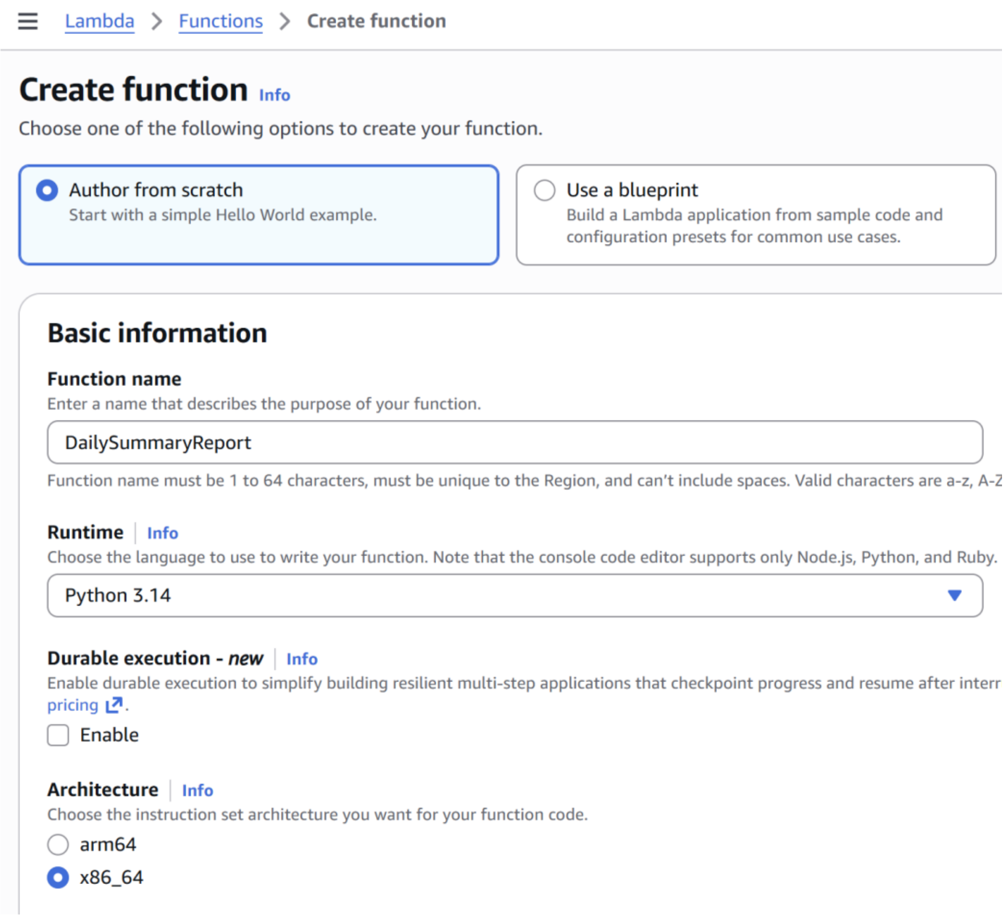
Status

✓ Confirmed

Protocol

EMAIL

20. Create New Function for Email Trigger:



Lambda > Functions > Create function

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.

Basic information

Function name
Enter a name that describes the purpose of your function.

DailySummaryReport

Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z

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

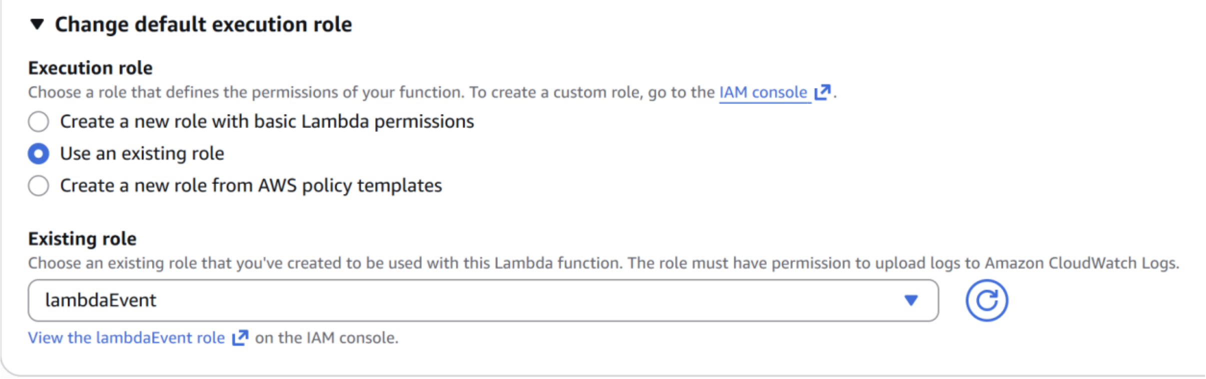
Python 3.14

Durable execution - *new* | Info
Enable durable execution to simplify building resilient multi-step applications that checkpoint progress and resume after interr
pricing.

