

# Amazon Message Platforms

## Overview

- Configure simple notification service(SNS)
- Create a lambda function and subscribe to the sns topic
- Configure simple queue service & Simple email service

## Configure simple notification service(SNS)

1.Navigate to SNS dashboard navigate to topic and click on create new topic

2.You can choose the type as standard

3.Provide the topic name, display name

A:

**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
- High throughput, up to 300 publishes/second
- Subscription protocols: SQS

☒ Standard

- Best-effort message ordering
- At-least once message delivery
- Highest throughput in publishes/second
- Subscription protocols: SQS, Lambda, HTTP, SMS, email, mobile application endpoints

Name

sns\_1

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.

sns-01

Maximum 100 characters.

4.Navigate to Access policy Define who can publish messages to the topic -> Only the topic owner. Define who can subscribe to this topic -> Everyone

A:

Amazon SNS provides in-transit encryption by default. Enabling server-side encryption adds at-rest encryption to your topic.

**Access policy - optional Info**  
This policy defines who can access your topic. By default, only the topic owner can publish or subscribe to the topic.

Choose method

☒ Basic  
Use simple criteria to define a basic access policy.

☐ Advanced  
Use a JSON object to define an advanced access policy.

Publishers  
Specify who can publish messages to the topic.

Only the topic owner  
Only the owner of the topic can publish to the topic

Subscribers  
Specify who can subscribe to this topic.

Everyone  
Any AWS account can subscribe to the topic

JSON preview

```
{
  "Version": "2008-10-17",
  "Id": "__default_policy_ID",
  "Statement": [
    {
      "Sid": "__default_statement_ID",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "SNS:Publish",

```

5.Navigate to IAM role, select create a new service role click Create new roles

6.Click on Allow and new service role will be created now

A:

**IAM roles**  
Amazon SNS requires permission to write logs to CloudWatch Logs. You can use separate roles for successful and failed message deliveries.

Service role [Info](#)

☐ Use existing service role  
Choose an existing service role from your account

☒ Create new service role  
Create a new service role in your account

[Create new roles](#)

IAM role for successful deliveries  
arn:aws:iam::628132916033:role/SNSSuccessFeedback

IAM role for failed deliveries  
arn:aws:iam::628132916033:role/SNSFailureFeedback

7.Click on create topic now and your sns topic should be created now

A:

**Topic sns\_1 created successfully.**  
You can create subscriptions and send messages to them from this topic. [Publish message](#)

[Amazon SNS](#) > [Topics](#) > sns\_1

**sns\_1** [Edit](#) [Delete](#) [Publish message](#)

**Details**

Name sns_1	Display name sns_new
ARN arn:aws:sns:ap-south-1:628132916033:sns_1	Topic owner 628132916033
Type Standard	

8.Navigate to subscriptions tab and click on create subscription(confirm through mail)

9.In the choose the topic from the dropdown, choose the protocol ex: email

10. Leave all others default, and then click on create subscription

A:

**Create subscription**

**Details**

Topic ARN

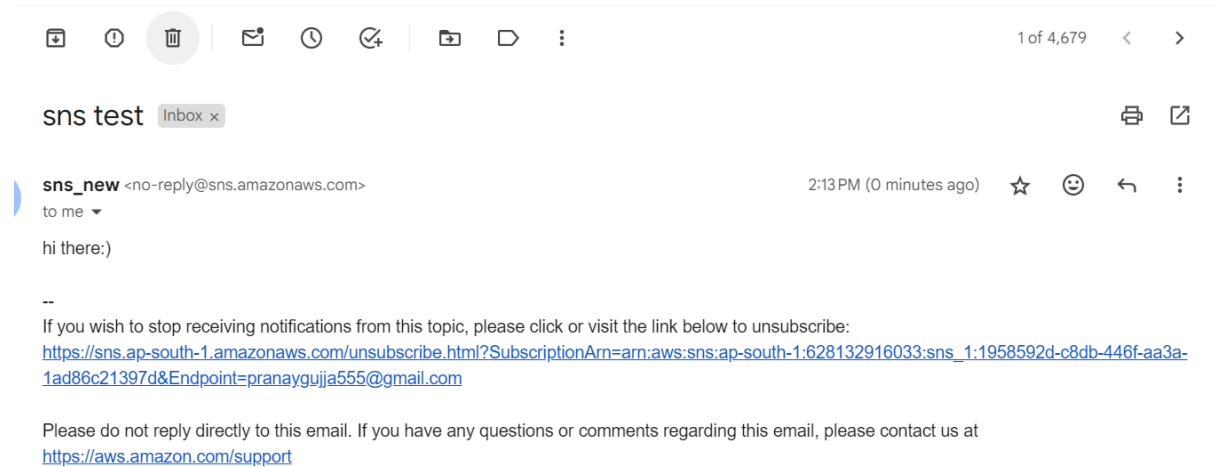
Protocol  
The type of endpoint to subscribe

Endpoint  
An email address that can receive notifications from Amazon SNS.

