

## Assignment – 1 - Lambda

### Problem Statement:

You work for XYZ Corporation. Your corporation wants to launch a new web-based application and they do not want their servers to be running all the time. It should also be managed by AWS. Implement suitable solutions.

### Tasks To Be Performed:

1. Create a sample Python Lambda function.
2. Set the Lambda Trigger as SQS and send a message to test invocations

## Create a sample Python Lambda function.

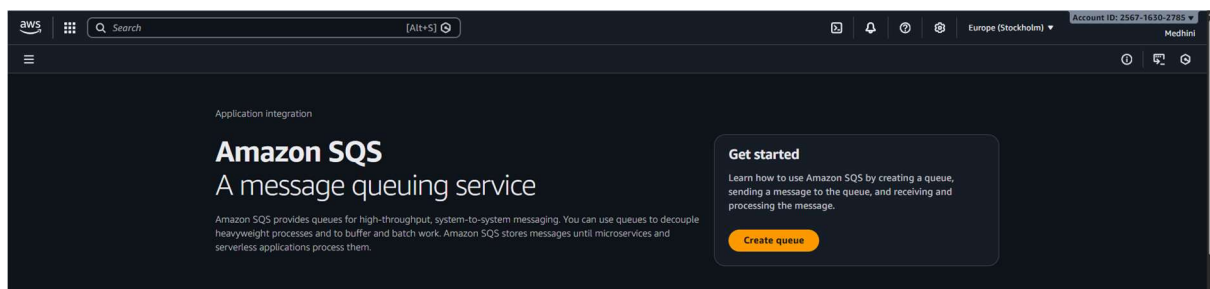
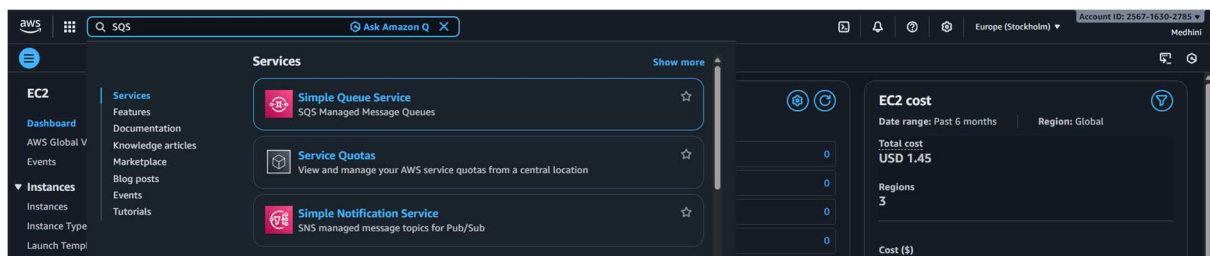
### STEP 1: Create an SQS Queue

#### Open SQS

- Login to AWS Console
- Search SQS
- Click Create queue

#### Queue Settings

- Type: Standard
- Queue name: xyz-lambda-queue
- Leave all defaults
- Click Create queue



## Assignment – 1 - Lambda

The screenshot shows the 'Create queue' page in the AWS Management Console. The 'Details' section has 'Standard' selected as the queue type. The 'Name' field is 'xyz-lambda-queue'. The 'Configuration' section shows a 'Visibility timeout' of 30 seconds and a 'Message retention period' of 4 days.

**Create queue**

**Details**

Type  
Choose the queue type for your application or cloud infrastructure.

☒ **Standard** info  
At-least-once delivery, message ordering isn't preserved

- At-least once delivery
- Best-effort ordering

☐ **FIFO** info  
First-in-first-out delivery, message ordering is preserved

- First-in-first-out delivery
- Exactly-once processing

You can't change the queue type after you create a queue.

Name  
xyz-lambda-queue  
A queue name is case-sensitive and can have up to 80 characters. You can use alphanumeric characters, hyphens (-), and underscores (\_).

**Configuration** info  
Set the maximum message size, visibility to other consumers, and message retention.

Visibility timeout info  
30 Seconds  
Should be between 0 seconds and 12 hours.

