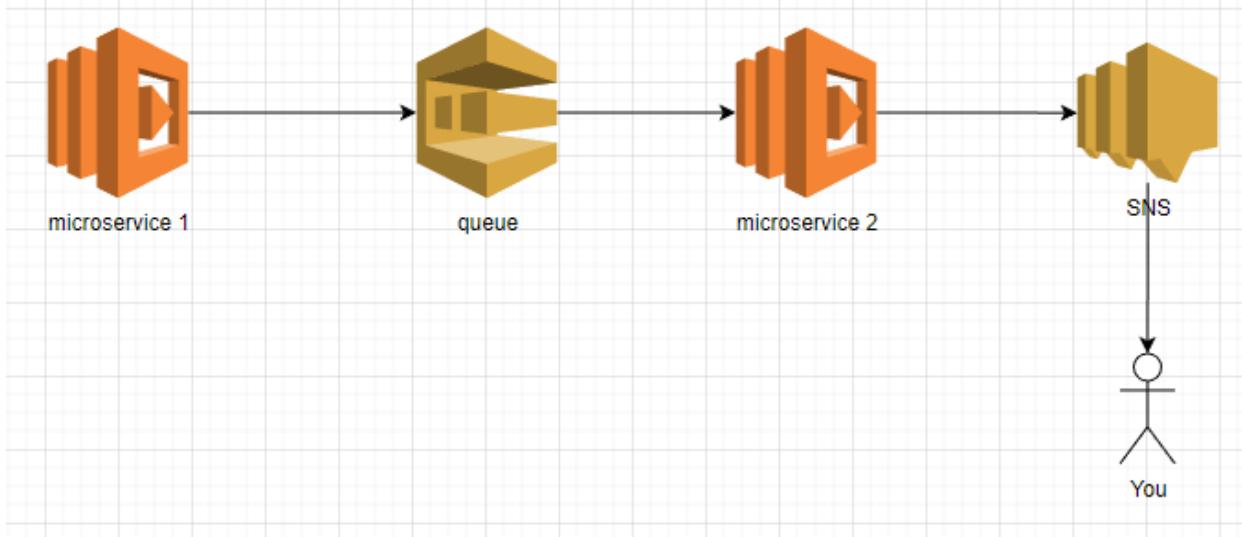


## Assignment 6 – SQS, SNS, and event-driven architecture

### Task 1 – Mock services

Create 2 microservices decoupled with SQS. Once you received the message from the microservice 1 in the microservice 2, send the message to yourself via SNS.



- Create a standard queue.
- Create a SNS topic. Subscribe it with your email.
- In microservice 1,
  - you need the “sendMessage” policy on the queue. Add the policy to the LabRole. You must specify the queue by putting the queue ARN in the policy.
  - Send a message to the queue. Refer [Examples on the official AWS documentation](#) and [the SDK documentation](#) for sending a message to the queue from the microservice 1.
- In microservice 2,
  - you need the “publish” policy to send the message to the topic. Add the policy to the LabRole. You must specify the topic by putting the topic ARN in the policy.
  - Add a SQS trigger and select your queue.
  - You will receive the message in the Event object. Log it out to see where the message sits in the payload.
  - Send the message to the SNS topic. Refer [Publishing Messages in Amazon SNS](#) and [the SDK documentation](#) for sending a message to an SNS

**Task 1:** Create 2 microservices decoupled with SQS. Once you received the message from the microservice 1 in the microservice 2, send the message to yourself via SNS.

#### 1. Create SNS 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

lab6-SNS

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

## 2. Create subscription with email for the topic (SNS) -

## Create subscription

### Details

Topic ARN

arn:aws:sns:us-east-1:242306694058:lab6-SNS X

Protocol

The type of endpoint to subscribe

Email

Endpoint

An email address that can receive notifications from Amazon SNS.

lab6sns@dispostable.com

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

### 3. Create SQS queue -

The screenshot shows the 'Create queue' wizard in the Amazon SQS console. The top navigation bar includes 'Application integration' and the 'Amazon SQS' logo. Below the navigation is a banner for 'Amazon SQS' with the subtitle 'A message queuing service'. A descriptive text explains that Amazon SQS provides queues for high-throughput, system-to-system messaging, used for decoupling processes and buffering work. To the right is a 'Get started' section with a 'Create queue' button, which is highlighted with a red border.

Below the banner, the breadcrumb navigation shows 'Amazon SQS > Queues > Create queue'. The main title is 'Create queue'. The first step, 'Details', is selected. In this step, there is a note: 'Type' followed by 'Choose the queue type for your application or cloud infrastructure.' A callout box contains the message: 'You can't change the queue type after you create a queue.' Two options are available: 'Standard Info' (selected) and 'FIFO Info'. The 'Standard Info' section describes at-least-once delivery and non-preserved ordering, listing 'At-least once delivery' and 'Best-effort ordering'. The 'FIFO Info' section describes first-in-first-out delivery and preserved ordering, listing 'First-in-first-out delivery' and 'Exactly-once processing'. The 'Name' field is filled with 'lab6-SQS'. A note below the name field specifies character restrictions: 'A queue name is case-sensitive and can have up to 80 characters. You can use alphanumeric characters, hyphens (-), and underscores (\_).'