[Info](#) After your subscription is created, you must confirm it. [Info](#)

11. Navigate to topics page, select the topic and then click public message
12. Provide the subject, provide the message body, and click on publish message
13. Verify you received the subscription mail

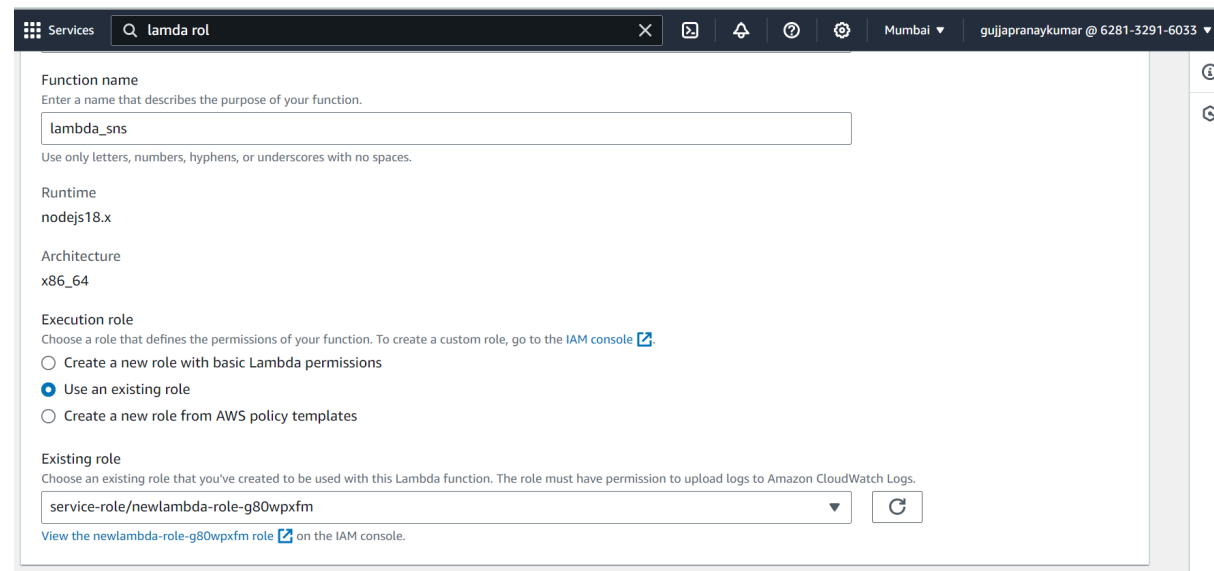
A:



## Create a lambda function and subscribe to the sns topic

1. Navigate to lambda services and navigate to the function -> create function
2. Select use a blueprint option, search for sns message using node.js
3. Provide the function name , select the existing lambda role you have created before

A:



4. Choose the sns topic that you have created
5. Click on create function it will get created and verify the sns subscription page

A:

The screenshot shows the AWS Lambda console. At the top, there's a search bar with 'lamda rol' and a 'Services' dropdown. Below the search bar, the breadcrumb navigation shows 'Lambda > Functions'. The main heading is 'Functions (2)' with a refresh button and an 'Actions' dropdown. A search bar is present with the placeholder 'Filter by tags and attributes or search by keyword'. Below this is a table with the following data:

<input type="checkbox"/>	Function name	Description	Package type	Runtime	Last modified
<input type="checkbox"/>	lamda	-	Zip	Python 3.11	3 months ago
<input type="checkbox"/>	lambda_sns	An Amazon SNS trigger that logs the message pushed to the SNS topic.	Zip	Node.js 18.x	40 seconds ago

