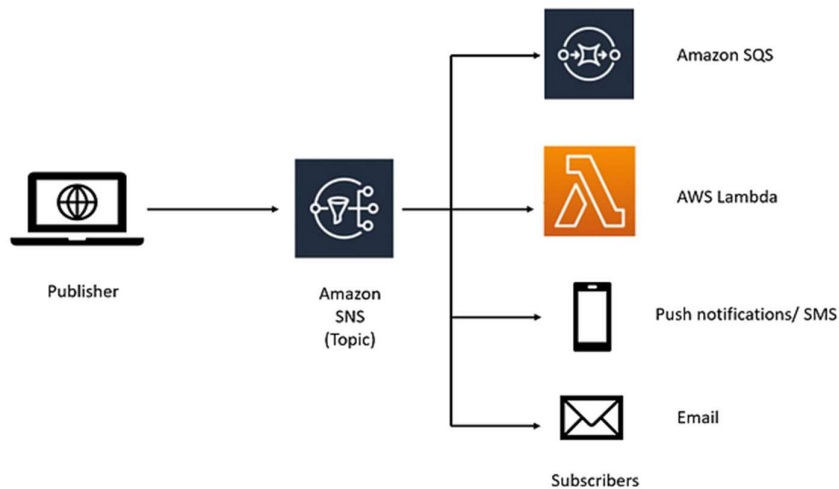


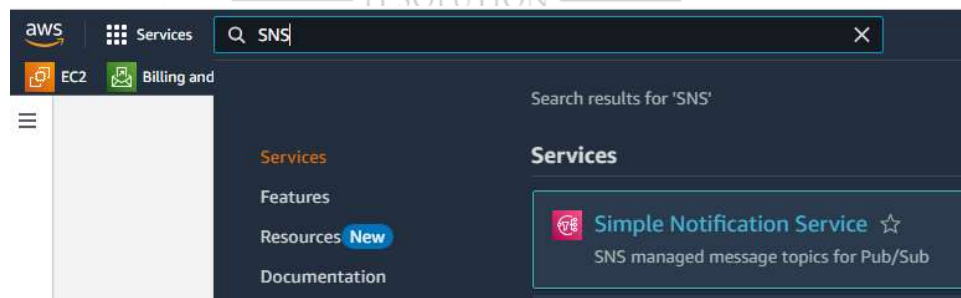
Amazon SNS: (Push Based)

- Amazon Simple Notification Service (Amazon SNS) is a managed service that provides message delivery from publishers to subscribers (also known as producers and consumers).
- Publishers communicate asynchronously with subscribers by sending messages to a topic, which is a logical access point and communication channel.
- Clients can subscribe to the Amazon SNS topic and receive published messages using a supported endpoint type, such as Amazon Data Firehose, Amazon SQS, AWS Lambda, HTTP, email, mobile push notifications, and mobile text messages (SMS).



Steps:

1. Login to AWS Console and search for SNS.



2. Click on Simple Notification Service.
3. Create a Topic.

Create topic

Topic name

A topic is a message channel. When you publish a message to a topic, it fans out the message to all subscribed endpoints.

[Next step](#)

[Start with an overview](#)

4. Go with default settings and click on Create Topic.

Amazon SNS > Topics > Create 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
- 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

test-topic

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.

Test Topic

Maximum 100 characters.

5. Click on Create subscription.

Subscriptions | Access policy | Archive policy | Delivery status logging | Encryption | Tags | Integrations

Subscriptions (0) [Edit](#) [Delete](#)

ID	Endpoint	Status
No subscriptions found You don't have any subscriptions to this topic.		

[Create subscription](#)

6. Select type as email, provide your email ID and click on create subscription.

Create subscription

Details

Topic ARN

arn:aws:sns:ap-south-1:834362069350:test-topic

Protocol

The type of endpoint to subscribe

Email

Endpoint

An email address that can receive notifications from Amazon SNS.

msmanisa18@gmail.com

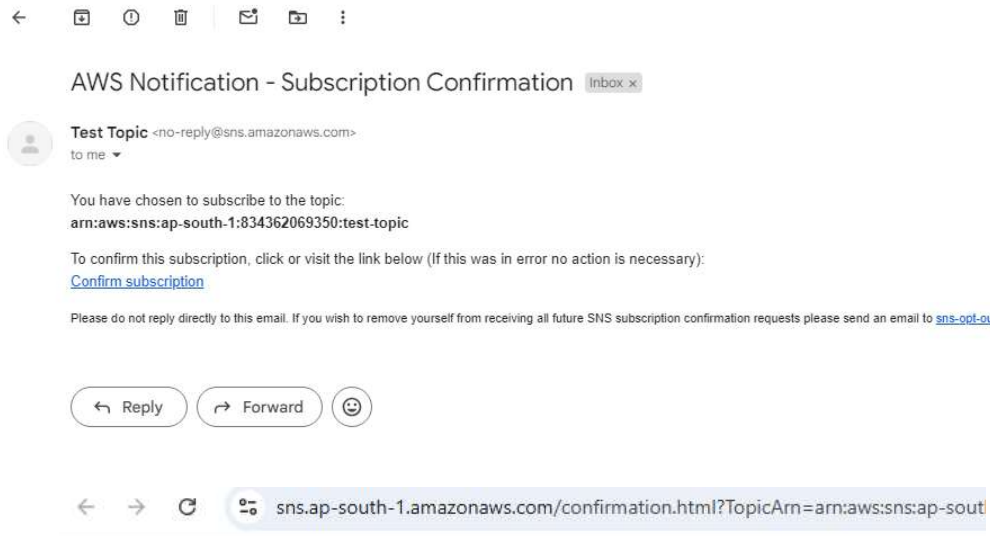
After your subscription is created, you must confirm it. [Info](#)

