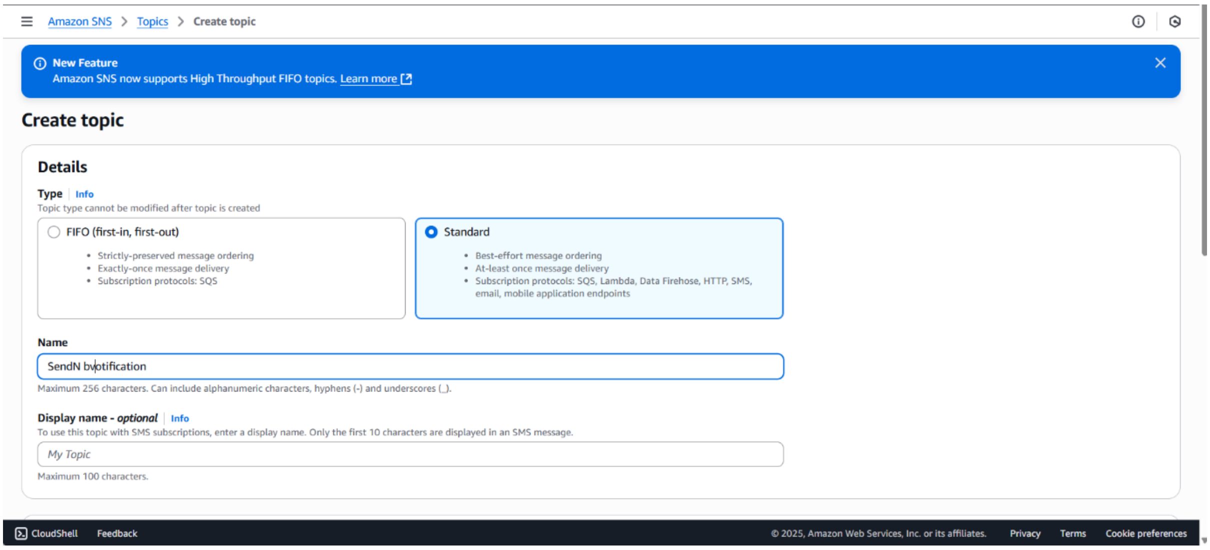
Task 2

The Project



Amazon SNS > Topics > Create topic

New Feature
Amazon SNS now supports High Throughput FIFO topics. Learn more

Create topic

Details

Type | Info
Topic type cannot be modified after topic is created

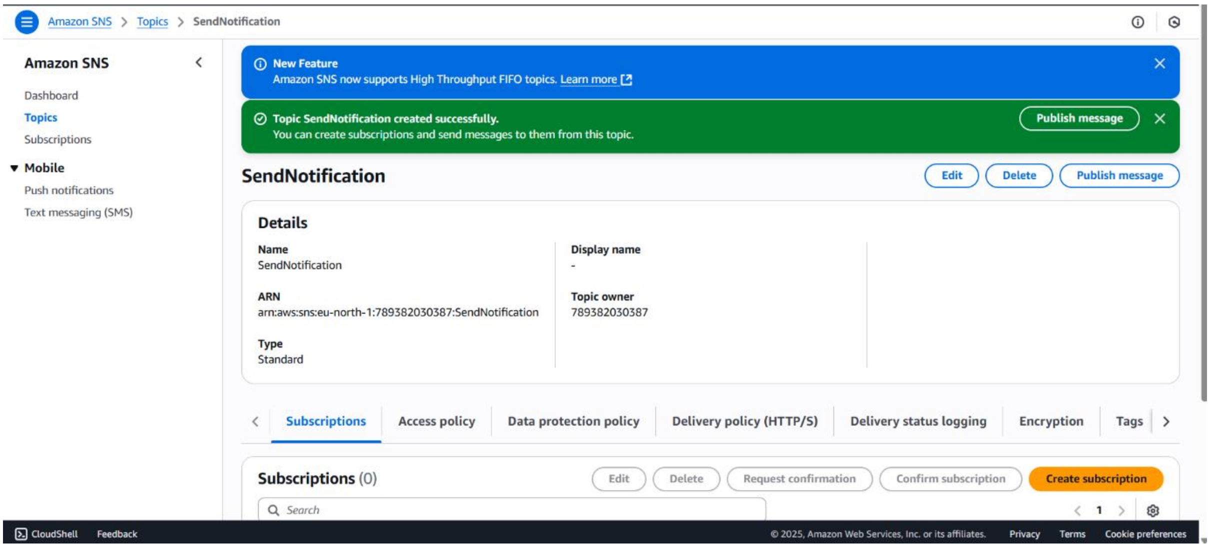
○ FIFO (first-in, first-out)
- Strictly-preserved message ordering
- Exactly-once message delivery
- Subscription protocols: SQS

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

Name
SendN bvotification
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.

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



Amazon SNS > Topics > SendNotification

New Feature
Amazon SNS now supports High Throughput FIFO topics. Learn more

Topic SendNotification created successfully.
You can create subscriptions and send messages to them from this topic.