The screenshot shows the AWS SNS console. At the top, there's a search bar with '[Alt+S]' and a 'Services' dropdown. Below the search bar, the breadcrumb navigation shows 'Amazon SNS > Subscriptions'. The main heading is 'Subscriptions (3)' with buttons for 'Edit', 'Delete', 'Request confirmation', 'Confirm subscription', and 'Create subscription'. A search bar is present with the placeholder 'Search'. Below this is a table with the following data:

<input type="radio"/>	ID	Endpoint	Status	Protocol	Topic
<input type="radio"/>	f8a59666-a457-...	pranaygujja555...	Confirmed	EMAIL	autoscaling
<input type="radio"/>	1958592d-c8db-...	pranaygujja555...	Confirmed	EMAIL	sns_1
<input type="radio"/>	63202332-8b04-...	arn:aws:lambda:a...	Confirmed	LAMBDA	sns_1

6.Publish a new message and verify the mail is triggered for lambda

A:

The screenshot shows an email received from SNS. The email header includes 'sns\_new <no-reply@sns.amazonaws.com>' and '3:38 PM (0 minutes ago)'. The body of the email contains the following text:

hi there  
a sample mail for lambda sns testing

--

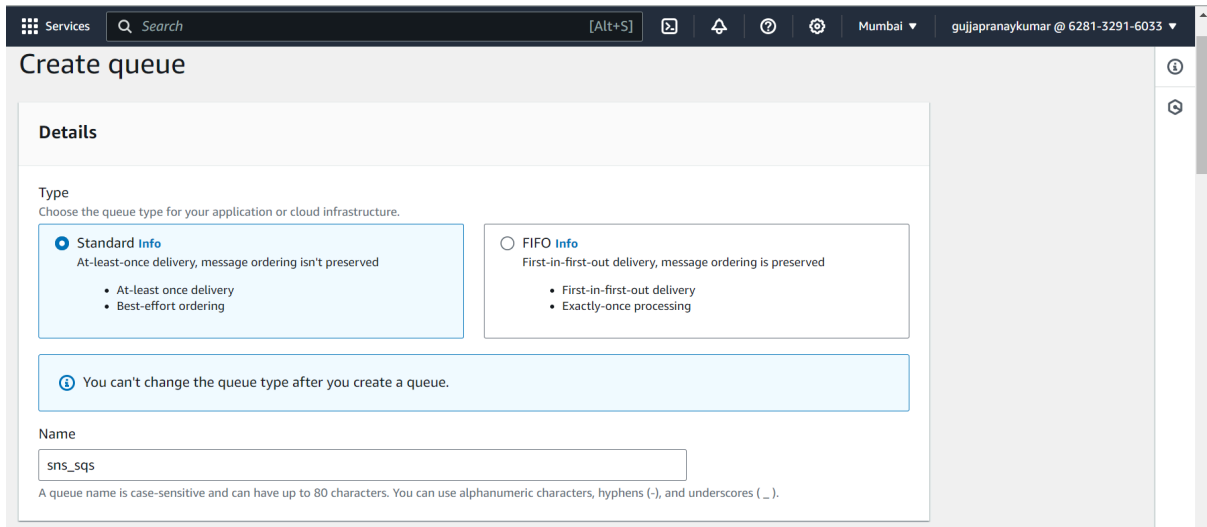
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:  
[https://sns.ap-south-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:ap-south-1:628132916033:sns\\_1:1958592d-c8db-446f-aa3a-1ad86c21397d&Endpoint=pranaygujja555@gmail.com](https://sns.ap-south-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:ap-south-1:628132916033:sns_1:1958592d-c8db-446f-aa3a-1ad86c21397d&Endpoint=pranaygujja555@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>