□ Enable

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

○ arm64
● x86_64



▼ 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.

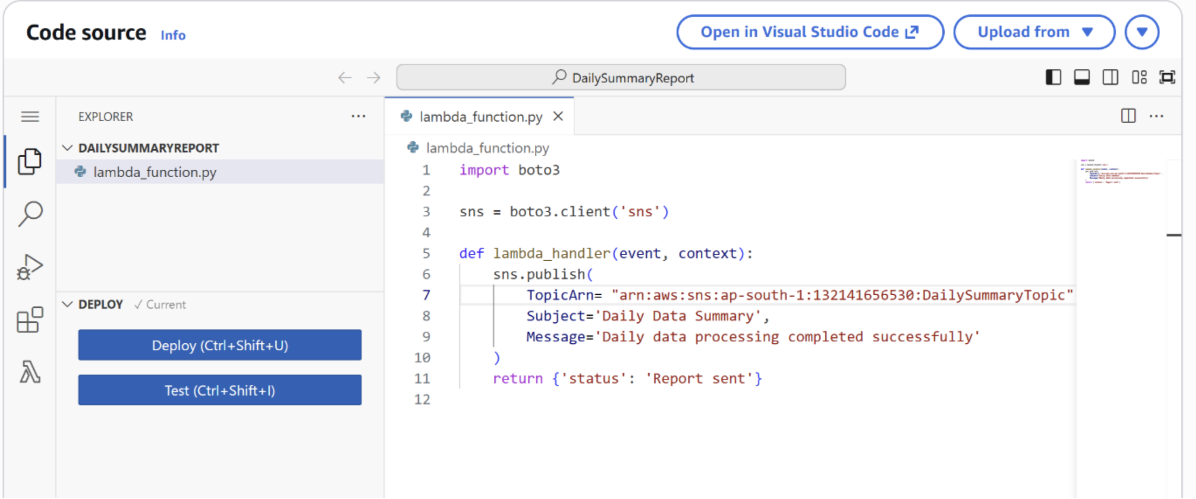
○ Create a new role with basic Lambda permissions
● Use an existing role
○ Create a new role from AWS policy templates

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.

lambdaEvent

View the lambdaEvent role on the IAM console.

21. Go inside the function and Scroll down , Inside the Code Tab write the Code we want to trigger in response to event:

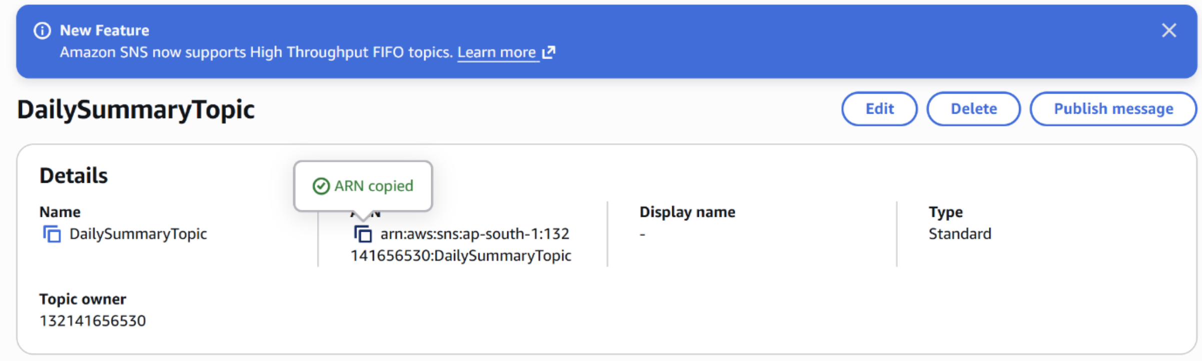


```python
import boto3

sns = boto3.client('sns')

def lambda_handler(event, context):
    sns.publish(
        TopicArn= "arn:aws:sns:ap-south-1:132141656530:DailySummaryTopic"
        Subject='Daily Data Summary',
        Message='Daily data processing completed successfully'
    )
    return {'status': 'Report sent'}
```

Sns Topic ARN can be copied inside the Topic:


New Feature
Amazon SNS now supports High Throughput FIFO topics. Learn more
Edit
Delete
Publish message
ARN copied