SendNotification

Details

Name
SendNotification

ARN
arn:aws:sns:eu-north-1:789382030387:SendNotification

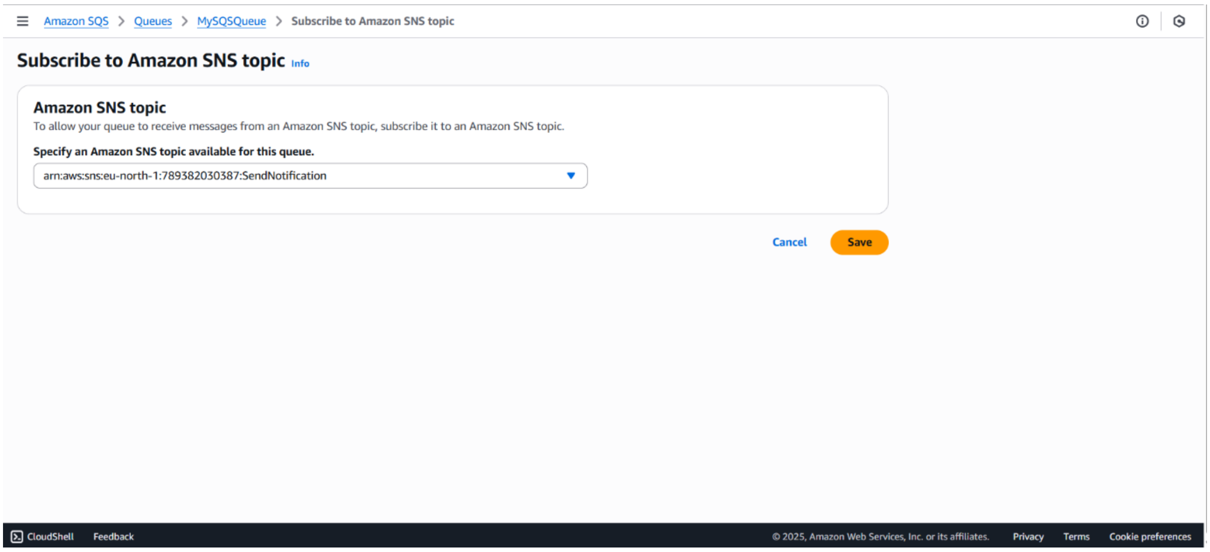
Type
Standard

Display name
-

Topic owner
789382030387

Subscriptions (0)

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



Amazon SQS > Queues > MySQSQueue > Subscribe to Amazon SNS topic

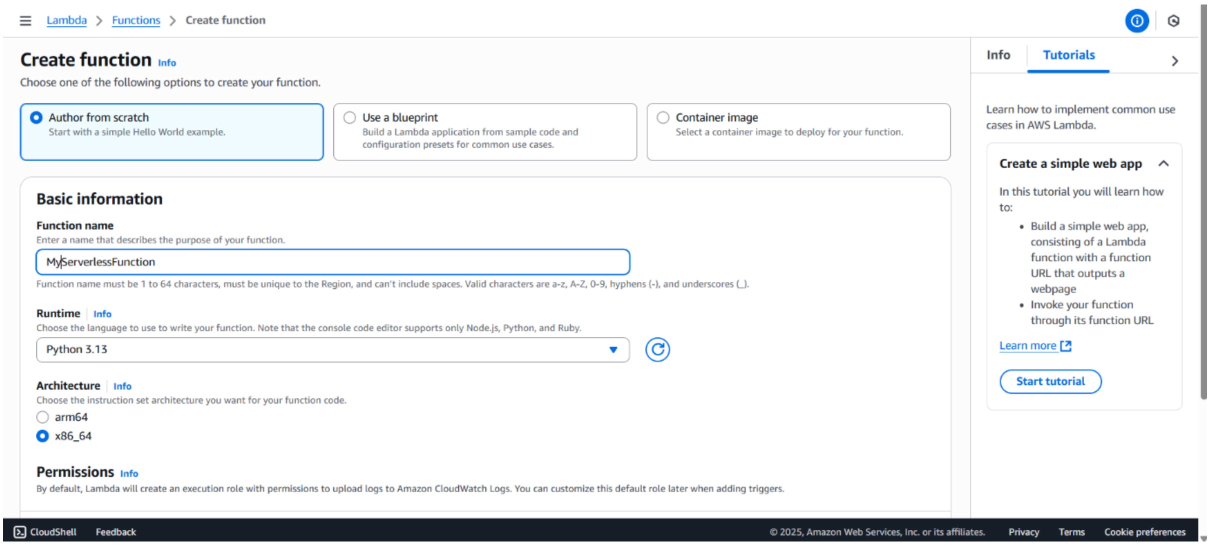
Subscribe to Amazon SNS topic Info

Amazon SNS topic

To allow your queue to receive messages from an Amazon SNS topic, subscribe it to an Amazon SNS topic.

Specify an Amazon SNS topic available for this queue.
arn:aws:sns:eu-north-1:789382030387:SendNotification ▼

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



Lambda > Functions > Create function
Info
Tutorials
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

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.
MyServerlessFunction
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 3.13

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

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.