## Configure simple queue service subscribe to sns topic to receive the message

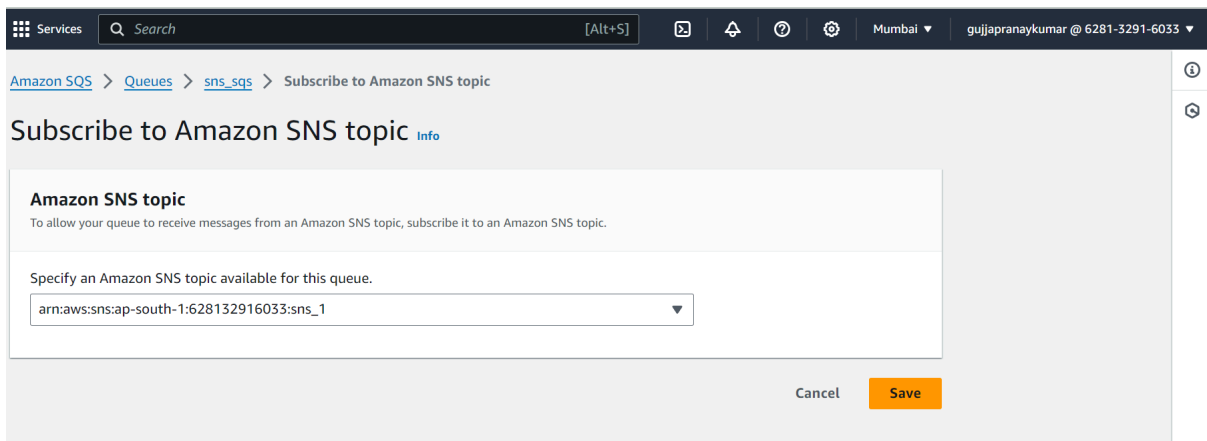
- 1.Navigate to sqs dashboard and click on create queue
- 2.Provide the queue name and leave all the configurations as default and proceed creation of sqs

A:



3. Now you can subscribe to Amazon SNS topic that you have created previously

A:



4. Navigate to the sns topic to view the subscription

5. You can publish a message from SNS topic

A:

he Inbox x Print Share

**sns\_new** <no-reply@sns.amazonaws.com> 4:14 PM (2 minutes ago) ☆ 😊 ↩ ⋮

to me ▼

hi this mail is to test sqs

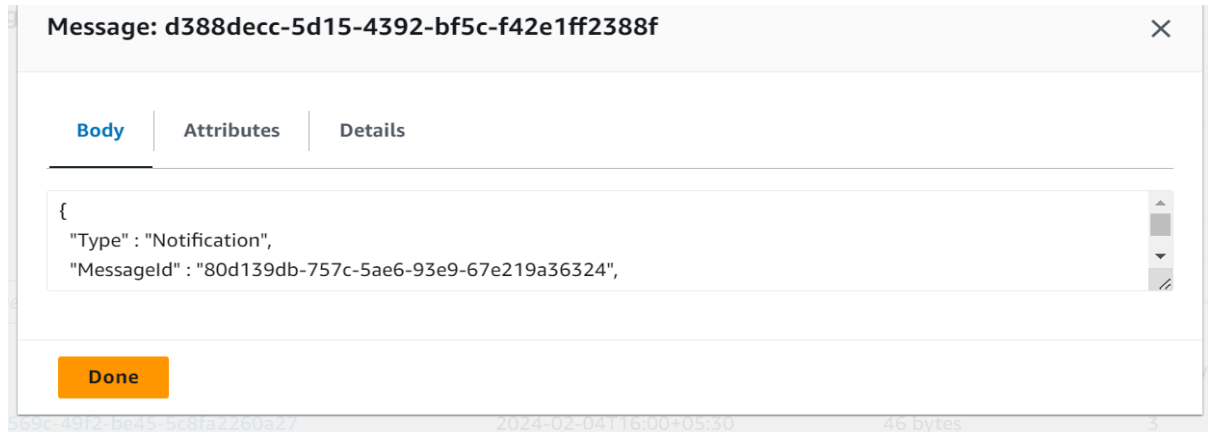
--

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-1:628132916033:sns\\_1:1958592d-c8db-446f-aa3a-1ad86c21397d&Endpoint=pranayguja555@gmail.com](https://sns.ap-south-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:ap-south-1:628132916033:sns_1:1958592d-c8db-446f-aa3a-1ad86c21397d&Endpoint=pranayguja555@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>

