

SNS

1. Created sns topic-DG-SNS.

The screenshot shows the Amazon SNS console. On the left is a navigation menu with options: Dashboard, Topics (selected), Subscriptions, Mobile, Push notifications, Text messaging (SMS), and Origination numbers. The main panel displays the 'Details' for the topic 'DG-SNS'. The details include: Name (DG-SNS), Display name (-), ARN (arn:aws:sns:ap-northeast-1:335516814222:DG-SNS), Topic owner (335516814222), and Type (Standard). Below the details are tabs for Subscriptions, Access policy, Data protection policy, Delivery retry policy (HTTP/S), Delivery status logging, Encryption, Tags, and Integrations. The 'Subscriptions' tab is active, showing a list of 3 subscriptions. The list has columns for ID, Endpoint, Status, and Protocol. All three subscriptions are confirmed.

ID	Endpoint	Status	Protocol
1e7f882e-75f3-4f2c-ba58-d930d6...	rrakshithgr@outlook.com	Confirmed	EMAIL
97dc04ca-942a-4b44-8210-75c06...	arn:aws:lambda:ap-northeast-1:33...	Confirmed	LAMBDA
e18c210c-4c9b-4a00-8d03-274cfa...	+918660736768	Confirmed	SMS

->with my email-id,phone number and lambda function named getsnsmessage as the subscribers.

->getsnsmessage lambda function will print the message and subject data and store it in cloudwatch logs.

```
import json
import boto3
import time

def lambda_handler(event, context):
    logs = boto3.client('logs')
    message=event['Records'][0]['Sns']['Subject']
    subject=event['Records'][0]['Sns']['Message']
    print("message:",message)
    print("subject:",subject)

    LOG_GROUP='getsnsmessage007'
    LOG_STREAM='getsnsmessagestream007'
    logs.create_log_group(logGroupName=LOG_GROUP)
    logs.create_log_stream(logGroupName=LOG_GROUP,
logStreamName=LOG_STREAM)
    timestamp = int(round(time.time() * 1000))
```

```

response = logs.put_log_events(
    logGroupName=LOG_GROUP,
    logStreamName=LOG_STREAM,
    logEvents=[
        {
            'timestamp': timestamp,
            'message': message
        }
    ]
)

```

```

response = logs.put_log_events(
    logGroupName=LOG_GROUP,
    logStreamName=LOG_STREAM,
    logEvents=[
        {
            'timestamp': timestamp,
            'message': subject
        }
    ]
)

```

The screenshot shows the AWS CloudWatch console interface. On the left is a navigation sidebar with options like Dashboards, Alarms, Logs, Metrics, X-Ray traces, Events, Application monitoring, and Insights. The main panel displays the 'Log events' page for the log group `/aws/lambda/getsnsmessage` at the latest time. It includes a search bar, filter options, and a table of log events. The events table has columns for 'Timestamp' and 'Message'. The visible events include an 'INIT_START' message, a 'START' message with a request ID, and a 'REPORT' message with duration and memory usage details. There are also 'Copy' buttons next to some messages.

Timestamp	Message
2023-03-03T01:13:44.269+05:30	INIT_START Runtime Version: python:3.8.v15 Runtime Version ARN: arn:aws:lambda:ap-northeast-1::runtime:a20b0c8ab6d995b9d2e883d98c3354...
2023-03-03T01:13:44.518+05:30	START RequestId: 1e98b039-10ab-4009-aaca-0f422819a136 Version: \$LATEST
2023-03-03T01:13:45.285+05:30	message: lambda function
2023-03-03T01:13:45.285+05:30	subject: text from lambda
2023-03-03T01:13:45.664+05:30	END RequestId: 1e98b039-10ab-4009-aaca-0f422819a136
2023-03-03T01:13:45.664+05:30	REPORT RequestId: 1e98b039-10ab-4009-aaca-0f422819a136 Duration: 1146.48 ms Billed Duration: 1147 ms Memory Size: 128 MB Max Memory Us...

-> the above image has the message and subject data and store it in cloudwatch logs.

-> other lambda function named- publishmessage -will use boto3 to publish new messages to the SNS topic.

```
import json
import boto3

def lambda_handler(event, context):
    sns = boto3.client("sns")
    response = sns.publish(
        TopicArn='arn:aws:sns:ap-northeast-1:335516814222:DG-SNS',
        Message='text from lambda',
        Subject='lambda function',
        MessageStructure='string')
```