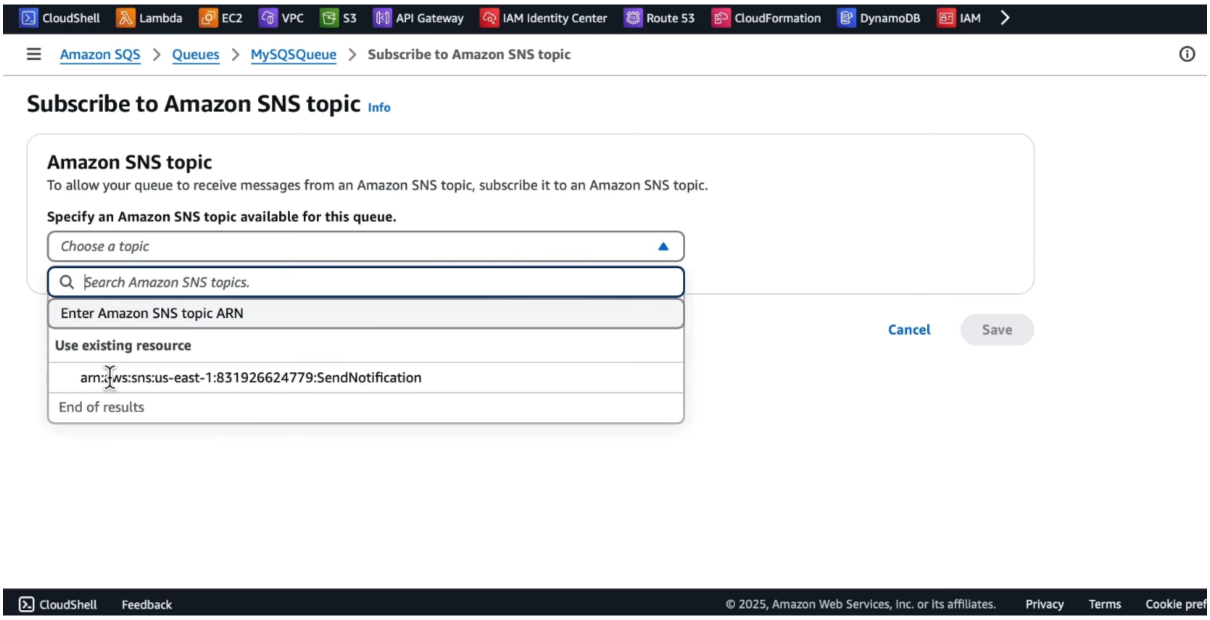
Learn how to implement common use cases in AWS Lambda.

Create a simple web app

In this tutorial you will learn how to:
- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

Learn more

Start tutorial



Subscribe to Amazon SNS topic Info

Amazon SNS topic

To allow your queue to receive messages from an Amazon SNS topic, subscribe it to an Amazon SNS topic.

Specify an Amazon SNS topic available for this queue.

Choose a topic

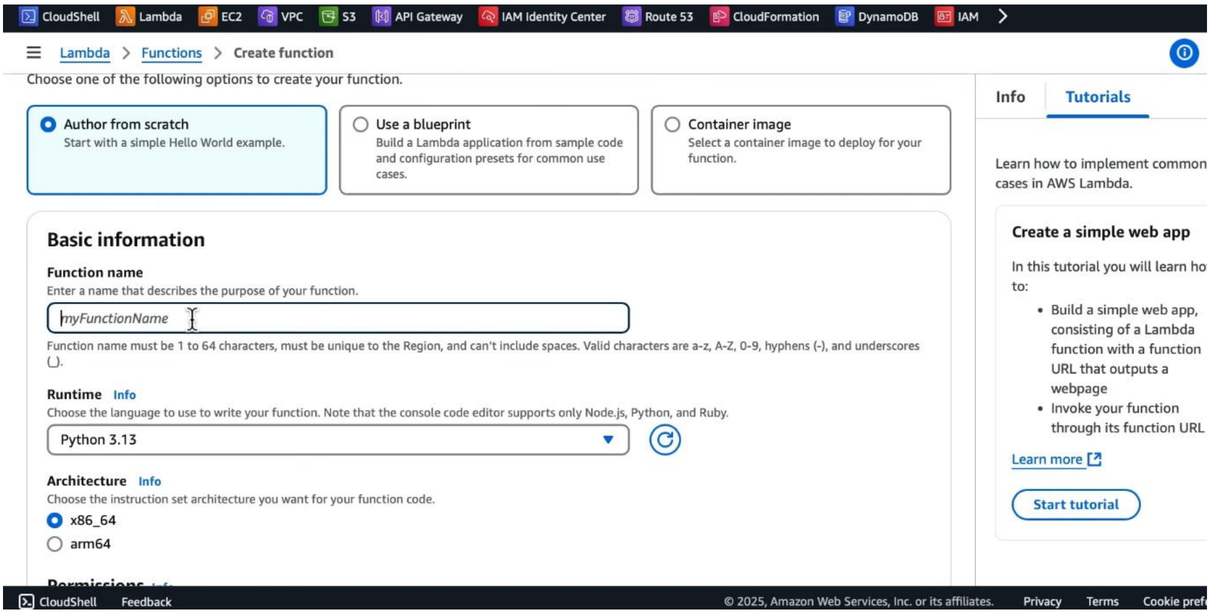
Search Amazon SNS topics.

Enter Amazon SNS topic ARN

Use existing resource

arn:aws:sns:us-east-1:831926624779:SendNotification

End of results



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.
myFunctionName
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 3.13