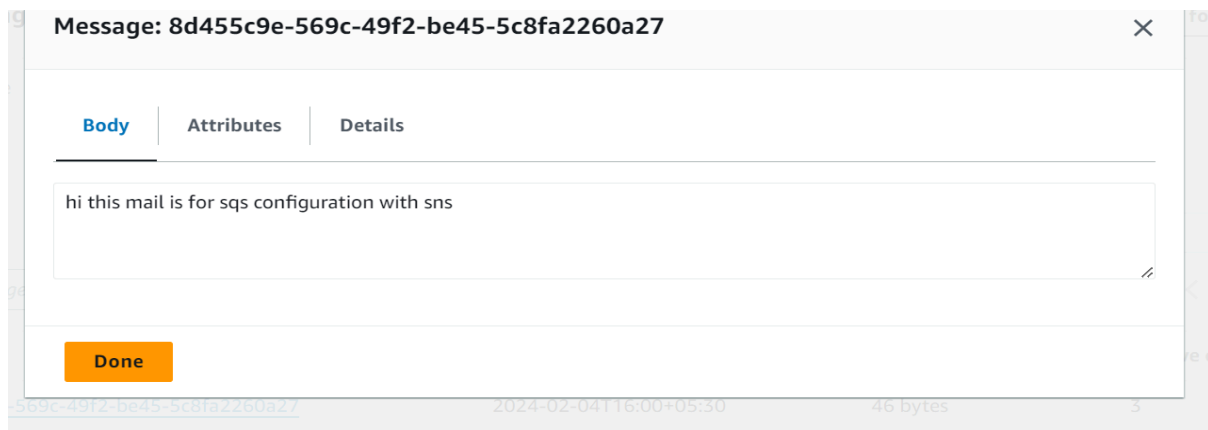
6. Navigate to sqs -> queues -> send and receive messages and verify the message has been received . Click poll for messages and click on the message received

A:



7. You can also send messages from queue and check the messages are updated on clicking poll for messages

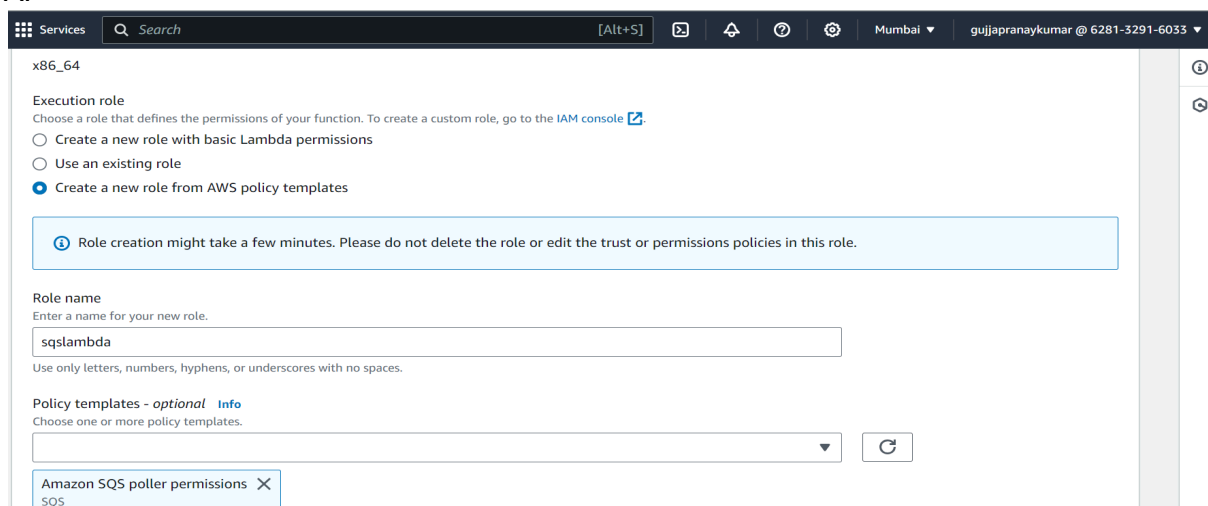
A:



## Create a lambda function and subscribe to sqs

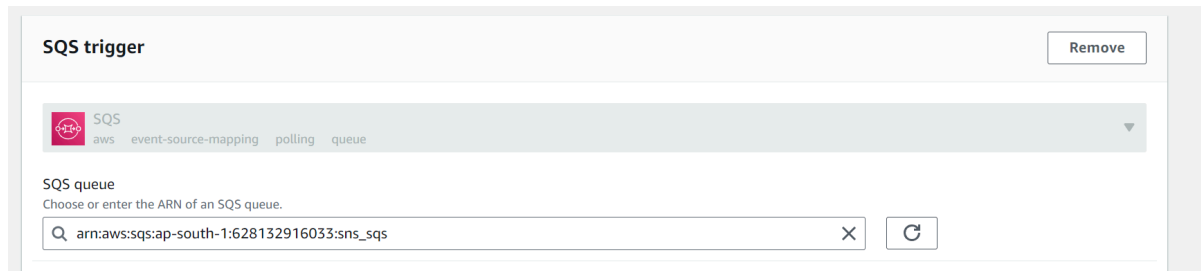
1. Navigate to lambda services and navigate to the function -> create function
2. Select use a blueprint option, search for sqs poller and click
3. Provide function name
4. Create a new role , provide the role name