#### 4. Now create policy for lambda functions to publish message to SNS topic and SQS queue -

Create policy

1 2 3

A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. [Learn more](#)

Visual editor    JSON    Import managed policy

Expand all | Collapse all

▼ Select a service    Clone | Remove    Enter service manually

▼ Service    Select a service below  
close    SNS

SNS ⓘ

Actions Choose a service before defining actions

Resources Choose actions before applying resources

Request conditions Choose actions before specifying conditions

+ Add additional permissions

▶ Service SNS

▼ Actions Specify the actions allowed in SNS ⓘ    Switch to deny permissions ⓘ  
close    Filter actions

Manual actions (add actions)  
 All SNS actions (sns:\*)

Access level  
▶  List  
▶  Read  
▶  Tagging  
▼  Write (1 selected)

Expand all | Collapse all

ConfirmSubscription ⓘ     DeleteSMSsandboxPhoneNumber ⓘ     SetSMSAttributes ⓘ  
 CreatePlatformApplication ⓘ     DeleteTopic ⓘ     SetSubscriptionAttributes ⓘ  
 CreatePlatformEndpoint ⓘ     OptInPhoneNumber ⓘ     Subscribe ⓘ  
 CreateSMSSandboxPhoneNumber ⓘ     Publish ⓘ     Unsubscribe ⓘ  
 CreateTopic ⓘ     PutDataProtectionPolicy ⓘ     VerifySMSSandboxPhoneNumber ⓘ  
 DeleteEndpoint ⓘ     SetEndpointAttributes ⓘ  
 DeletePlatformApplication ⓘ     SetPlatformApplicationAttributes ⓘ

▶  Permissions management

▶ Resources Specify topic resource ARN for the Publish action.

**5. Copy the ARN from SNS topic -**

The screenshot shows the AWS SNS Topics page. At the top, there's a breadcrumb navigation: Amazon SNS > Topics > lab6-SNS. Below the navigation, the topic name "lab6-SNS" is displayed. On the right side of the topic name are three buttons: "Edit", "Delete", and "Publish message". Under the "Details" section, there are two columns: "Name" (lab6-SNS) and "Display name" (empty). In the "ARN" column, the value "arn:aws:sns:us-east-1:242306694058:lab6-SNS" is shown and highlighted with a red box. To the right of the ARN is the "Topic owner" information: "Topic owner" (242306694058). Below the "Details" section is a "Subscriptions" tab, which is currently selected. The "Subscriptions (1)" table has columns: "ID", "Endpoint", "Status", and "Protocol". A single row is listed: "Pending confirmation" (ID), "lab6sns@dispostable.com" (Endpoint), "Pending confirmation" (Status), and "EMAIL" (Protocol). The "Create subscription" button is located at the top right of the subscriptions table.

**6. Paste the ARN in add ARN topic name -**

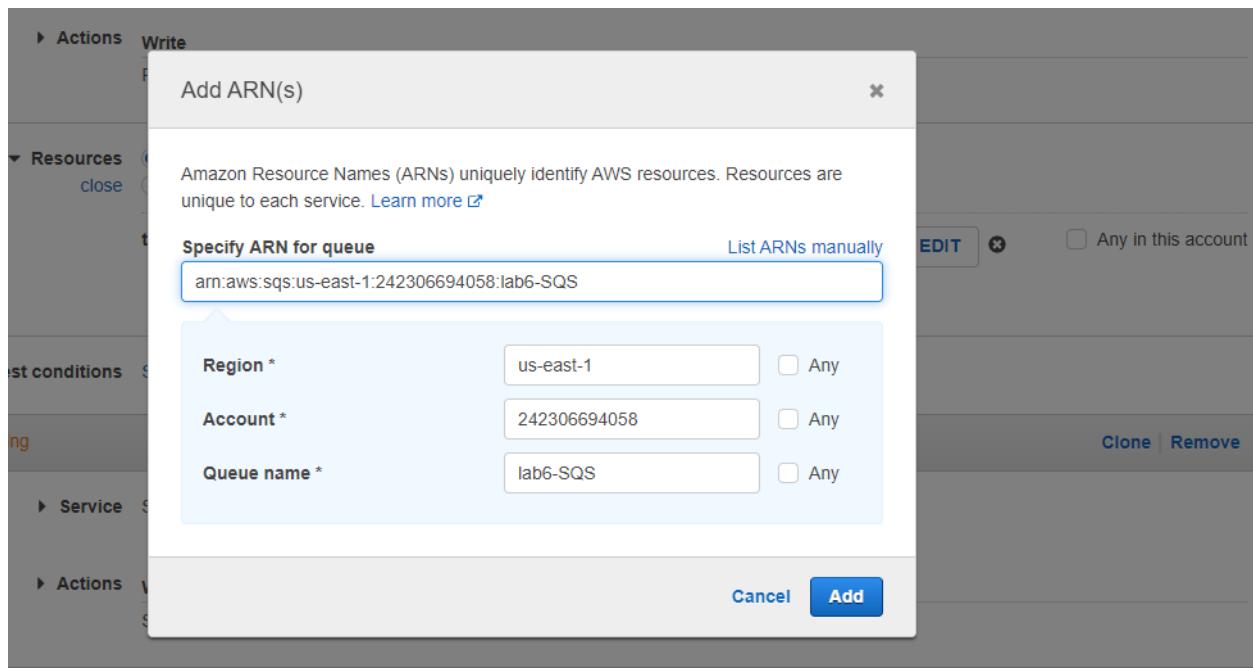
The screenshot shows the AWS IAM "Create policy" dialog. On the left, the policy editor interface is visible, showing a single action: "SNS (1 action) ▲ 1 warning". On the right, a modal dialog titled "Add ARN(s)" is open. The modal contains a descriptive text about ARNs and a "Specify ARN for topic" input field where the ARN "arn:aws:sns:us-east-1:242306694058:lab6-SNS" is entered. Below this, there are three dropdown fields: "Region \*", "Account \*", and "Topic name \*". Each dropdown has a placeholder value ("us-east-1", "242306694058", and "lab6-SNS" respectively) and a "Any" checkbox. At the bottom of the modal are "Cancel" and "Add" buttons. A tooltip at the bottom of the modal says "Specify topic resource ARN for the Publish action." and "Add ARN to restrict access".