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

Permissions Info

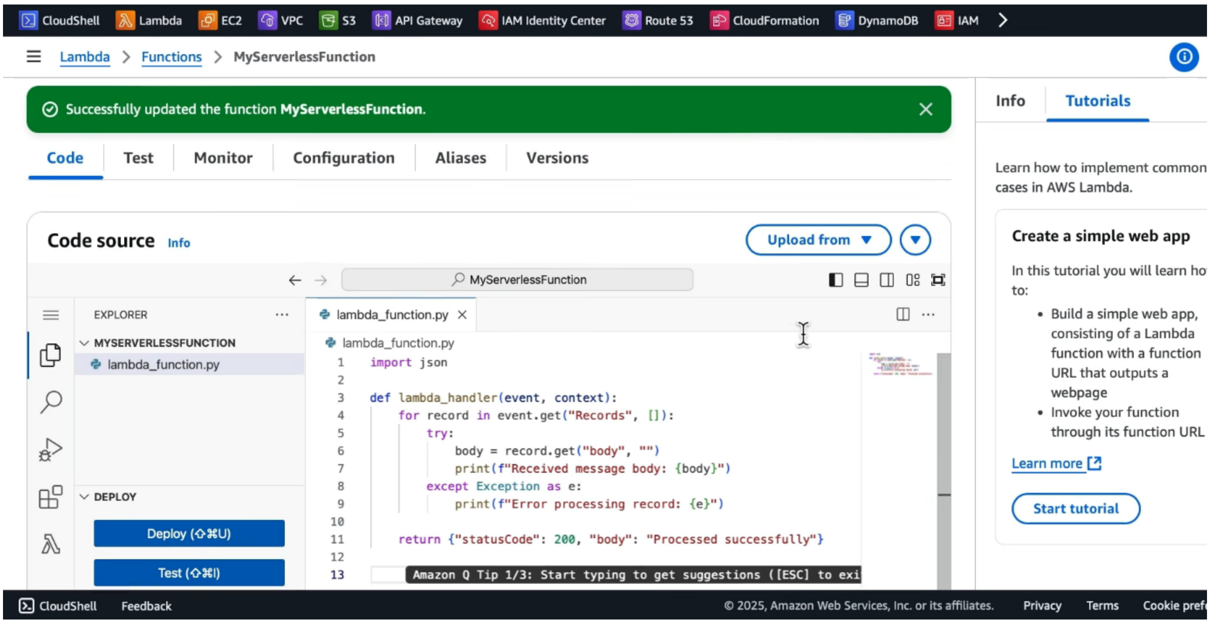
Learn how to implement common cases in AWS Lambda.

Create a simple web app

In this tutorial you will learn ho to:
- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

Learn more

Start tutorial



Successfully updated the function **MyServerlessFunction**.

lambda_function.py

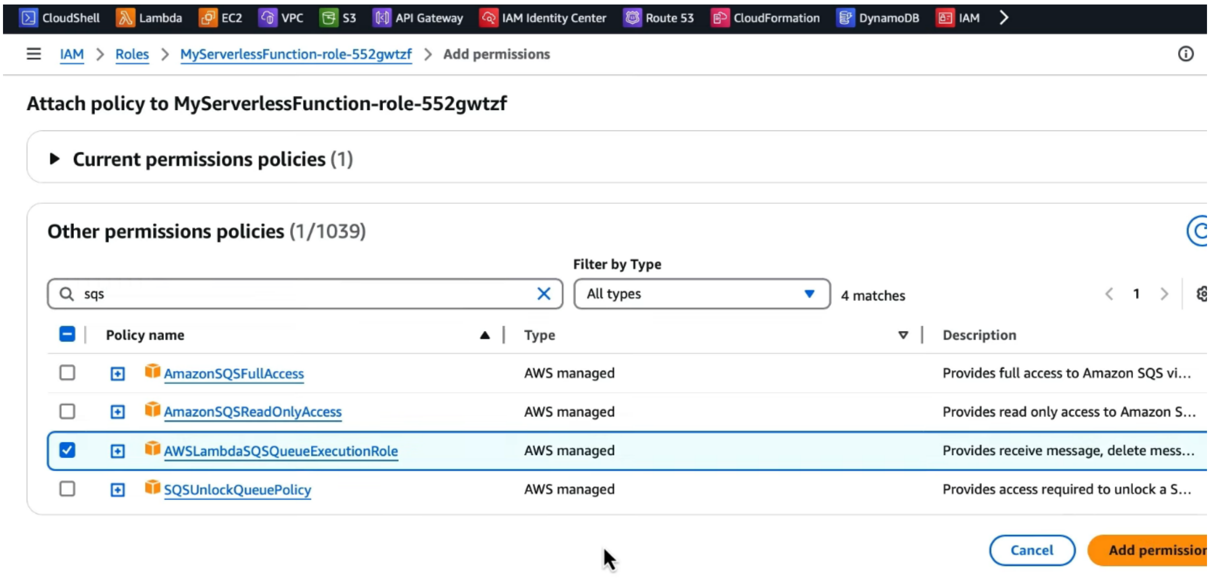
```
import json

def lambda_handler(event, context):
    for record in event.get("Records", []):
        try:
            body = record.get("body", "")
            print(f"Received message body: {body}")
        except Exception as e:
            print(f"Error processing record: {e}")

    return {"statusCode": 200, "body": "Processed successfully"}

 Amazon Q Tip 1/3: Start typing to get suggestions ([ESC] to exi
```

Learn how to implement common cases in AWS Lambda.

Create a simple web app

In this tutorial you will learn ho to:

- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL



Attach policy to MyServerlessFunction-role-552gwtzf

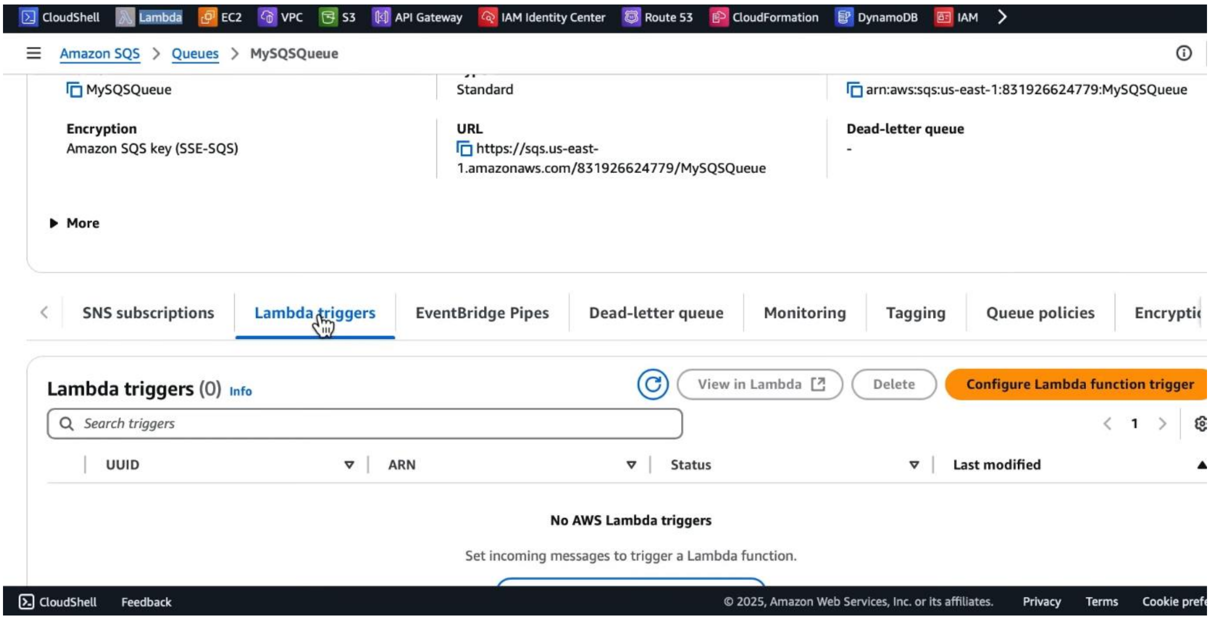
▶ Current permissions policies (1)

Other permissions policies (1/1039)

Filter by Type

sqs | All types | 4 matches | 1

	Policy name	Type	Description
☐	AmazonSQSFullAccess	AWS managed	Provides full access to Amazon SQS vi...
☐	AmazonSQSReadOnlyAccess	AWS managed	Provides read only access to Amazon S...
☑	AWSLambdaSQSQueueExecutionRole	AWS managed	Provides receive message, delete mess...
☐	SQSUnlockQueuePolicy	AWS managed	Provides access required to unlock a S...



MySQSQueue | Standard | arn:aws:sqs:us-east-1:831926624779:MySQSQueue

Encryption
Amazon SQS key (SSE-SQS)

URL
https://sqs.us-east-1.amazonaws.com/831926624779/MySQSQueue

Dead-letter queue
-

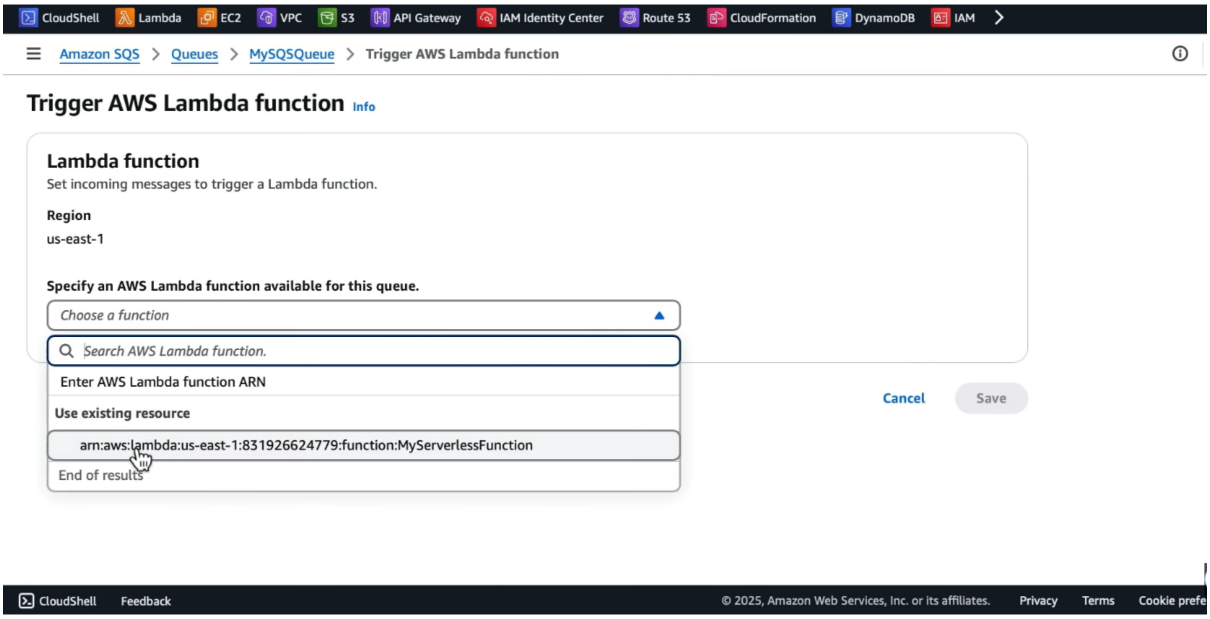
▶ More

Lambda triggers (0) Info

UUID	ARN	Status	Last modified

No AWS Lambda triggers

Set incoming messages to trigger a Lambda function.



Trigger AWS Lambda function

Lambda function

Set incoming messages to trigger a Lambda function.

Region

us-east-1

Specify an AWS Lambda function available for this queue.