A:




5. Select the sqs queue you have created, leave all other settings as default and create the function

A:



**SQS trigger** Remove

 **SQS**  
aws event-source-mapping polling queue

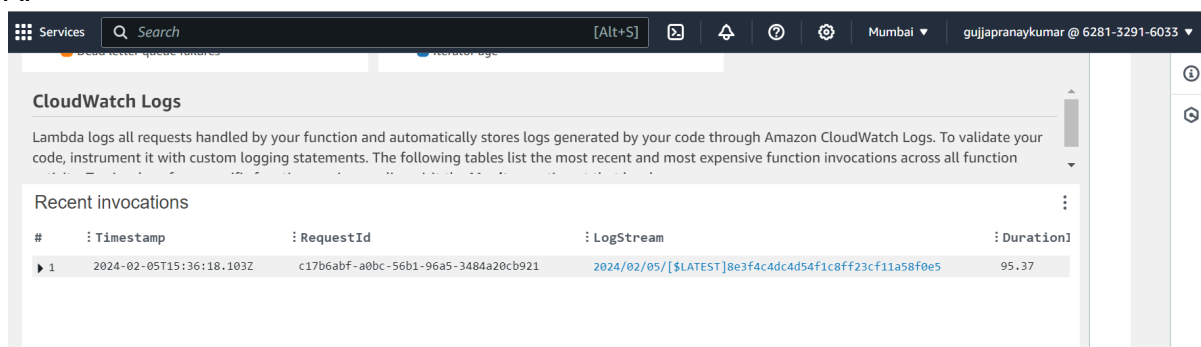
**SQS queue**  
Choose or enter the ARN of an SQS queue.

× ↻

6. Now publish a sns message from the topic you have created

7. Now navigate to lambda functions open the sqs function you have created, navigate to monitor->logs->view logs in cloud watch

A:



**CloudWatch Logs**

Lambda logs all requests handled by your function and automatically stores logs generated by your code through Amazon CloudWatch Logs. To validate your code, instrument it with custom logging statements. The following tables list the most recent and most expensive function invocations across all function

**Recent invocations**

#	Timestamp	RequestId	LogStream	Duration
1	2024-02-05T15:36:18.103Z	c17b6abf-a0bc-56b1-96a5-3484a20cb921	2024/02/05/[\$LATEST]8e3f4c4dc4d54f1c8ff23cf11a58f0e5	95.37

8. Click on the logs to view the message

A:



▶	2024-02-05T21:05:36.460+05:30	END RequestId: 09b2a23f-3341-54c4-894b-fc0431c9390b
▶	2024-02-05T21:05:36.460+05:30	REPORT RequestId: 09b2a23f-3341-54c4-894b-fc0431c9390b Duration: 1.68 ms Billed Durati...
▶	2024-02-05T21:06:18.008+05:30	START RequestId: c17b6abf-a0bc-56b1-96a5-3484a20cb921 Version: \$LATEST
▼	2024-02-05T21:06:18.083+05:30	2024-02-05T15:36:18.083Z c17b6abf-a0bc-56b1-96a5-3484a20cb921 INFO SQS message 9e92460...

```
2024-02-05T15:36:18.083Z c17b6abf-a0bc-56b1-96a5-3484a20cb921 INFO SQS message 9e92460a-cdac-4910-a5d3-613df931636b: {"Type": "Notification", "MessageId": "3705e36c-1c3f-50f9-b5bb-bd571e8eff38", "TopicArn": "arn:aws:sns:ap-south-1:628132916033:sns_1", "Subject": "lambda sqs", "Timestamp": "2024-02-05T15:36:17.909Z", "SignatureVersion": "1", "Signature": "FkUBadFI203oM2hmoh/pP9ZcyXMBjFPDcNYRMc1ky9VXtevrFvksgeis9Z0y2HrTPUiiYQKY8zhvdWlmzJ5BoIesBaYl0/PZV9hNrMv5XLhyKqM4hIZyh12Vphfe1Me28299oxw7u7NsUIf/tmk3PY9kf346+Bj5QLL9XF1Atikkrx2IbnHjQ3/Yx01ceJG3vojMX5riegh998ug5DS8xnI/vKZZTmP7Xc7aUoHx8mGiI7q0mPNRefEaVnd1ncdzQnIfIm6+SzgcCe1RA4Rb2s3q4nRChVOpnFxmHsa/RMYAf4xmL3/M/wwJhGNRfnij1ECRM313XOk83ec7RgQ=", "SigningCertURL": "https://sns.ap-south-1.amazonaws.com/SimpleNotificationService-60eadc530605d63b8e62a523676ef735.pem", "UnsubscribeURL": "https://sns.ap-south-1.amazonaws.com/?Action=Unsubscribe&SubscriptionArn=arn:aws:sns:ap-south-1:628132916033:sns_1:9e11c7f2-5826-47dd-b673-061c1b6946f8"}
Copy
```