## 7. Add additional permission for SQS -

The screenshot shows the AWS IAM Policy Editor interface. At the top, there's a search bar with 'sq' typed into it, and a dropdown menu showing 'Service' and 'Select a service below'. Below the search bar, there's a red box highlighting the 'SQS' option in the dropdown. The main area has sections for 'Actions', 'Resources', and 'Request conditions'. Under 'Actions', a red box highlights the 'Specify the actions allowed in SQS' section. It includes a 'Manual actions (add actions)' section with a checkbox for 'All SQS actions (sns:\*)'. Below that is an 'Access level' section with checkboxes for 'Read', 'Tagging', and 'Write (1 selected)'. Under 'Write', a red box highlights the 'SendMessage' checkbox, which is checked. There are also other actions like 'DeleteQueue', 'CreateQueue', 'PurgeQueue', and 'DeleteMessage'. At the bottom of the 'Actions' section, a red box highlights the 'Resources' section with the instruction 'Specify queue resource ARN for the SendMessage action.' On the right side of the policy editor, there are buttons for 'Clone', 'Remove', 'Cancel', 'Next: Tags', and 'Clone | Remove'.

## 8. Copy ARN from SQS and paste in policy resource ARN -

The screenshot shows the AWS SQS Queue Details page for a queue named 'lab6-SQS'. The top navigation bar includes links for 'Amazon SQS', 'Queues', and 'lab6-SQS'. On the right, there are buttons for 'Edit', 'Delete', 'Purge', and 'Send and receive messages'. Below the navigation, there are tabs for 'Details' and 'Info'. The 'Details' tab shows the queue's name ('lab6-SQS'), type ('Standard'), and encryption settings ('Amazon SQS key (SSE-SQS)'). The 'Info' tab shows the queue's URL ('https://sqs.us-east-1.amazonaws.com/242306694058/lab6-SQS'). A red box highlights a message box in the top right corner that says 'ARN copied.' with a copy icon. To the right of the message, the ARN 'arn:aws:sqs:us-east-1:242306694058:lab6-SQS' is displayed.



**9. Create Policy (if you do not have permission, just copy the JSON and keep it for future reference)**

-

## Create policy

1 2 3

### Review policy

Name\* lab6-policy  
Use alphanumeric and '+=\_@-' characters. Maximum 128 characters.

Description  
Maximum 1000 characters. Use alphanumeric and '+=\_@-' characters.

Summary

Filter			
Service	Access level	Resource	Request condition
Allow (2 of 354 services) Show remaining 352			
SNS	Limited: Write	TopicName   string like   lab6-SNS	None
SQS	Limited: Write	QueueName   string like   lab6-SQS	None

## Create policy

A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the

Visual editor JSON

```
1 {  
2     "Version": "2012-10-17",  
3     "Statement": [  
4         {  
5             "Sid": "VisualEditor0",  
6             "Effect": "Allow",  
7             "Action": [  
8                 "sns:Publish",  
9                 "sns:SendMessage"  
10            ],  
11            "Resource": [  
12                "arn:aws:sns:us-east-1:242306694058:lab6-SNS",  
13                "arn:aws:sqs:us-east-1:242306694058:lab6-SQS"  
14            ]  
15        }  
16    ]  
17}
```

**10. Now create lambda function for FIRST Microservice m1 -**

Lambda > Functions > Create function

## Create function Info

AWS Serverless Application Repository applications have moved to [Create application](#).

### Author from scratch Info

Start with a simple Hello World example.

### Use a blueprint Info

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.

lab6-m1

Use only letters, numbers, hyphens, or underscores with no spaces.

### Runtime Info

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

Node.js 16.x

### Architecture Info

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

x86\_64

arm64

## Permissions Info

By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs.

### ▼ 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

LabRole

[View the LabRole role on the IAM console](#).

**11.** Put the following code inside the lambda -

```
12. const AWS = require("aws-sdk");
13. const sqs = new AWS.SQS({ apiVersion: "2012-11-05"});
14.
15. exports.handler = async (event) => {
16.   await sqs.sendMessage({
17.     MessageBody: event.message,
18.     QueueUrl: "https://sns.us-east-1.amazonaws.com/242306694058/lab6-SQS",
19.   }).promise();
20.
21. const response = {
22.   statusCode: 200,
23.   body: JSON.stringify('Success'),
24. };
25. return response;
26. };
```

The screenshot shows the AWS Lambda Code source editor interface. At the top, there are tabs for 'Code source' and 'Info'. Below the tabs is a menu bar with File, Edit, Find, View, Go, Tools, Window, and a toolbar with Test, Deploy, and a status message 'Changes not deployed'. The main area is titled 'index.js'. On the left, there's a sidebar labeled 'Environment' which shows a folder named 'lab6-m1 - /' containing an 'index.js' file. The code editor displays the following JavaScript code:

```
1  const AWS = require("aws-sdk");
2  const sqs = new AWS.SQS({ apiVersion: "2012-11-05"});
3
4  exports.handler = async (event) => {
5    await sqs.sendMessage({
6      MessageBody: event.message,
7      QueueUrl: "https://sns.us-east-1.amazonaws.com/242306694058/lab6-SQS",
8    }).promise();
9
10 const response = {
11   statusCode: 200,
12   body: JSON.stringify('Success'),
13 };
14 return response;
15
16
```

## 12. Deploy and create a test event for testing the lambda -

A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function's invocation result.

To invoke your function without saving an event, configure the JSON event, then choose Test.

### Test event action

Create new event

Edit saved event

### Event name

test-m1

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

### Event sharing settings

Private

This event is only available in the Lambda console and to the event creator. You can configure a total of 10. [Learn more](#)

Shareable

This event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)

### Template - optional

hello-world



### Event JSON

[Format JSON](#)

```
1 {  
2   "message": "message from m1"  
3 }
```

## 13. Test result -

The screenshot shows the AWS Lambda Test console interface. At the top, there are tabs for 'Code source' and 'Info'. Below that is a menu bar with 'File', 'Edit', 'Find', 'View', 'Go', 'Tools', 'Window', 'Test' (which is highlighted in orange), and 'Deploy'. To the right of the menu is an 'Upload from' button with a dropdown arrow. The main area has tabs for 'index.js' and 'Execution result'. Under 'Execution result', the 'Test Event Name' is set to 'test-m1'. The 'Response' section shows a JSON object with 'statusCode': 200 and 'body': '\u005c'. The 'Function Logs' section displays the execution details: START RequestId: c58279a8-df03-4ecc-b70e-6053195ac2b2 Version: \$LATEST, END RequestId: c58279a8-df03-4ecc-b70e-6053195ac2b2, REPORT RequestId: c58279a8-df03-4ecc-b70e-6053195ac2b2 Duration: 978.06 ms Billed Duration: 979 ms Memory Size: 128 MB Max Memory Used: 83 MB Init Duration: 425.20 ms. The 'Request ID' is also listed as c58279a8-df03-4ecc-b70e-6053195ac2b2.