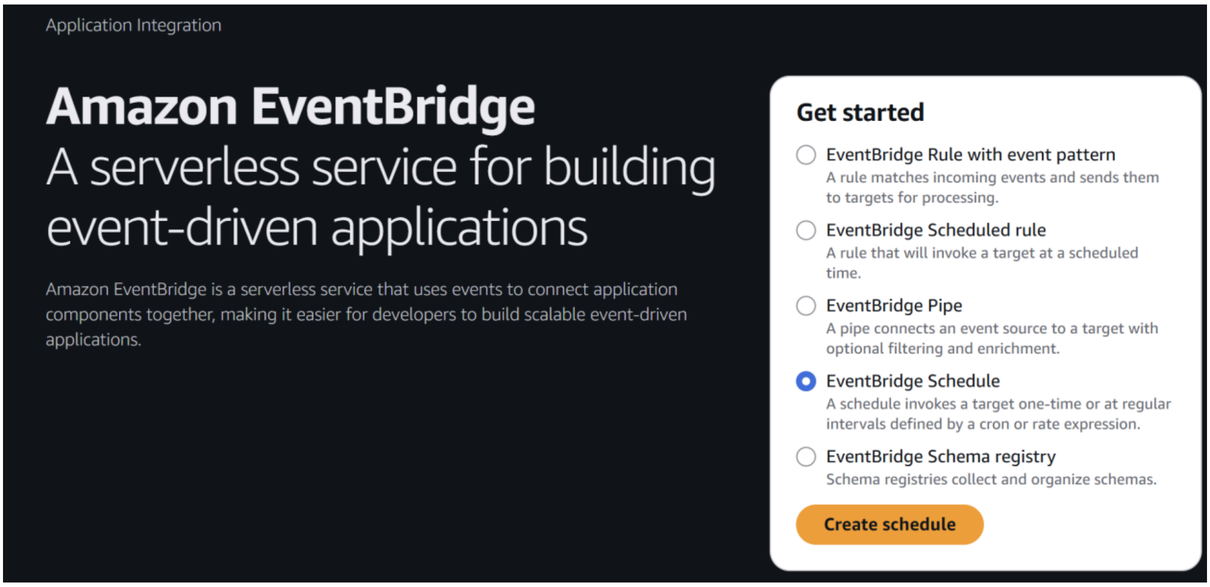
DailySummaryTopic

Details

Name	ARN	Display name	Type
DailySummaryTopic	arn:aws:sns:ap-south-1:132141656530:DailySummaryTopic	-	Standard

Topic owner
132141656530

22. Now Go to AWS EventBridge and select EventBridge Schedule to automate tasks based on time:



Application Integration

Amazon EventBridge
A serverless service for building event-driven applications

Amazon EventBridge is a serverless service that uses events to connect application components together, making it easier for developers to build scalable event-driven applications.

Get started

○ EventBridge Rule with event pattern
A rule matches incoming events and sends them to targets for processing.

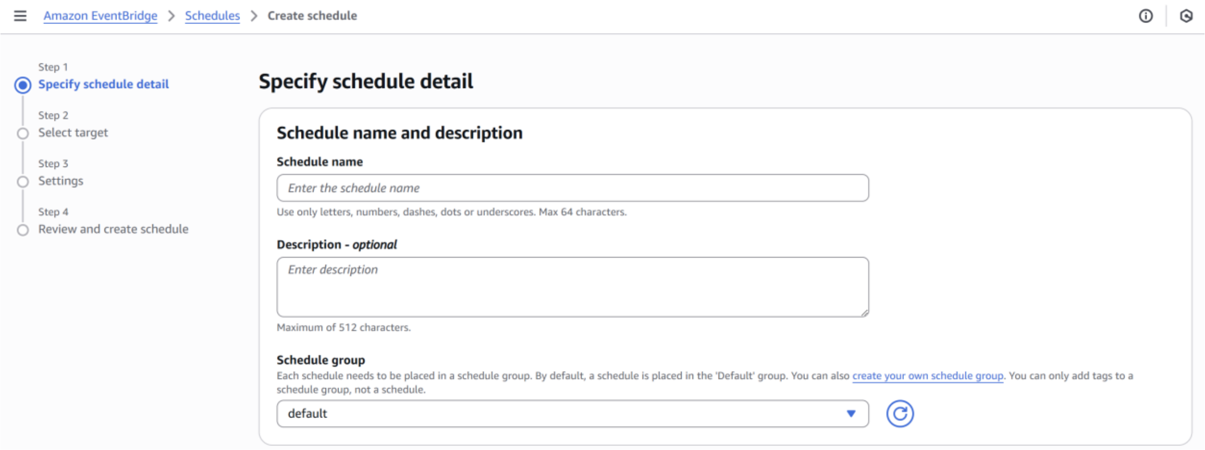
○ EventBridge Scheduled rule
A rule that will invoke a target at a scheduled time.

○ EventBridge Pipe
A pipe connects an event source to a target with optional filtering and enrichment.

● EventBridge Schedule
A schedule invokes a target one-time or at regular intervals defined by a cron or rate expression.

○ EventBridge Schema registry
Schema registries collect and organize schemas.