Choose a function

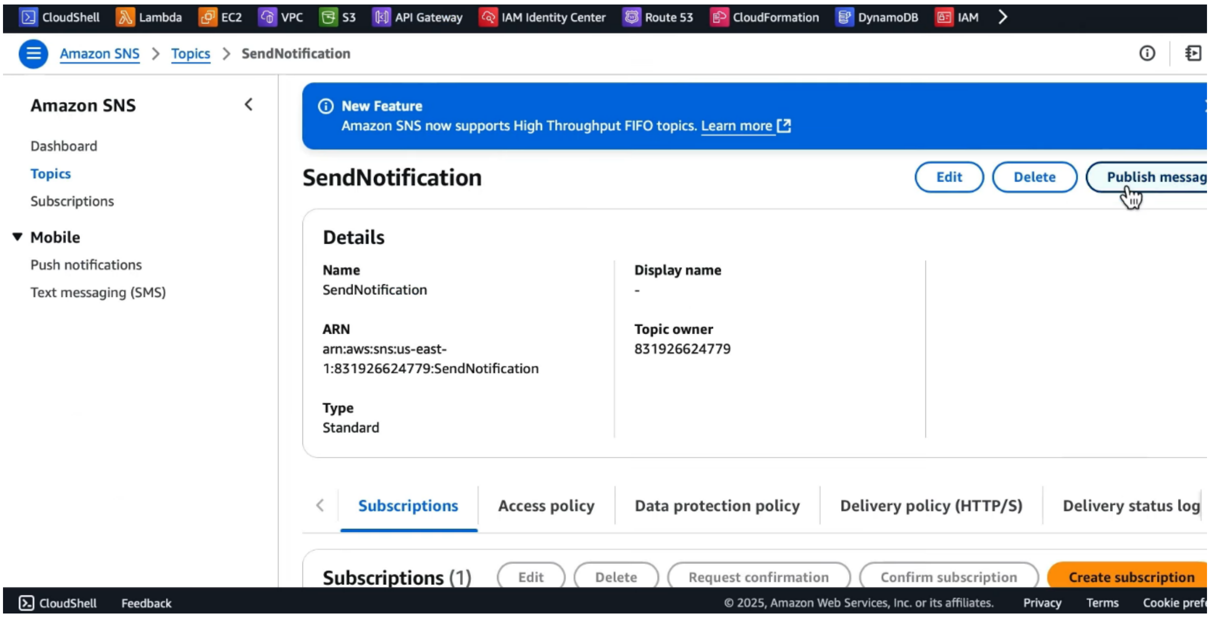
Search AWS Lambda function.

Enter AWS Lambda function ARN

Use existing resource

arn:aws:lambda:us-east-1:831926624779:function:MyServerlessFunction

End of results



New Feature

Amazon SNS now supports High Throughput FIFO topics. Learn more

SendNotification

Details

Name

SendNotification

Display name

-

ARN

arn:aws:sns:us-east-1:831926624779:SendNotification

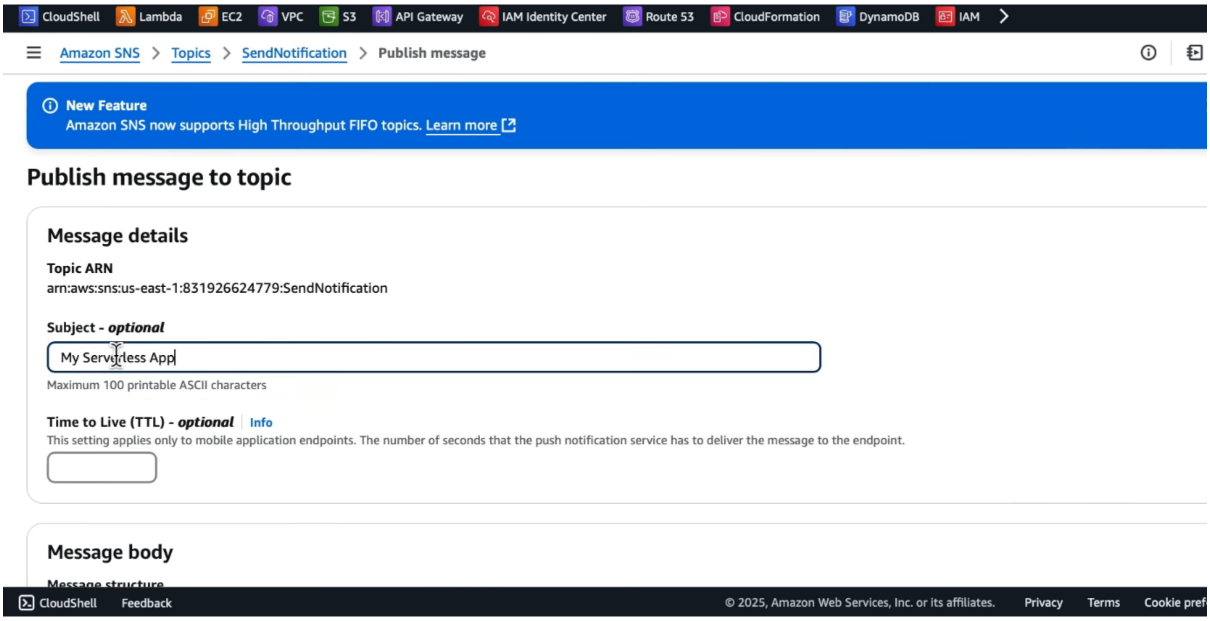
Topic owner

831926624779

Type

Standard

Subscriptions (1)