**14. Now check in SQS with message polling if the message is sent to queue -**

The image consists of three vertically stacked screenshots of the AWS SQS console.

**Screenshot 1: Queue Details**

This screenshot shows the details of the queue "lab6-SQS". The "Send and receive messages" button is highlighted with a red box.

Name	Type	ARN
lab6-SQS	Standard	arn:aws:sqs:us-east-1:242306694058:lab6-SQS

**Screenshot 2: Receive Messages**

This screenshot shows the "Receive messages" interface. The "Poll for messages" button is highlighted with a red box.

Messages available	Polling duration	Maximum message count	Polling progress
0	30	10	27% 1 receives/second

**Screenshot 3: Receive Messages (After Message Sent)**

This screenshot shows the "Receive messages" interface after a message has been sent. The "Polling progress" section shows a green circle with a checkmark and the text "1 receives/second".

Messages available	Polling duration	Maximum message count	Polling progress
1	30	10	1 receives/second

**15. Check message body -**

This screenshot shows the details of a specific message in the queue.

**Message: 90b79ef5-8e55-40d5-a821-e0bcf9ab3849**

The "Body" tab is selected, showing the message content: "message from m1".

**Details    Body    Attributes**

**Done**

## 16. Now create SECOND microservice m2 -

Lambda > Functions > Create function

### Create function Info

AWS Serverless Application Repository applications have moved to [Create application](#).

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     Select a cont

#### Basic information

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

Use only letters, numbers, hyphens, or underscores with no spaces.

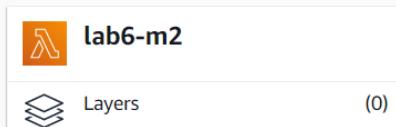
Runtime Info  
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.  
 ▾

## 17. Add trigger for SQS -

Lambda > Functions > lab6-m2

lab6-m2

▼ Function overview [Info](#)



[+ Add trigger](#)

[+ Add destination](#)

### Add trigger

Trigger configuration [Info](#)



#### SQS queue

Choose or enter the ARN of an SQS queue.



lab6-SQS



[Activate trigger](#)

Select to activate the trigger now. Keep unchecked to create the trigger in a deactivated state for testing (recommended).

#### Batch size - optional

The number of records in each batch to send to the function.

10

The maximum is 10,000 for standard queues and 10 for FIFO queues.

#### Batch window - optional

The maximum amount of time to gather records before invoking the function, in seconds.

When the batch size is greater than 10, set the batch window to at least 1 second.

**18.** Write the following code inside lambda –

```
const AWS = require("aws-sdk");
const sns = new AWS.SNS({ apiVersion: "2010-03-31"});

exports.handler = async (event) => {
    console.log(JSON.stringify(event));

    await sns.publish({
        Message: event.Records[0].body,
        TopicArn: "arn:aws:sns:us-east-1:242306694058:lab6-SNS",
    }).promise();

    const response = {
        statusCode: 200,
        body: JSON.stringify('Success! Check your mail'),
    };
    return response;
};
```

Code source [Info](#)

File Edit Find View Go Tools Window Test Deploy Changes not deployed

Go to Anything (Ctrl-P)

Environment lab6-m2 - index.js

index.js

```
1 const AWS = require("aws-sdk");
2 const sns = new AWS.SNS({ apiVersion: "2010-03-31"});

3 exports.handler = async (event) => {
4     console.log(JSON.stringify(event));

5     await sns.publish({
6         Message: event.Records[0].body,
7         TopicArn: "arn:aws:sns:us-east-1:242306694058:lab6-SNS",
8     }).promise();

9     const response = {
10         statusCode: 200,
11         body: JSON.stringify('Success! Check your mail'),
12     };
13     return response;
14 };
15 };
16 };
17 };
18 }
```

**19.** Deploy and test Microservice 1 again and check your email if Microservice 2 is triggered and send message through SNS – Make sure you have confirmed the subscription from your email

 **Disposable**

inbox

## Inbox for lab6sns@dispostable.com

Continuously check for new messages

From	Subject	Date
01000184f0a671d1-9494c007-4519-4b1c-9560-bc5fe77fc92-000000@amazonses.com	<a href="#">AWS Notification Message</a>	8 Dec 2022, 07:32
01000184ef4141c4-21da6ee6-52a2-4e7d-ab2f-74129e715cb2-000000@amazonses.com	<a href="#">AWS Notification - Subscription Confirmation</a>	8 Dec 2022, 01:02

[Contact](#) | Unread messages older than 2 days, and read older than 2 months are automatically deleted.

 **Disposable**

lab6sns' inbox > "AWS Notification Message"

## Message "AWS Notification Message"

**From:** AWS Notifications <no-reply@sns.amazonaws.com>

[Back to inbox](#) | [Download EML](#) | [Delete message](#)

message from m1

--  
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:  
<https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:242306694058:lab6-SNS:af72a749-cc3c-40e0-bbf4-93a932f36283&Endpoint=lab6sns@dispostable.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>

[Contact](#) | Unread messages older than 2 days, and read older than 2 months are automatically deleted.

## Task 2 – Mock services with Step Functions.

Do the same task but using Step Functions.

1. First 2 new lambda functions for this without having any custom code or trigger like task 1 since we will design the workflow with step function -

## Basic information

### Function name

Enter a name that describes the purpose of your function.

lab6-m1-step

Use only letters, numbers, hyphens, or underscores with no spaces.