23. Give Name and keep schedule group as Default:



Amazon EventBridge > Schedules > Create schedule
Step 1
Specify schedule detail
Step 2
Select target
Step 3
Settings
Step 4
Review and create schedule

Specify schedule detail

Schedule name and description

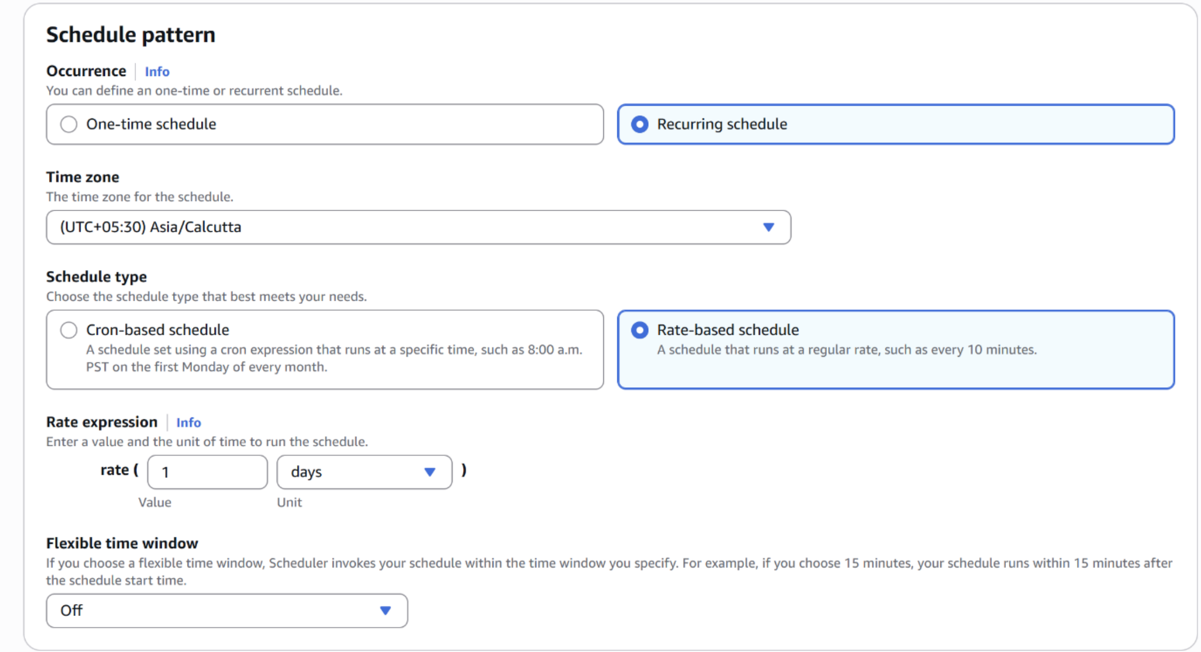
Schedule name
Enter the schedule name
Use only letters, numbers, dashes, dots or underscores. Max 64 characters.

Description - *optional*
Enter description
Maximum of 512 characters.

Schedule group
Each schedule needs to be placed in a schedule group. By default, a schedule is placed in the 'Default' group. You can also create your own schedule group. You can only add tags to a schedule group, not a schedule.

default

24. In Schedule Pattern Tab , Select the followings and Click Next:



Schedule pattern

Occurrence | Info

You can define an one-time or recurrent schedule.

○ One-time schedule

● Recurring schedule

Time zone

The time zone for the schedule.

(UTC+05:30) Asia/Calcutta

Schedule type

Choose the schedule type that best meets your needs.

○ Cron-based schedule
A schedule set using a cron expression that runs at a specific time, such as 8:00 a.m. PST on the first Monday of every month.

● Rate-based schedule
A schedule that runs at a regular rate, such as every 10 minutes.

Rate expression | Info

Enter a value and the unit of time to run the schedule.

rate (1 days)