New Feature
Amazon SNS now supports High Throughput FIFO topics. Learn more

Publish message to topic

Message details

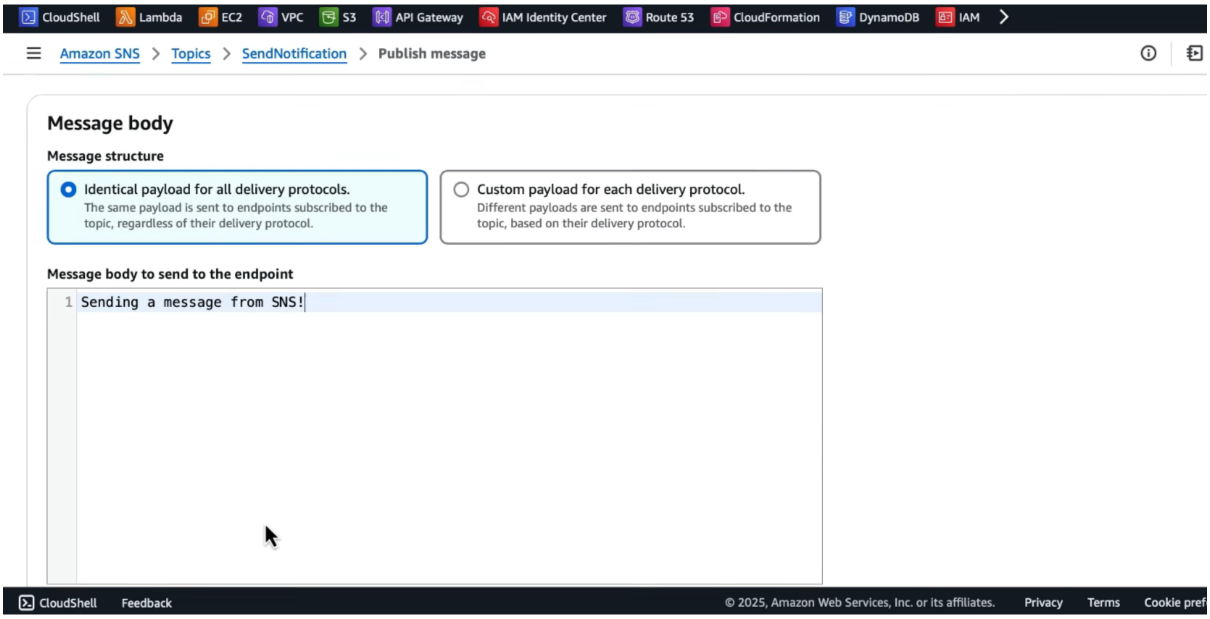
Topic ARN
arn:aws:sns:us-east-1:831926624779:SendNotification

Subject - *optional*
My Serverless App
Maximum 100 printable ASCII characters

Time to Live (TTL) - *optional* Info
This setting applies only to mobile application endpoints. The number of seconds that the push notification service has to deliver the message to the endpoint.

Message body

Message structure



Message body

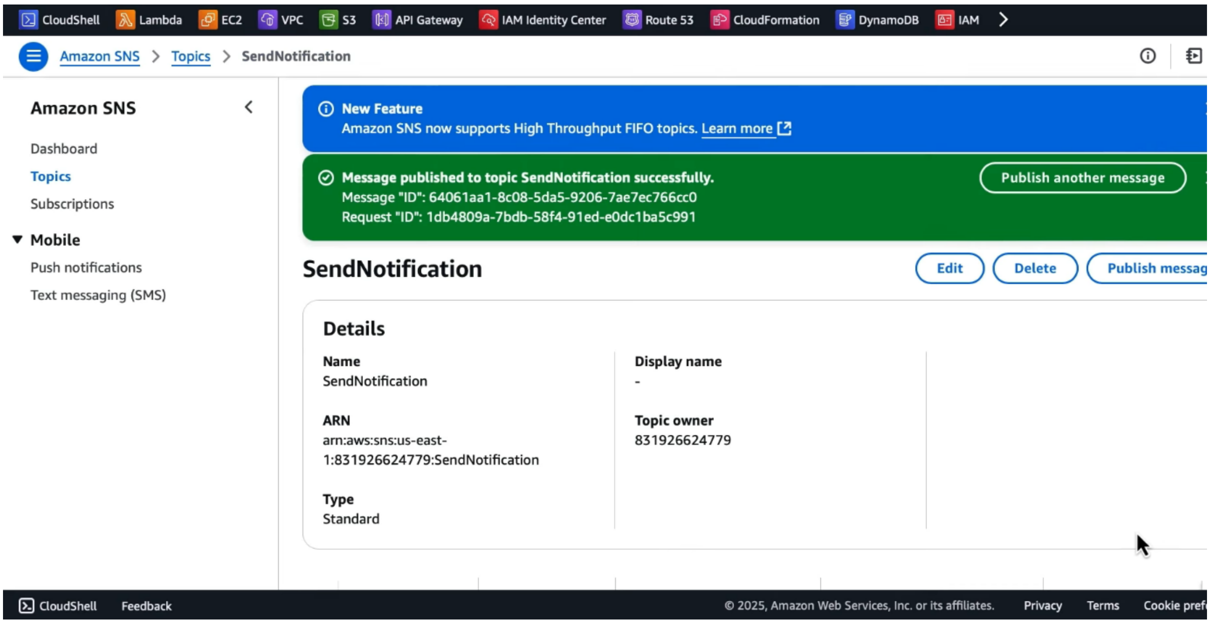
Message structure

● Identical payload for all delivery protocols.
The same payload is sent to endpoints subscribed to the topic, regardless of their delivery protocol.