Message retention period info  
4 Days  
Should be between 1 minute and 14 days.

The screenshot shows the 'xyz-lambda-queue' page in the AWS Management Console. A green banner at the top indicates the queue was created successfully. The 'Details' section shows the queue name, type (Standard), ARN, and URL. The 'Access policy' section shows the default policy.

**Queue xyz-lambda-queue created successfully**  
You can now send and receive messages.

**xyz-lambda-queue** Edit Delete Purge Send and receive messages Start DLQ redrive

**Details** info

Name xyz-lambda-queue	Type Standard	ARN arn:aws:sqs:eu-north-1:256716302785:xyz-lambda-queue
Encryption Amazon SQS key (SSE-SQS)	URL https://sqs.eu-north-1.amazonaws.com/256716302785/xyz-lambda-queue	Dead-letter queue -

► More

**Queue policies** Monitoring SNS subscriptions Lambda triggers EventBridge Pipes Dead-letter queue Tagging Encryption Dead-letter queue redrive tasks

**Access policy** info Edit  
Define who can access your queue.

```
{
  "Version": "2012-10-17",
  "Id": "default_policy_id",
  "Statement": [
    {
      "Sid": "AllowAccess"
    }
  ]
}
```

## STEP 2: Create IAM Role for Lambda

Lambda needs permission to read messages from SQS.

### Open IAM

- Go to **IAM - Roles**
- Click **Create role**

### Select Trusted Entity

- Select **AWS Service**
- Choose **Lambda**
- Click **Next**

### Attach Policies

Attach these policies:

## Assignment – 1 - Lambda

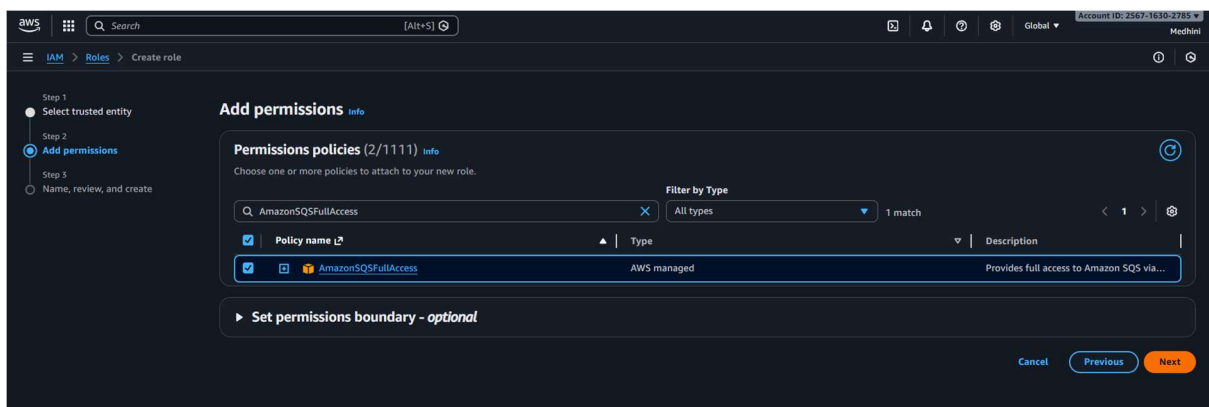
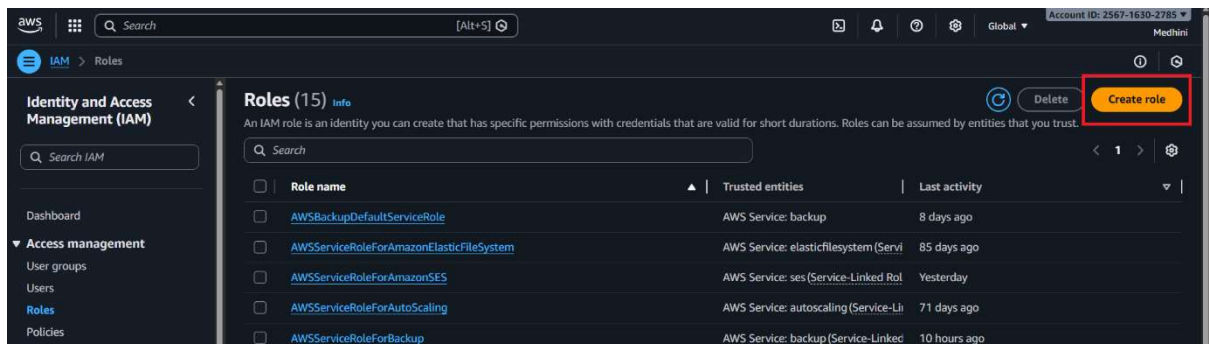
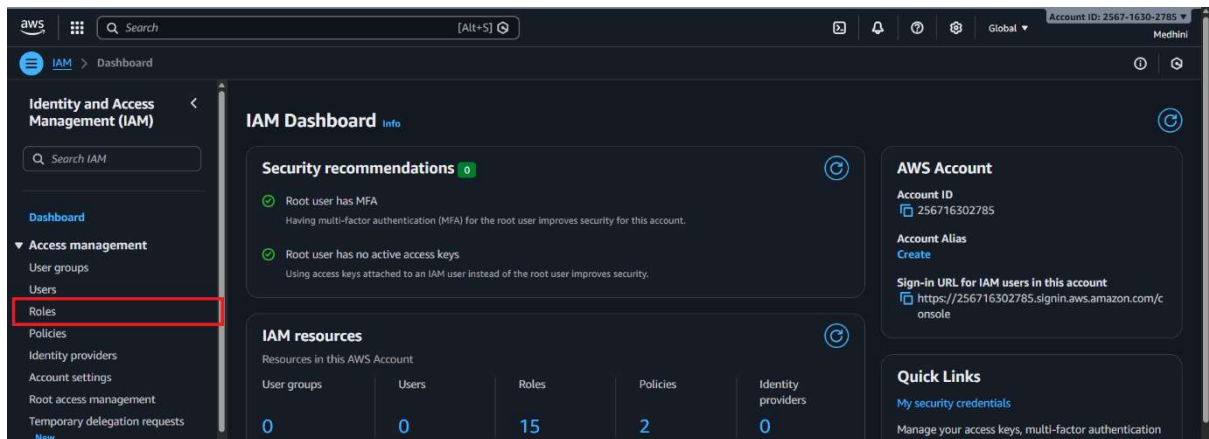
AWSLambdaBasicExecutionRole