Subscription filter policy - optional [Info](#)
This policy filters the messages that a subscriber receives.

Redrive policy (dead-letter queue) - optional [Info](#)
Send undeliverable messages to a dead-letter queue.

[Cancel](#) [Create subscription](#)

7. Verify your email, go to your email and click on Confirm Subscription.



Simple Notification Service

Subscription confirmed!

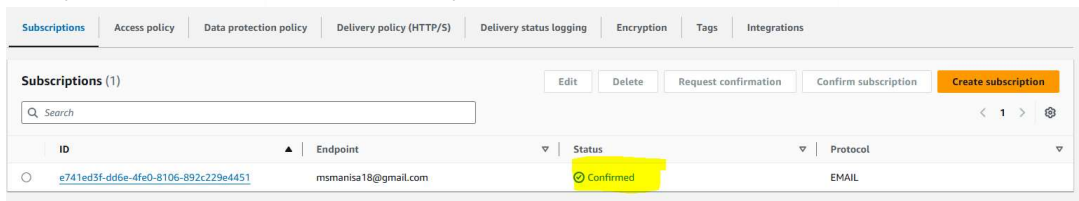
You have successfully subscribed.

Your subscription's id is:

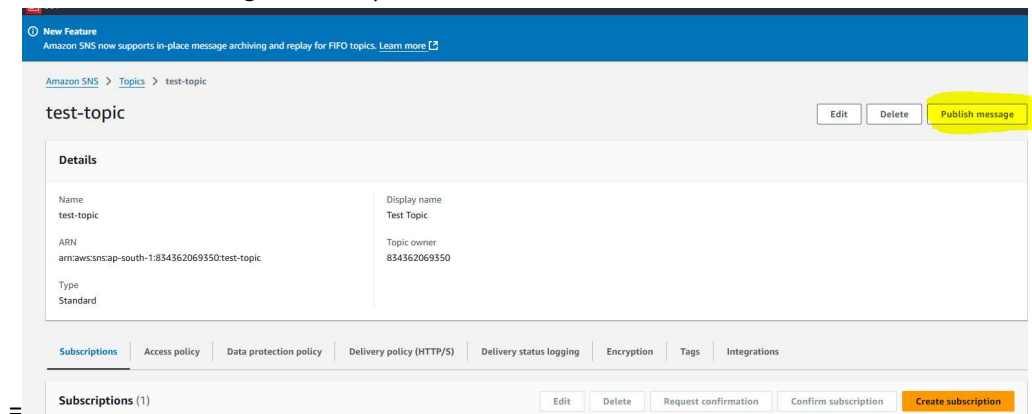
arn:aws:sns:ap-south-1:834362069350:test-topic:e741ed3f-dd6e-4fe0-8106-892c229e4451

If it was not your intention to subscribe, [click here to unsubscribe](#).

8. Now if you check the status under Subscriptions, it should show as Confirmed.



9. Now click on Publish Message under topic.



10. Provide the subject and message body and click on Publish Message.

Topic ARN
arn:aws:sns:ap-south-1:834362069350:test-topic

Subject - *optional*

Welcome Message

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

☒ 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

```

1 Hello,
2
3   This is test message.
4
5 Regards,
6   Team

```

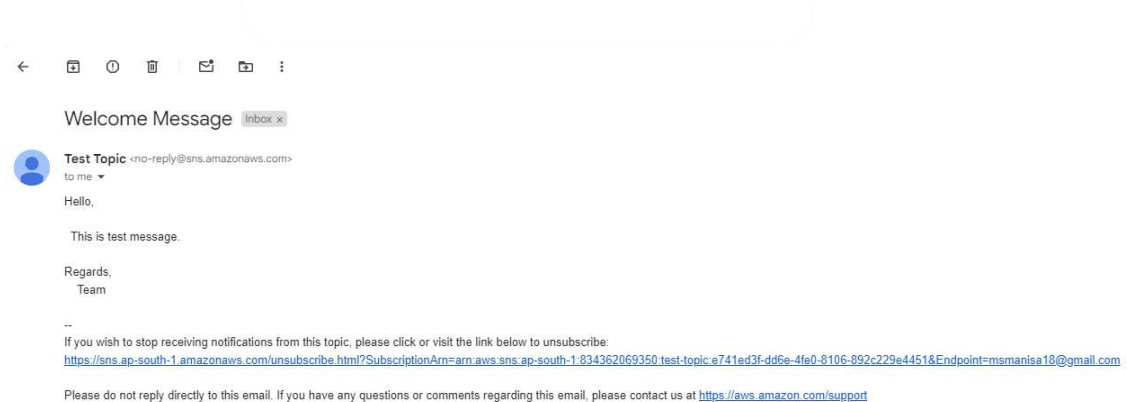
Message attributes [info](#)