### Runtime [Info](#)

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

Node.js 16.x

### Architecture [Info](#)

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

- x86\_64
- arm64

### 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.

## ▼ 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

## Basic information

### Function name

Enter a name that describes the purpose of your function.

lab6-m2-step

Use only letters, numbers, hyphens, or underscores with no spaces.

### Runtime [Info](#)

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

Node.js 16.x

### Architecture [Info](#)

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

- x86\_64
- arm64

### 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.

## ▼ 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

2. Now create the step function state machine -

The screenshot shows the AWS Step Functions console with the following details:

- Header:** Application integration, AWS Step Functions, Assemble functions into business-critical applications.
- Get started:** Run a Hello World workflow in just a few clicks. A yellow box highlights the "Get started" button.
- Pricing (US):** Pricing information for the service.
- Step Functions sidebar:**
  - State machines:** Selected.
  - Activities
  - Data flow simulator
  - Feature spotlight
  - Local Development
  - Join our feedback panel
- State machines list:** Shows 0 state machines. A red box highlights the "Create state machine" button.
- Create state machine wizard:**
  - Step 1: Choose authoring method:**
    - Design your workflow visually:** Selected. Description: Drag and drop your workflow together with Step Functions Workflow Studio. A green box highlights this section.
    - Write your workflow in code:** Description: Author your workflow using Amazon States Language. You can generate code snippets to easily build out your workflow steps.
    - Run a sample project:** Description: Deploy and run a fully functioning sample project in minutes using CloudFormation.
  - Step 2: Type:**
    - Standard:** Selected. Description: Durable, checkpointed workflows for machine learning, order fulfillment, IT/DevOps automation, ETL jobs, and other long-duration workloads.
    - Express:** Description: Event-driven workflows for streaming data processing, microservices orchestration, IoT data ingestion, mobile backends, and other short duration, high-event-rate workloads.
  - Buttons:** Help me decide, Next (highlighted with a red box), and Cancel.

**3. In designer, first drag and drop Lambda (invoke) and configure with microservice 1 -**

The screenshot shows the AWS Workflow Studio interface. On the left, there's a search bar with 'aws lambda invoke' and a sidebar with categories like MOST POPULAR, COMPUTE, and others. The main workspace has a flowchart starting with a 'Start' node, followed by a 'Lambda: Invoke invoke m1' step, and ending with an 'End' node. To the right, the 'invoke m1' configuration panel is open, showing:

- Configuration:** State name is set to 'invoke m1'.
- API:** Lambda: Invoke, with a note to specify the function name.
- Integration type:** Info, set to 'Optimized'.
- API Parameters:** A dropdown menu is open, showing options like 'AssignmentOneLambda:\$LATEST', 'MainMonitoringFunction:\$LATEST', etc., with 'lab6-m1-step:\$LATEST' selected.

The second part of the screenshot shows a similar configuration for another Lambda invoke step, with the 'Select function name' dropdown also showing 'lab6-m1-step:\$LATEST'.

**4. Now drag and drop SQS (SendMessage) and configure -**

The screenshot shows the AWS Workflow Studio interface. On the left, there's a search bar with 'sq' and a sidebar with categories like MOST POPULAR, COMPUTE, and others. The main workspace has a flowchart starting with a 'Start' node, followed by a 'Lambda: Invoke invoke m1' step, then an 'SQS: SendMessage SQS SendMessage' step, and finally an 'End' node. To the right, the 'SQS SendMessage' configuration panel is open, showing:

- Configuration:** State name is set to 'SQS SendMessage'.
- API:** SQS, with a note to enter the queue URL.
- Select queue URL:** The URL 'https://sqs.us-east-1.amazonaws.com/242306694058/lab6-SQS' is entered.

5. Now drag and drop second lambda invoke and configure -

6. Finally include SNS (SNS publish) component in the last step and configure -

7. Go next and you can check the generated code -

```

1 + {
2 +   "Comment": "A description of my state machine",
3 +   "StartAt": "invoke m1",
4 +   "States": [
5 +     {
6 +       "Invoke m1": {
7 +         "Type": "Task",
8 +         "Resource": "arn:aws:states:::lambda:invoke",
9 +         "OutputPath": "$.Payload",
10 +         "Parameters": {
11 +           "Payload.$": "$",
12 +           "FunctionName": "arn:aws:lambda:us-east-1:242306694058:function:lab6-m1-step:$LATEST"
13 +         },
14 +         "Retry": [
15 +           {
16 +             "ErrorEquals": [
17 +               "Lambda.ServiceException",
18 +               "Lambda.SdkClientException",
19 +               "Lambda.SdkClientException",
20 +               "Lambda.SdkClientException"
21 +             ],
22 +             "IntervalSeconds": 2,
23 +             "MaxAttempts": 6,
24 +             "BackoffRate": 2
25 +           }
26 +         }
27 +       }
28 +     },
29 +     {
30 +       "SQS SendMessage": {
31 +         "Type": "Task",
32 +         "Resource": "arn:aws:states:::aws-sqs:sendMessage"
33 +       }
34 +     },
35 +     {
36 +       "invoke m2": {
37 +         "Type": "Task",
38 +         "Resource": "arn:aws:states:::lambda:invoke"
39 +       }
40 +     },
41 +     {
42 +       "SNS Publish": {
43 +         "Type": "Task",
44 +         "Resource": "arn:aws:states:::sns:publish"
45 +       }
46 +     }
47 +   ]
48 + }

```

## 8. Next and create the state machine -

Step Functions > State machines > Create state machine

Step 1  
Choose authoring method

Step 2  
Design workflow

Step 3 - optional  
Review generated code

Step 4  
Specify state machine settings

### Specify state machine settings

**Name**

State machine name  
 Must be 1-80 characters. Can use alphanumeric characters, dashes, or underscores.

**Permissions**

Execution role  
The IAM role that defines which resources your state machine has permission to access during execution. To create a custom role, go to the [IAM console](#).