AmazonSQSFullAccess (for learning purpose)

Click **Next**

**Role Name**

- Role name: lambda-sqs-role
- Click **Create role**



## Assignment – 1 - Lambda

The screenshot shows the 'Add permissions' step in the AWS IAM console. The left sidebar indicates the progress: Step 1 (Select trusted entity), Step 2 (Add permissions), and Step 3 (Name, review, and create). The main content area is titled 'Add permissions' and includes a search bar with 'AWSLambdaBasicExecutionRole' and a filter dropdown set to 'All types', showing 3 matches. A table lists the policies:

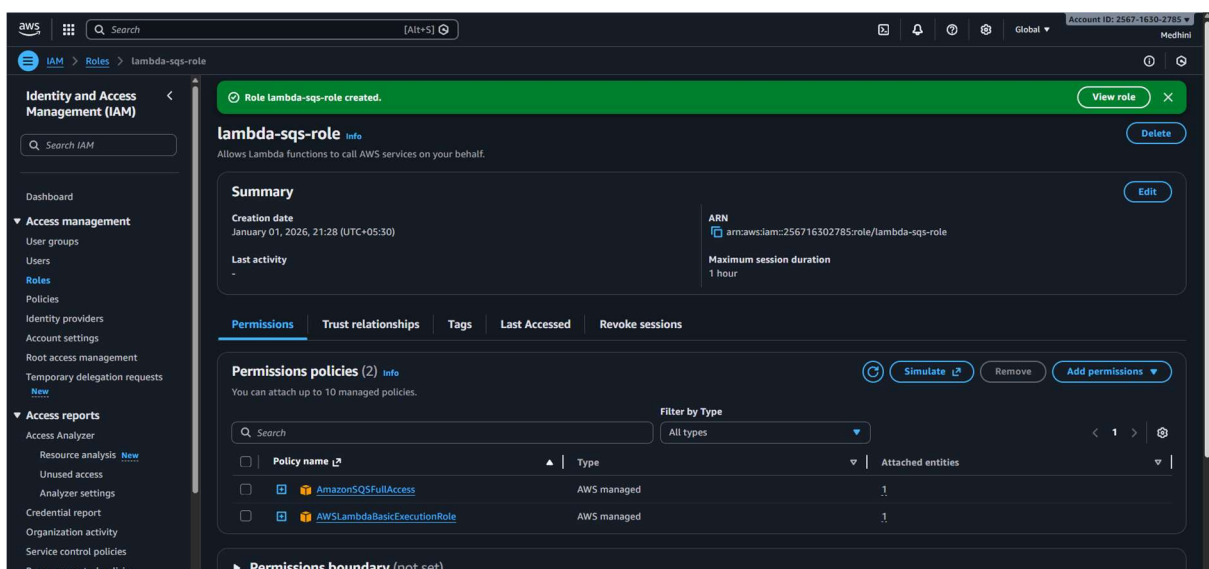
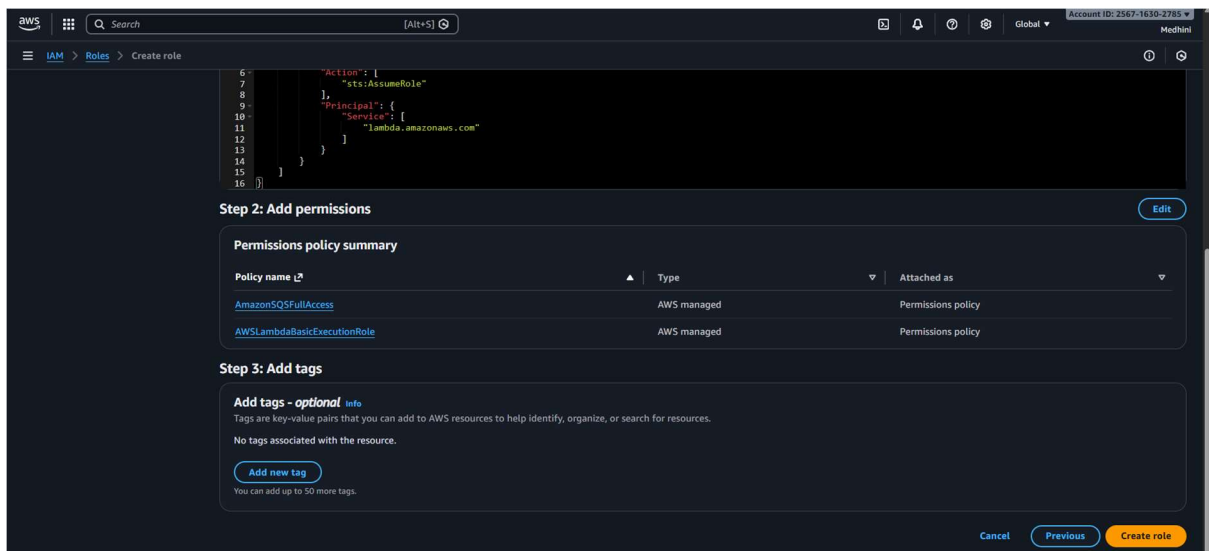
Policy name	Type	Description
<input checked="" type="checkbox"/> AWSLambdaBasicExecutionRole	AWS managed	Provides write permissions to CloudWa...
<input type="checkbox"/> AWSLambdaBasicExecutionRole-023b85e2-9079-473d-8a73-c0fe...	Customer managed	-
<input type="checkbox"/> AWSLambdaBasicExecutionRole-cdcd4ac6-9bfd-4bd6-ba34-2270...	Customer managed	-

Below the table is a link: 'Set permissions boundary - optional'. At the bottom right are 'Cancel', 'Previous', and 'Next' buttons.