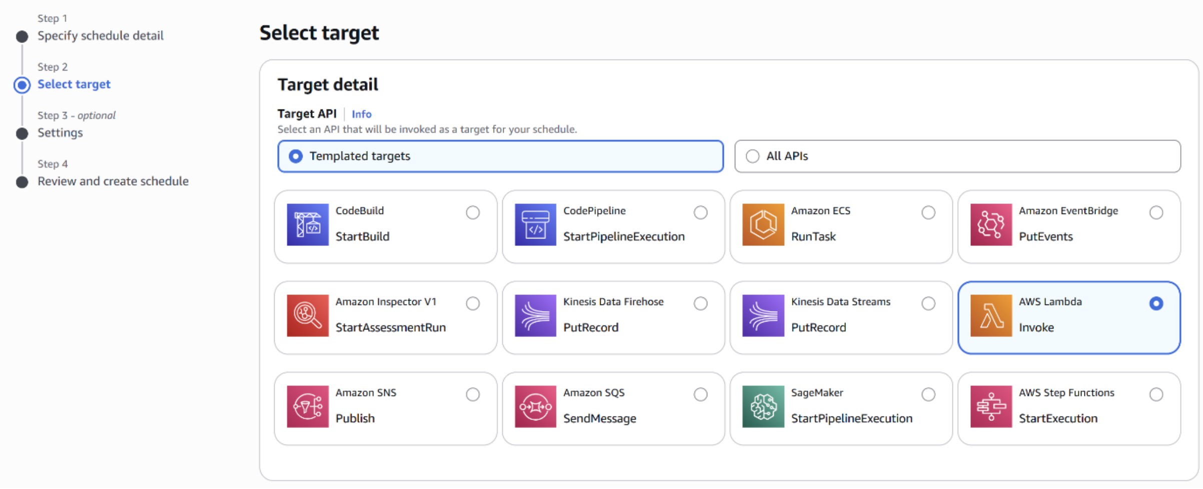
Value Unit

Flexible time window

If you choose a flexible time window, Scheduler invokes your schedule within the time window you specify. For example, if you choose 15 minutes, your schedule runs within 15 minutes after the schedule start time.

Off

25. Select AWS Lambda in Target:



Step 1
● Specify schedule detail

Step 2
● Select target

Step 3 - *optional*
● Settings

Step 4
● Review and create schedule

Select target

Target detail

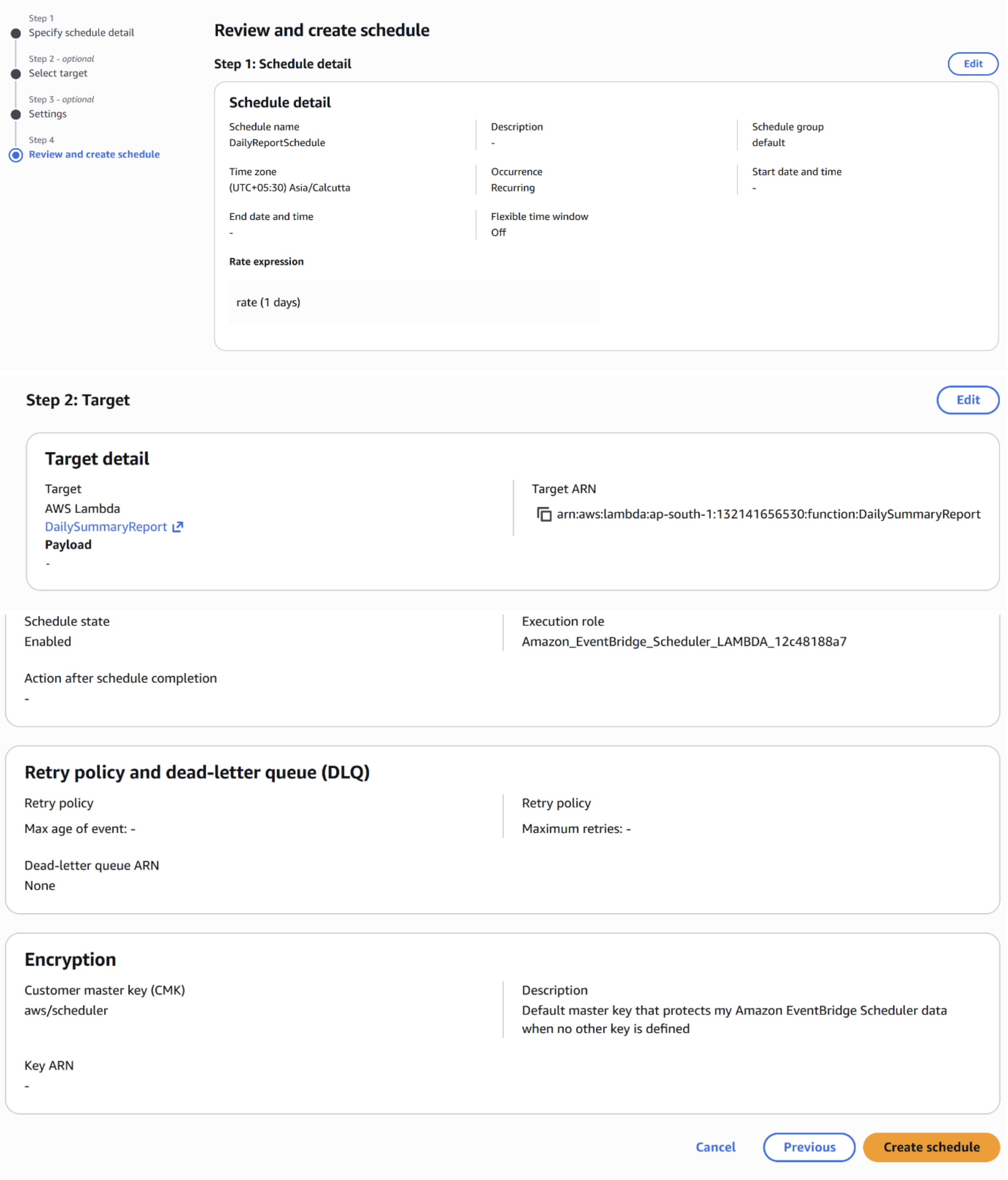
Target API | Info

Select an API that will be invoked as a target for your schedule.