Message attributes let you provide structured metadata items (such as timestamps, geospatial data, signatures, and identifiers) for the message.

Type	Name	Value	
Select attribute type ▼	Enter attribute name	value or ["value1", "value2"]	Remove
Add another attribute			

Cancel
Publish message

11. Check your email, you must get the email notification.



Amazon Simple Queue Service (SQS): (Pull Based)

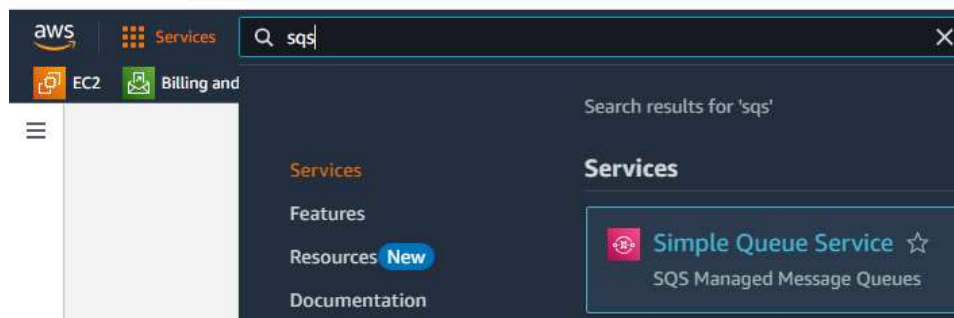
- Amazon Simple Queue Service (Amazon SQS) offers a secure, durable, and available hosted queue that lets you integrate and decouple distributed software systems and components.
- It provides a generic web services API that you can access using any programming language that the AWS SDK supports.

Benefits of using Amazon SQS

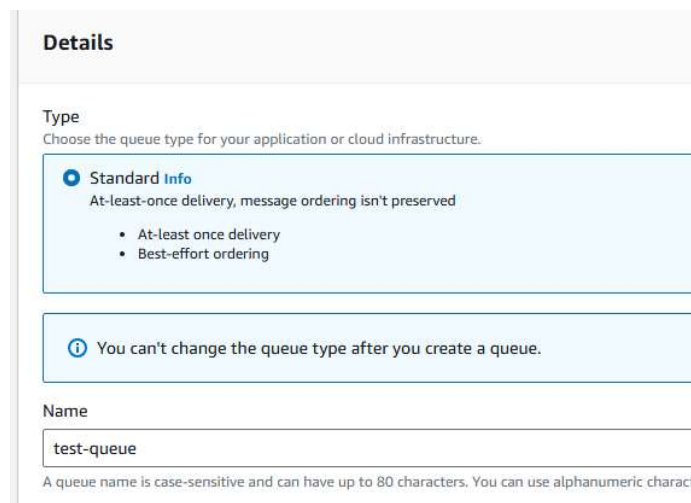
- Security – You control who can send messages to and receive messages from an Amazon SQS queue.
- Durability – For the safety of your messages, Amazon SQS stores them on multiple servers.
- Availability – Amazon SQS uses redundant infrastructure to provide highly-concurrent access to messages and high availability for producing and consuming messages.
- Scalability – Amazon SQS can process each buffered request independently, scaling transparently to handle any load increases or spikes without any provisioning instructions.
- Reliability – Amazon SQS locks your messages during processing, so that multiple producers can send and multiple consumers can receive messages at the same time.
- Customization – Your queues don't have to be exactly alike—for example, you can set a default delay on a queue. You can store the contents of messages larger than 256 KB using Amazon Simple Storage Service (Amazon S3) or Amazon DynamoDB, with Amazon SQS holding a pointer to the Amazon S3 object, or you can split a large message into smaller messages.

Steps:

1. Login to AWS console and search for SQS.



2. Click on Create Queue. Give a name, go with default settings and create the queue.

A screenshot of the 'Details' section in the AWS console for creating a new queue. The 'Type' dropdown is set to 'Standard'. Below it, a note states: 'At-least-once delivery, message ordering isn't preserved'. A list of features for the Standard type is shown: 'At-least once delivery' and 'Best-effort ordering'. A warning message says: 'You can't change the queue type after you create a queue.' The 'Name' field is filled with 'test-queue'. A note at the bottom states: 'A queue name is case-sensitive and can have up to 80 characters. You can use alphanumeric characters.'