The screenshot shows the 'Select trusted entity' step in the AWS IAM console. The left sidebar indicates the progress: Step 1 (Select trusted entity), Step 2 (Add permissions), and Step 3 (Name, review, and create). The main content area is titled 'Select trusted entity' and includes a 'Trusted entity type' section with radio buttons for 'AWS service', 'AWS account', 'Web identity', 'SAML 2.0 federation', and 'Custom trust policy'. The 'AWS service' option is selected and highlighted with a red box. Below this is a 'Use case' section with a dropdown menu for 'Service or use case' set to 'Lambda', also highlighted with a red box. Under 'Use case', the 'Lambda' option is selected. At the bottom right are 'Cancel' and 'Next' buttons, with the 'Next' button highlighted by a red box.

The screenshot shows the 'Name, review, and create' step in the AWS IAM console. The left sidebar indicates the progress: Step 1 (Select trusted entity), Step 2 (Add permissions), and Step 3 (Name, review, and create). The main content area is titled 'Name, review, and create' and includes a 'Role details' section. The 'Role name' field contains 'lambda-sqs-role' and has a note: 'Maximum 64 characters. Use alphanumeric and \*+.,@\_- characters.' The 'Description' field contains 'Allows Lambda functions to call AWS services on your behalf.' and has a note: 'Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: \_+.,@-/\*\$%&'^()#`~:|'". At the bottom left is the text 'Step 1: Select trusted entities' and at the bottom right is an 'Edit' button.