● Templated targets ○ All APIs

CodeBuild StartBuild ○	CodePipeline StartPipelineExecution ○	Amazon ECS RunTask ○	Amazon EventBridge PutEvents ○
Amazon Inspector V1 StartAssessmentRun ○	Kinesis Data Firehose PutRecord ○	Kinesis Data Streams PutRecord ○	AWS Lambda Invoke ●
Amazon SNS Publish ○	Amazon SQS SendMessage ○	SageMaker StartPipelineExecution ○	AWS Step Functions StartExecution ○

26. Review and Create:



Review and create schedule

Step 1: Schedule detail

Schedule detail

Schedule name DailyReportSchedule	Description -	Schedule group default
Time zone (UTC+05:30) Asia/Calcutta	Occurrence Recurring	Start date and time -
End date and time -	Flexible time window Off	

Rate expression

rate (1 days)

Step 2: Target

Target detail

Target
AWS Lambda
DailySummaryReport
Payload
-

Target ARN
arn:aws:lambda:ap-south-1:132141656530:function:DailySummaryReport

Schedule state
Enabled

Execution role
Amazon_EventBridge_Scheduler_LAMBDA_12c48188a7

Action after schedule completion
-

Retry policy and dead-letter queue (DLQ)

Retry policy
Max age of event: -

Retry policy
Maximum retries: -

Dead-letter queue ARN
None

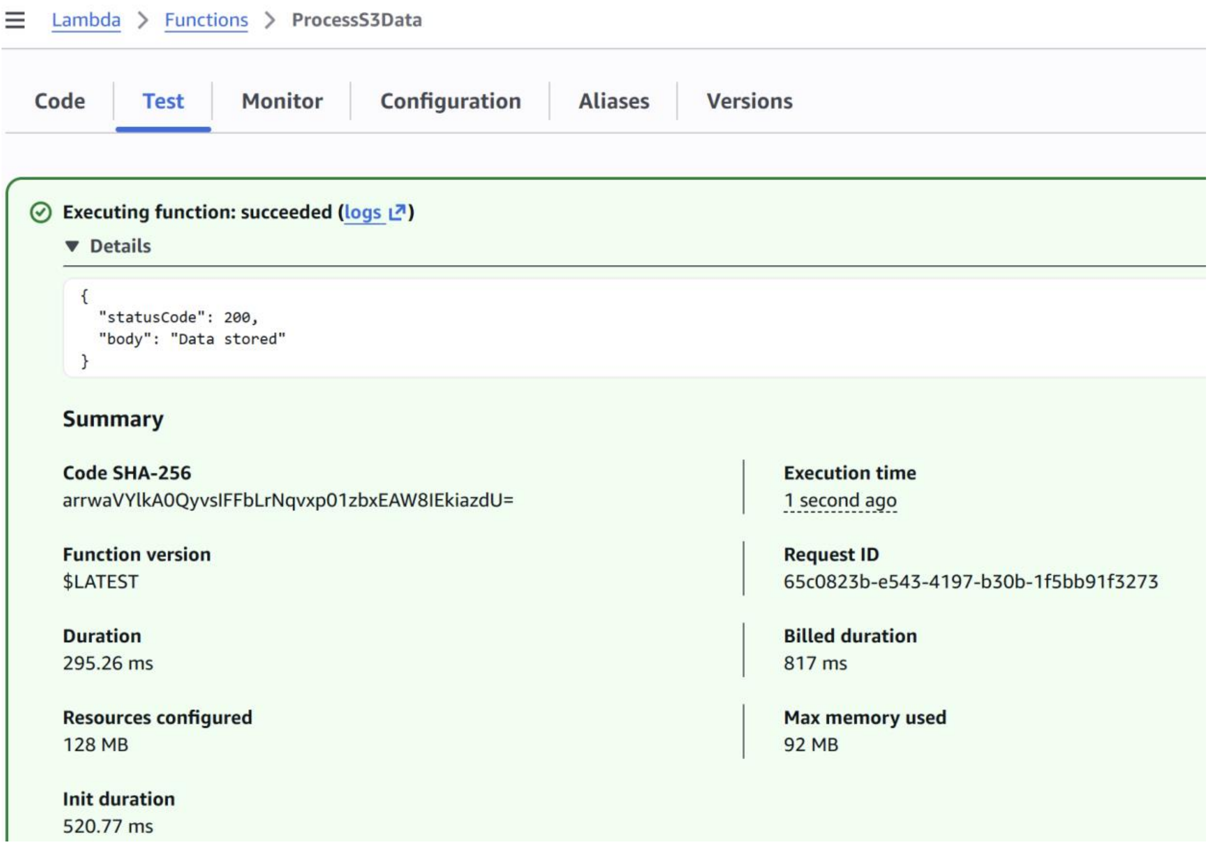
Encryption

Customer master key (CMK)
aws/scheduler

Description
Default master key that protects my Amazon EventBridge Scheduler data when no other key is defined

Key ARN
-

27. Testing Lambda Functions:



Executing function: succeeded (logs)

▼ Details

```
{
  "statusCode": 200,
  "body": "Data stored"
}
```

Summary

Code SHA-256
arrwaVYlkA0QyvsIFFbLrNqvxp01zbxEAW8IEkiazdU=

Execution time
1 second ago

Function version
$LATEST

Request ID
65c0823b-e543-4197-b30b-1f5bb91f3273

Duration
295.26 ms

Billed duration
817 ms

Resources configured
128 MB

Max memory used
92 MB

Init duration
520.77 ms

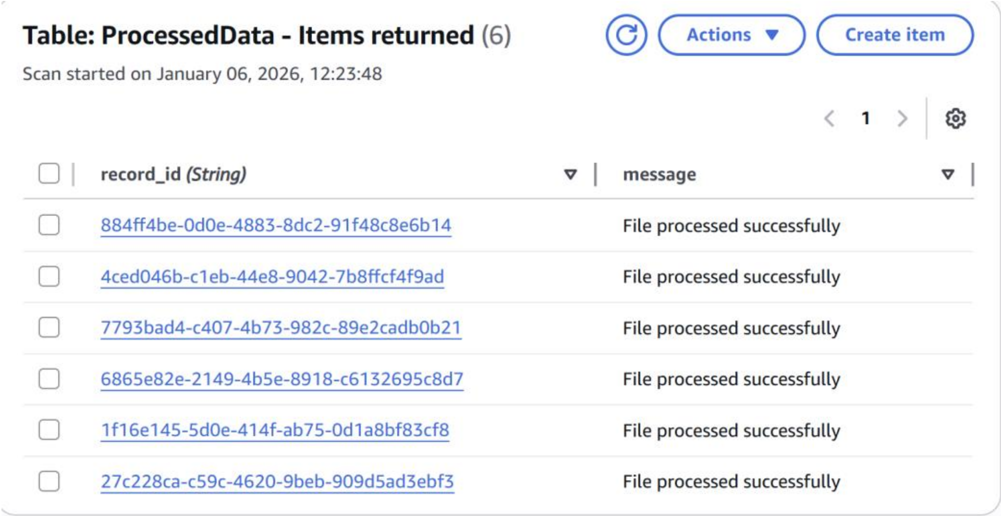
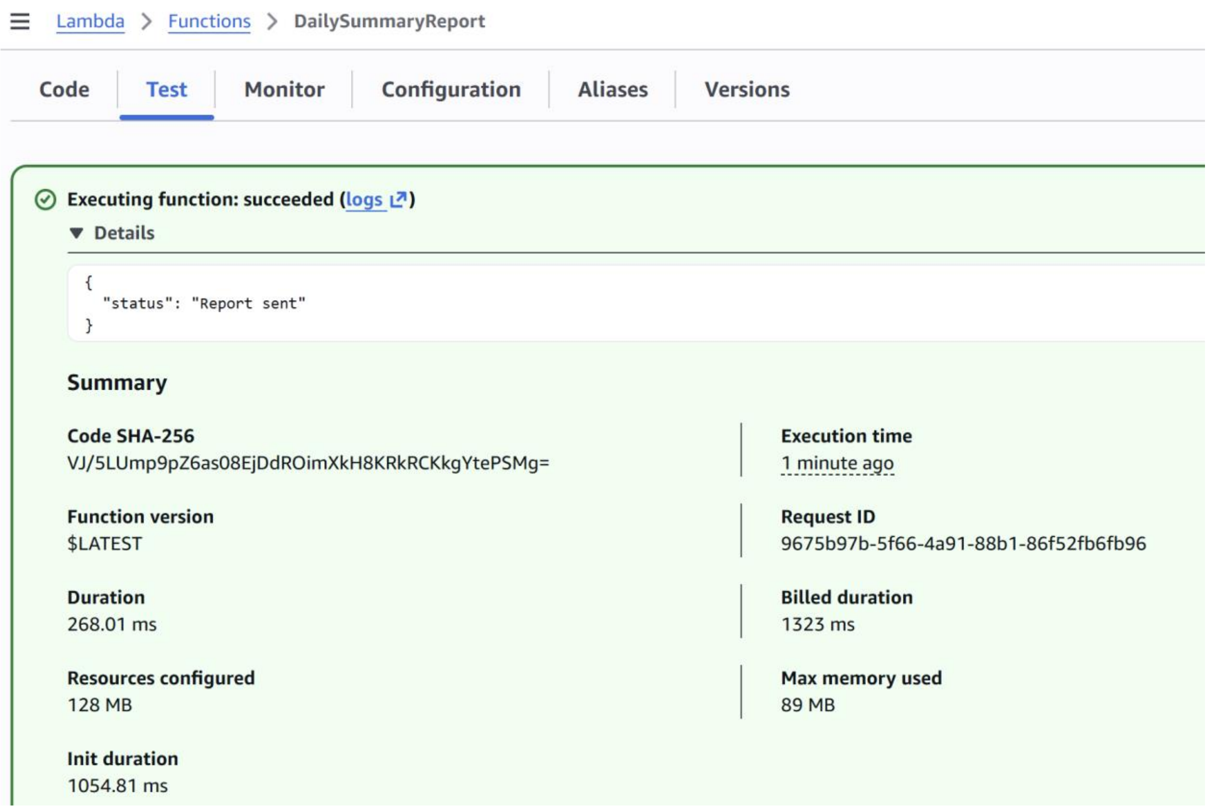