Create new role  
Let Step Functions create a new role for you based on your state machine's definition and configuration details.

Choose an existing role  
Choose an existing IAM role for your state machine.

Enter a role ARN  
Enter the ARN of an existing IAM role.

Existing roles  
 [Edit](#) [View details](#)

## 9. Start execution -

Step Functions > State machines > lab6-step-function

### lab6-step-function

[Edit](#) [Start execution](#) [Delete](#) [Actions ▾](#)

**Details**

ARN arn:aws:states:us-east-1:242306694058:stateMachine:lab6-step-function	Type Standard
IAM role ARN arn:aws:iam::242306694058:role/LabRole	Creation date Dec 8, 2022 02:39:38.058 AM

[Executions](#) [Logging](#) [Definition](#) [Tags](#)

**Executions (0)**

[Filter by status](#) [View details](#) [Stop execution](#) [Start execution](#)

Name	Status	Started	End Time
No executions			

[Start execution](#)

**Start execution**

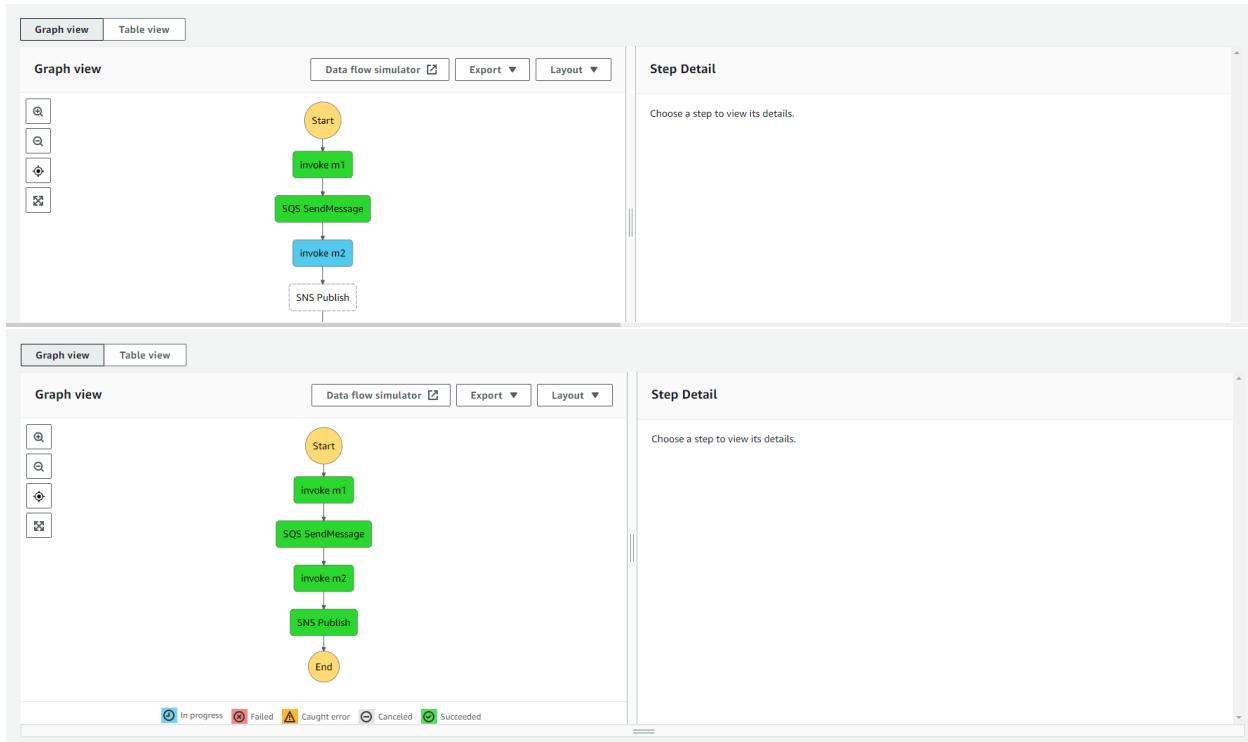
Start an execution using the latest definition of the state machine. [Learn more](#)

Name - optional

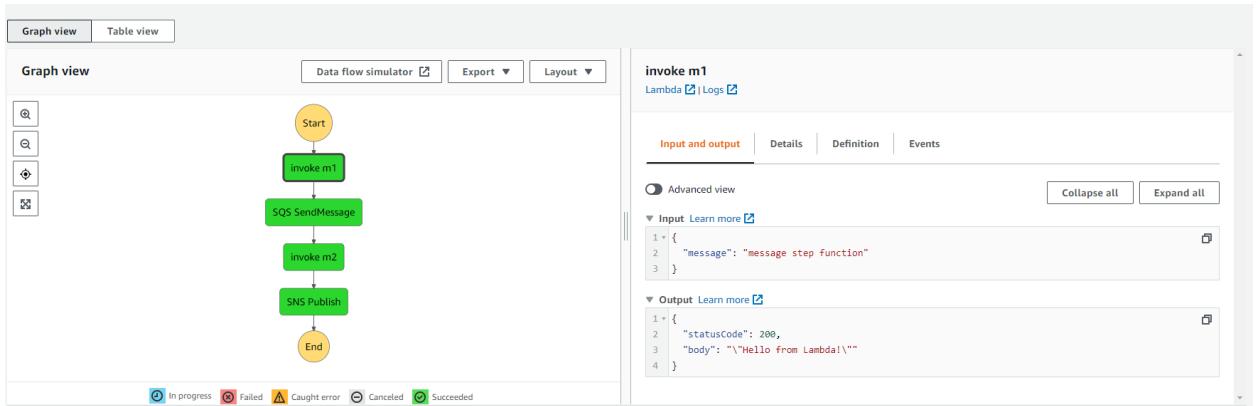
Input - optional  
Enter input values for this execution in JSON format

```
1: [
  2:   "message": "message step function"
  3: ]
```

Open in a new browser tab [Cancel](#) [Start execution](#)



**10. By clicking each step, you can see the details of each one with input and output of the step -**



**11. Finally** check your email to get the message in inbox which is the has been sent from SNS –

The screenshot shows an email from AWS Notifications with the subject "AWS Notification Message". The message body contains the JSON payload {"statusCode":200,"body":"Hello from Lambda!"}. It includes unsubscribe and support links, and a note about message deletion.