## Assignment – 1 - Lambda



### STEP 3: Create Python Lambda Function

#### Open Lambda

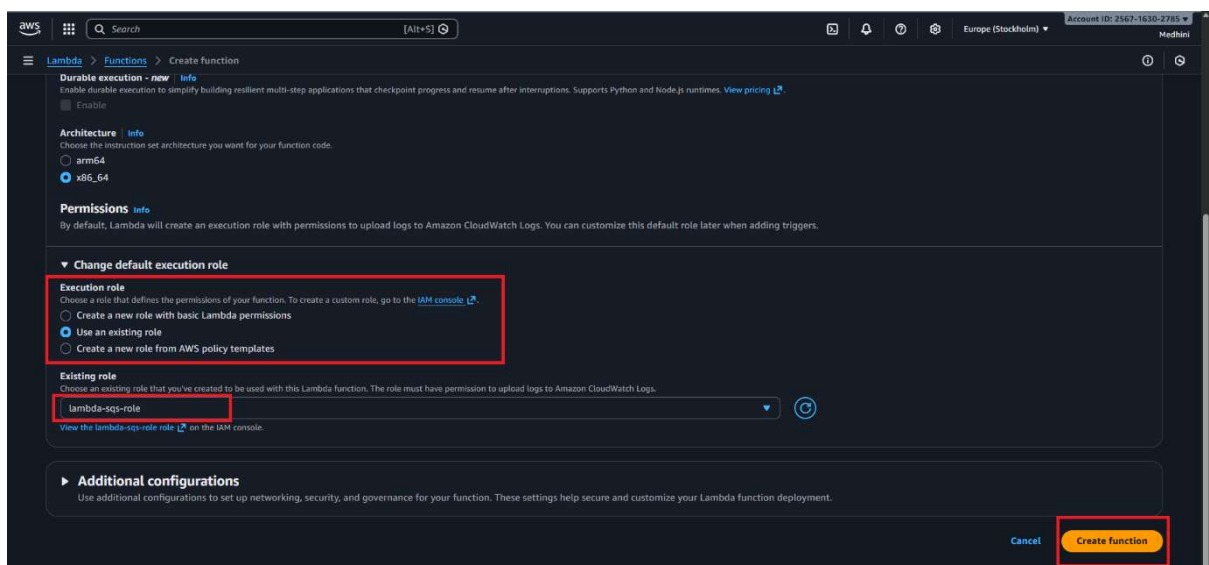