Table: ProcessedData - Items returned (6)

Scan started on January 06, 2026, 12:23:48

□	record_id (String)	message
□	884ff4be-0d0e-4883-8dc2-91f48c8e6b14	File processed successfully
□	4ced046b-c1eb-44e8-9042-7b8ffcf4f9ad	File processed successfully
□	7793bad4-c407-4b73-982c-89e2cadb0b21	File processed successfully
□	6865e82e-2149-4b5e-8918-c6132695c8d7	File processed successfully
□	1f16e145-5d0e-414f-ab75-0d1a8bf83cf8	File processed successfully
□	27c228ca-c59c-4620-9beb-909d5ad3ebf3	File processed successfully



≡ Lambda > Functions > DailySummaryReport

Code | Test | Monitor | Configuration | Aliases | Versions

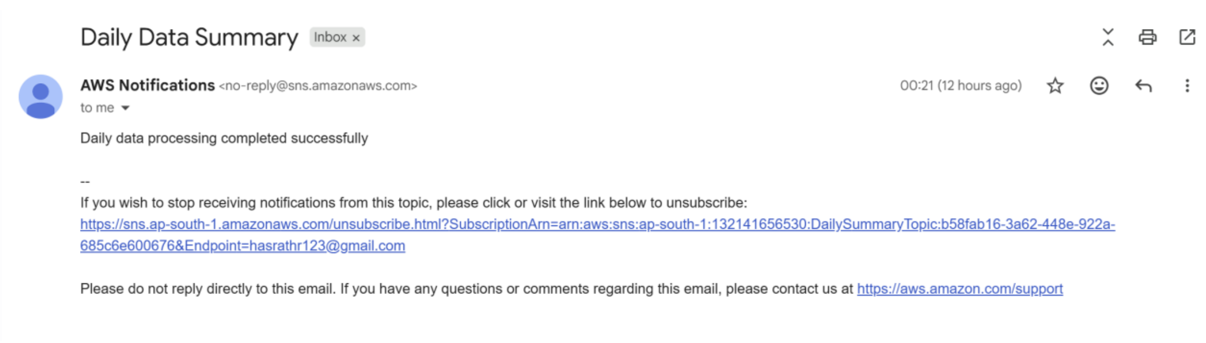
Executing function: succeeded (logs ↗)

▼ Details

```
{
  "status": "Report sent"
}
```

Summary

Code SHA-256 VJ/5LUmp9pZ6as08EjDdROimXkH8KRkRCKkgYtePSMg=	**Execution time** 1 minute ago
Function version $LATEST	**Request ID** 9675b97b-5f66-4a91-88b1-86f52fb6fb96
Duration 268.01 ms	**Billed duration** 1323 ms
Resources configured 128 MB	**Max memory used** 89 MB
Init duration 1054.81 ms	



Daily Data Summary

AWS Notifications <no-reply@sns.amazonaws.com> 00:21 (12 hours ago)

to me

Daily data processing completed successfully

--

If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.ap-south-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:ap-south-1:132141656530:DailySummaryTopic:b58fab16-3a62-448e-922a-685c6e600676&Endpoint=hasrathr123@gmail.com

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at https://aws.amazon.com/support