○ Custom payload for each delivery protocol.
Different payloads are sent to endpoints subscribed to the topic, based on their delivery protocol.

Message body to send to the endpoint

```
Sending a message from SNS!
```



New Feature
Amazon SNS now supports High Throughput FIFO topics. Learn more

Message published to topic SendNotification successfully.
Message "ID": 64061aa1-8c08-5da5-9206-7ae7ec766cc0
Request "ID": 1db4809a-7bdb-58f4-91ed-e0dc1ba5c991

SendNotification

Details

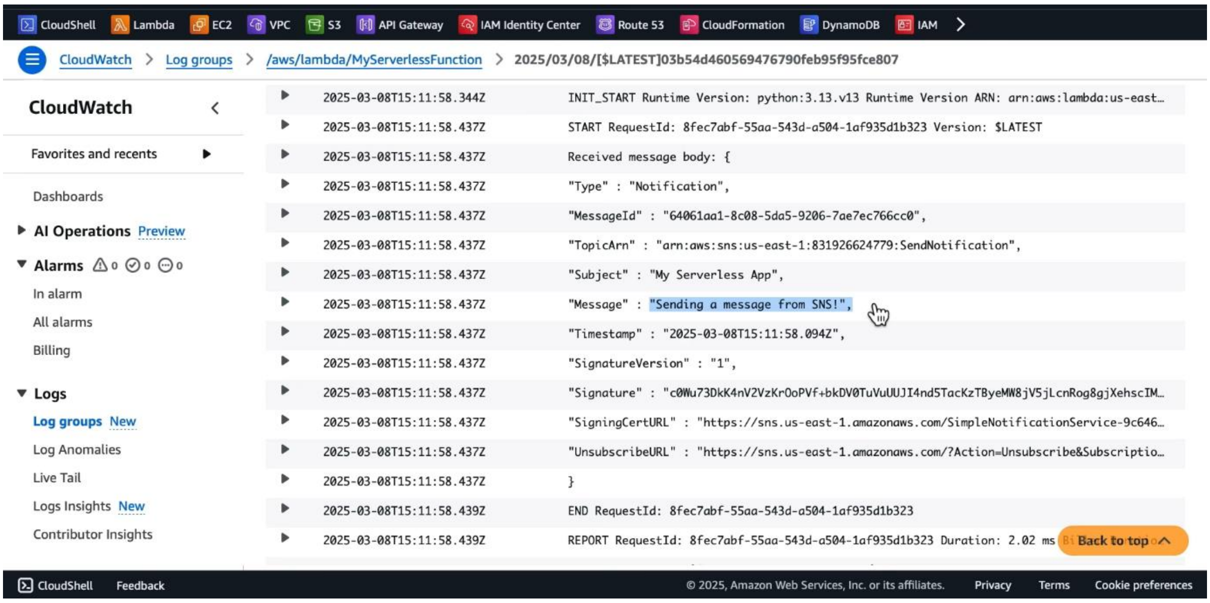
Name
SendNotification

Display name
-

ARN
arn:aws:sns:us-east-1:831926624779:SendNotification

Topic owner
831926624779

Type
Standard



```
2025-03-08T15:11:58.344Z    INIT_START Runtime Version: python:3.13.v13 Runtime Version ARN: arn:aws:lambda:us-east…
2025-03-08T15:11:58.437Z    START RequestId: 8fec7abf-55aa-543d-a504-1af935d1b323 Version: $LATEST
2025-03-08T15:11:58.437Z    Received message body: {
2025-03-08T15:11:58.437Z    "Type" : "Notification",
2025-03-08T15:11:58.437Z    "MessageId" : "64061aa1-8c08-5da5-9206-7ae7ec766cc0",
2025-03-08T15:11:58.437Z    "TopicArn" : "arn:aws:sns:us-east-1:831926624779:SendNotification",
2025-03-08T15:11:58.437Z    "Subject" : "My Serverless App",
2025-03-08T15:11:58.437Z    "Message" : "Sending a message from SNS!",
2025-03-08T15:11:58.437Z    "Timestamp" : "2025-03-08T15:11:58.094Z",
2025-03-08T15:11:58.437Z    "SignatureVersion" : "1",
2025-03-08T15:11:58.437Z    "Signature" : "c0Wu73DkK4nV2VzKrOoPVf+bkDV0TuVuUUJI4nd5TacKzTByeMW8jV5jLcnRog8gjXehscIM…
2025-03-08T15:11:58.437Z    "SigningCertURL" : "https://sns.us-east-1.amazonaws.com/SimpleNotificationService-9c646…
2025-03-08T15:11:58.437Z    "UnsubscribeURL" : "https://sns.us-east-1.amazonaws.com/?Action=Unsubscribe&Subscriptio…
2025-03-08T15:11:58.437Z    }
2025-03-08T15:11:58.439Z    END RequestId: 8fec7abf-55aa-543d-a504-1af935d1b323
2025-03-08T15:11:58.439Z    REPORT RequestId: 8fec7abf-55aa-543d-a504-1af935d1b323 Duration: 2.02 ms
```

Uses and Application

1. Automatic Backend Processing

   Serverless functions are ideal for automating tasks such as data processing, file conversions, image resizing, or sending notifications—without the need to manage backend servers.

2. Cost-Efficient Infrastructure
   Since you only pay for the compute time your function consumes, serverless architectures help significantly reduce cloud infrastructure costs, especially for infrequent or unpredictable workloads.

3. Scalability Without Management
   AWS automatically scales Lambda functions in response to the incoming workload, handling thousands of requests per second without manual intervention.