**Message "AWS Notification Message"**

**From:** AWS Notifications <no-reply@sns.amazonaws.com>

[Back to inbox](#) | [Download EML](#) | [Delete message](#)

```
{"statusCode":200,"body":"Hello from Lambda!"}
```

--  
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:  
<https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:242306694058:lab6-SNS:af72a749-cc3c-40e0-bbf4-93a932f36283&Endpoint=lab6sns@dispostable.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>

[Contact](#) | Unread messages older than 2 days, and read older than 2 months are automatically deleted.

Task 3 – S3 event notification with EventBridge.

### Simple Notification Service with Event Bridge

Task: Send email notification on object creation in s3 buckets.

#### 1.Creating bucket in S3 service

The screenshot shows the 'Create bucket' configuration page. A red box highlights the 'Bucket name' field, which is set to 'bucket-with-event-bridge-enabled'. Other fields include 'AWS Region' (US East (N. Virginia) us-east-1) and 'Object Ownership' (ACLs disabled). The 'Choose bucket' button is also visible.

**Create bucket** [Info](#)

Buckets are containers for data stored in S3. [Learn more](#)

**General configuration**

Bucket name:  Bucket name must be globally unique and must not contain spaces or uppercase letters. See rules for bucket naming

AWS Region: US East (N. Virginia) us-east-1

Copy settings from existing bucket - optional  
Only the bucket settings in the following configuration are copied.  
[Choose bucket](#)

**Object Ownership** [Info](#)

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

ACLs disabled (recommended)  
All objects in this bucket are owned by this account

ACLs enabled  
Objects in this bucket can be owned by other AWS accounts

#### 2.Enable event bridge in bucket

The screenshot shows the 'Edit Amazon EventBridge' configuration page in the Amazon S3 console. On the left, there's a sidebar with 'Buckets', 'Storage Lens', and other options. The main area has a heading 'Amazon EventBridge' with a sub-section about notifications. A radio button for 'On' is selected and highlighted with a red box. At the bottom right are 'Cancel' and 'Save changes' buttons.

### 3.Create topics and subscription in SNS service

The screenshot shows the 'Create topic' page in the Amazon SNS console. It includes sections for 'Details', 'Type' (with 'FIFO' and 'Standard' options), 'Name' (set to 'MyNewTopic'), and 'Display name - optional' (set to 'MyNewTopic'). The 'Name' field and the 'Standard' topic type are highlighted with red boxes.

aws Services Search [Alt+S] N. Virginia v vocabs/user2292642=chandrika @ 5263-1737-0942

Amazon SNS > Subscriptions > Create subscription

## Create subscription

**Details**

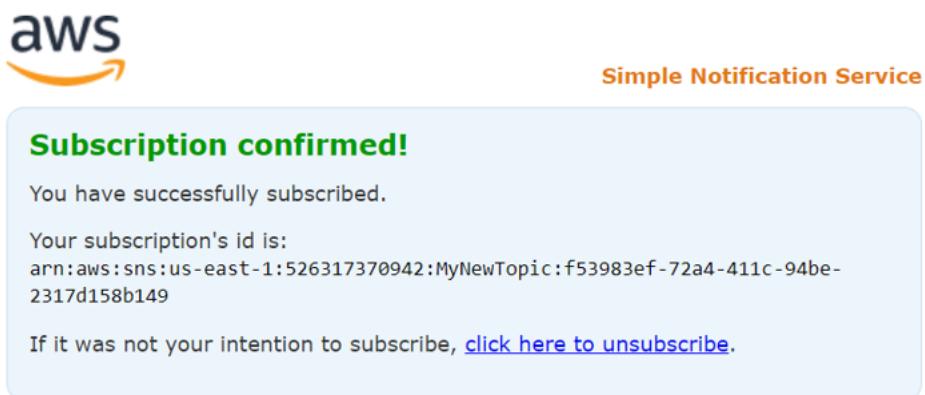
Topic ARN  
arn:aws:sns:us-east-1:526317370942:MyNewTopic

Protocol  
Email

Endpoint  
chandrika.thapa@miu.edu

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

4.verify email address



5. create rule in amazon event bridge service

The screenshot shows the Amazon EventBridge homepage. On the left, there's a navigation sidebar with sections like 'Developer resources', 'Buses', 'Pipes', and 'Scheduler'. The main content area features a large heading 'Amazon EventBridge' and a sub-heading 'A serverless service for building event-driven applications'. Below this, a paragraph explains that EventBridge connects application components to build scalable event-driven apps. A 'How it works' section includes a thumbnail of a video titled 'Serverless 101: Amazon EventBridge - old'. To the right, a 'Get started' section lists four options: 'EventBridge Rule' (selected), 'EventBridge Pipes', 'EventBridge Schedule', and 'EventBridge Schema registry'. A prominent orange 'Create rule' button is highlighted with a red box. At the bottom, there's a 'Pricing' section.

The screenshot shows the 'Define rule detail' step in the AWS Lambda Step Functions interface. On the left, a vertical sidebar lists steps: 'Step 1 Define rule detail' (current), 'Step 2 Build event pattern', 'Step 3 Select target(s)', 'Step 4 - optional Configure tags', and 'Step 5 Review and create'. The main panel is titled 'Rule detail'. It contains fields for 'Name' (set to 'object-created-rule'), 'Description - optional' (set to 'object-created-rule'), 'Event bus' (set to 'default'), and 'Enable the rule on the selected event bus' (checked). There are two tabs for 'Rule type': 'Rule with an event pattern' (selected) and 'Schedule'. The 'Rule with an event pattern' tab has a description: 'A rule that runs when an event matches the defined event pattern. EventBridge sends the event to the specified target.'