## Steps to configure Simple email service

1. Navigate to SES dashboard, you need to create identity

2. Choose the identity type as Email address and provide the email id( To verify ownership of an email address, you must have access to its inbox to open the verification email.)

3. Create the identity

A:

[Amazon SES](#) > Configuration: Verified identities

## Verified identities

A *verified identity* is a domain, subdomain, or email address you use to send email through Amazon SES. [Learn more](#)

**Identities (2)** [Info](#) Send test email Delete Create identity

<input type="checkbox"/>	Identity	Identity type	Identity status
<input type="checkbox"/>	<a href="#">pranaygujja555@gmail.com</a>	Email address	✓ Verified
<input type="checkbox"/>	<a href="#">extram586@gmail.com</a>	Email address	✓ Verified

4. Click on the identity you have created, and click on send a test mail

5. Choose the scenario, subject, body

A:

[Amazon SES](#) > [Configuration: Verified identities](#) > Create identity

## Create identity

A *verified identity* is a domain, subdomain, or email address you use to send email through Amazon SES. Identity verification at the domain level extends to all email addresses under one verified domain identity.

**Identity details** [Info](#)

☐ Domain  
To verify ownership of a domain, you must have access to its DNS settings to add the necessary records.

☒ Email address  
To verify ownership of an email address, you must have access to its inbox to open the verification email.

Email address

Email address can contain up to 320 characters, including plus signs (+), equals signs (=), and underscores ( \_ ).

[Amazon SES](#) > [Configuration: Verified identities](#) > Create identity

## Create identity

☒ Formatted  
Choose this option if you want to construct a simple test message using the form provided. SES takes the information entered in the form and parses it into email format for you.

☐ Raw  
Choose this option if you want to send a more complex test message, such as one that uses HTML or includes attachments. This option requires you to format the entire message yourself.

From-address

Scenario [Info](#)  
Choose the email sending scenario that you want to simulate. Each scenario corresponds to a different recipient email address managed by the mailbox simulator. To specify a custom recipient, select Custom.  

Successful delivery  
success@simulator.amazonses.com

Subject

Body - optional

6. Expand additional configuration you can provide reply to address and cc if you wish to provide



A:

Services

Search

[Alt+S]

Mumbai

gujjapranaykumar @ 6281-3291-6033

▼ Additional configurations - optional

Reply-to

Replies will be sent to the Reply-to address instead of the From-address.

extram586@gmail.com

Address can contain up to 320 characters, including plus signs (+), equal signs (=) and underscores (\_).

Return-path

Bounced messages will be sent to the specified Return-path address.

pranaylucky5@gmail.com

Address can contain up to 320 characters, including plus signs (+), equal signs (=) and underscores (\_).

Cc:

extram586@gmail.com

Address can contain up to 320 characters, including plus signs (+), equal signs (=) and underscores (\_).

Bcc:

Enter tertiary recipient

Address can contain up to 320 characters, including plus signs (+), equal signs (=) and underscores (\_).

Cancel

Send test email

7. Click on send test mail, you should receive a test mail in your inbox

A:

1 of 1,764

<

>

ses successful

Inbox x

pranaygujja555@gmail.com

via amazonses.com

11:36 (14 minutes ago)

☆

😊

↶

⋮

to success, me

▼

hi ur ses is working properly

↶ Reply

↶ Reply to all

↷ Forward

😊

END