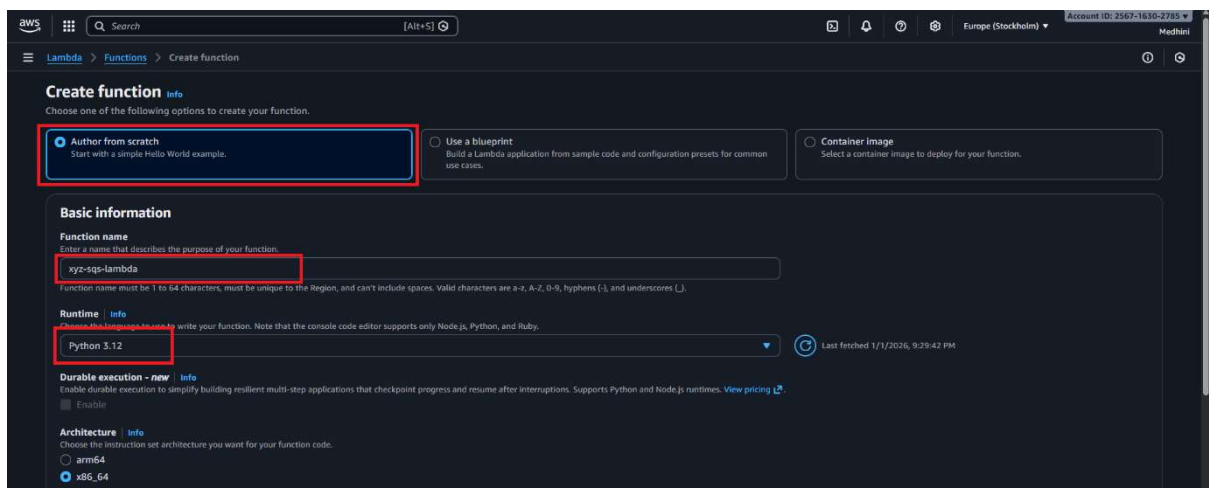
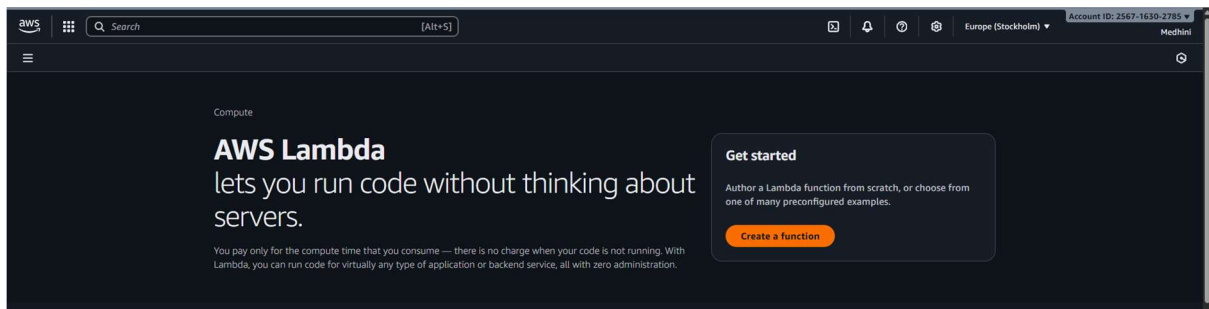
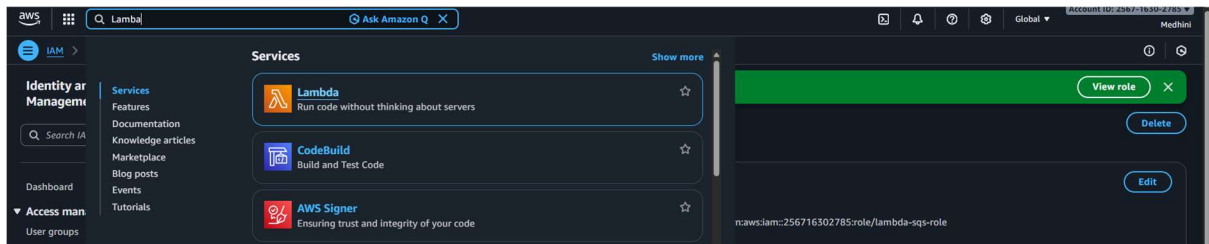
- Search **Lambda**
- Click **Create function**