Services Search [Alt+S] N. Virginia v vocabs/user2292642=chandrika @ 5263-1737-0942

Event pattern [Info](#)

Event source [AWS services](#)

AWS service [Simple Storage Service \(S3\)](#)

Event type [Amazon S3 Event Notification](#)

S3 Event Notifications will only match your rules if you have configured your S3 bucket(s) to publish event notifications to EventBridge. [Learn more.](#)

Any event  Specific event(s) [Object Created X](#)

Any bucket  Specific bucket(s) by name [bucket-with-event-bridge-enabled](#) [Remove](#)

[Add](#)

Event pattern  
Event pattern, or filter to match the events

```
1 {  
2   "source": ["aws.s3"],  
3   "detail-type": ["Object Created"],  
4   "detail": {  
5     "bucket": {  
6       "name": ["bucket-with-event-bridge-enabled"]  
7     }  
8   }  
9 }
```

[Copy](#) [Test pattern](#) [Edit pattern](#)

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Services Search [Alt+S] N. Virginia v vocabs/user2292642=chandrika @ 5263-1737-0942

## Review and create

Step 2 Build event pattern

Step 3 Select target(s)

Step 4 - optional Configure tags

Step 5 Review and create

### Step 1: Define rule detail

Define rule detail

Rule name	Status	Event bus
object-created-rule	Enabled	default
Description	Rule type	
object-created-rule	Standard rule	

Edit

Services Search [Alt+S] N. Virginia v vocabs/user2292642=chandrika @ 5263-1737-0942

### Step 2: Build event pattern

#### Event pattern Info

```
1 {
2   "source": ["aws.s3"],
3   "detail-type": ["Object Created"],
4   "detail": {
5     "bucket": {
6       "name": ["bucket-with-event-bridge-enabled"]
7     }
8 }
```

Permissions Note: When using the EventBridge console, EventBridge will automatically configure the proper permissions for the selected targets. If you're using the AWS CLI, SDK, or CloudFormation, you'll need to configure the proper permissions.

Build event pattern

Select target(s)

Configure tags

Review and create

### Target 1

Target types

Select an EventBridge event bus, EventBridge API destination (SaaS partner), or another AWS service as a target.

EventBridge event bus

EventBridge API destination

AWS service

Select a target Info

SNS topic

Topic

MyNewTopic

Additional settings

Add another target Cancel Previous Next

aws Services Search [Alt+S] N. Virginia vocabs/user2292642=chandrika @ 5263-1737-0942

### Select Event Bus

Amazon EventBridge

Developer resources

- Learn
- Sandbox
- Quick starts

Buses

- Event buses
- Rules**
- Global endpoints
- Archives
- Replays

Pipes

- Pipes **New**

Scheduler

- Schedules

Event bus

Select or enter event bus name

default

Rules (7/7)

Find rules Any status < 1 > **Create rule**

<input type="checkbox"/>	Name	Status	Type	Description
<input type="checkbox"/>	MonitoringRule	Enabled	Standard	MonitoringRule
<input type="checkbox"/>	myrulechan	Enabled	Standard	a rule to send email notification when creating object in s3 bucket
<input type="checkbox"/>	object-created-rule	Enabled	Standard	object-created-rule
<input type="checkbox"/>	resourceFunctionRule	Enabled	Standard	
<input type="checkbox"/>	voc-codebuild-cw-rule	Enabled	Standard	codebuild build state change events
<input type="checkbox"/>	voc-ec2-cw-rule	Enabled	Standard	ec2 state change events

## 5. Upload object in bucket and check email

aws Services Search [Alt+S] Global vocabs/user2292642=chandrika @ 5263-1737-0942

Drag and drop files and folders you want to upload here, or choose Add files, or Add folders.

Files and folders (1 Total, 357.3 KB)

All files and folders in this table will be uploaded.

Find by name < 1 >

<input type="checkbox"/>	Name	Folder	Type	Size
<input type="checkbox"/>	Assignment 3 - S3.pdf	-	application/pdf	357.3 KB

Destination

Destination <s3://bucket-with-event-bridge-enabled>

Destination details

Bucket settings that impact new objects stored in the specified destination.

Permissions

Grant public access and access to other AWS accounts.

Properties

Specify storage class, encryption settings, tags, and more.

Cancel **Upload**

## AWS Notification Message



M

MyNewTopic <no-reply@sns.amazonaws.com>



...

To: Chandrika Thapa

Tue 06/12/2022 16:26

```
{"version":"0","id":"9f2f8ac2-de55-5b1e-7bb7-a9e8dad8274c","detail-type":"Object Created","source":"aws.s3","account":"526317370942","time":"2022-12-06T22:25:58Z","region":"us-east-1","resources":["arn:aws:s3:::bucket-with-event-bridge-enabled"],"detail":{"version":"0","bucket":{"name":"bucket-with-event-bridge-enabled"},"object":{"key":"Assignment 3 - S3.pdf","size":365840,"etag":"1ed443272bcb2d870973f28faa96ada3","sequencer":"00638FC175E4934A46"},"request-id":"9K518JQ6JKS4W4WA","requester":"526317370942","source-ip-address":"209.152.96.166","reason":"PutObject"}}
```

--

If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:

[https://sns.us-east-1.amazonaws.com/unsubscribe.html?](https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:526317370942:MyNewTopic:f53983ef-72a4-411c-94be-2317d158b149&Endpoint=chandrika.thapa@miu.edu)

[SubscriptionArn=arn:aws:sns:us-east-1:526317370942:MyNewTopic:f53983ef-72a4-411c-94be-2317d158b149&Endpoint=chandrika.thapa@miu.edu](https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:526317370942:MyNewTopic:f53983ef-72a4-411c-94be-2317d158b149&Endpoint=chandrika.thapa@miu.edu)

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>

Reply

Forward