#### Function Settings

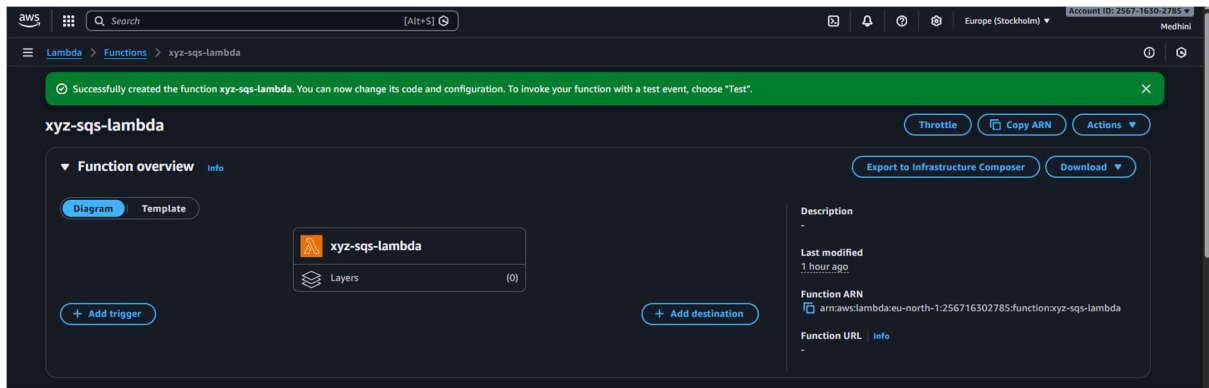
- **Author from scratch**
- **Function name:** xyz-sqs-lambda
- **Runtime:** Python 3.12
- **Execution role:** Use existing role
- Select **lambda-sqs-role**

## Assignment – 1 - Lambda

Click **Create function**



## Assignment – 1 - Lambda



### STEP 4: Add Python Code to Lambda

Replace default code with this

```
import json

def lambda_handler(event, context):

    for record in event['Records']:

        message_body = record['body']

        print("Message received from SQS:", message_body)

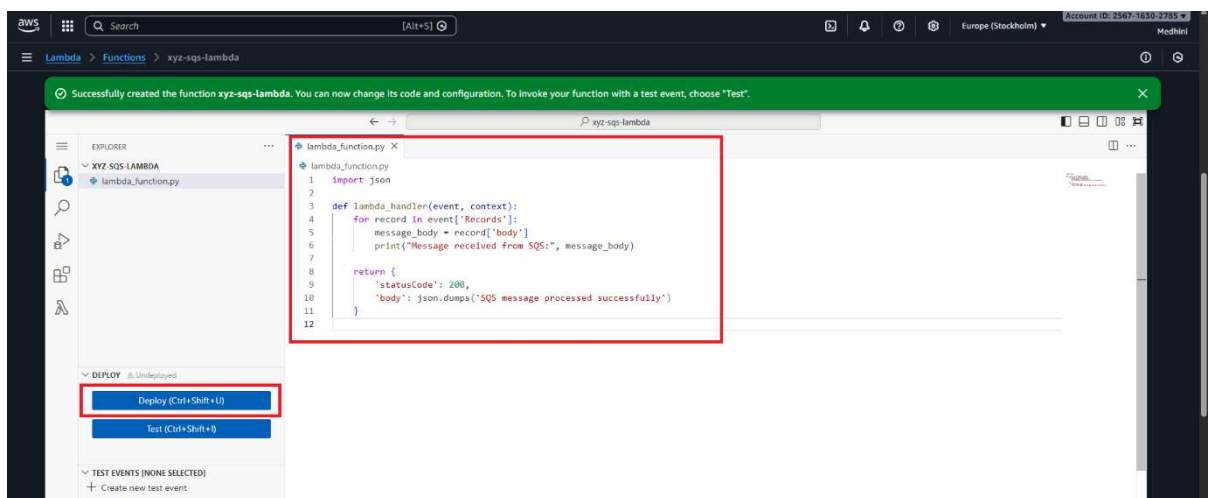
    return {

        'statusCode': 200,

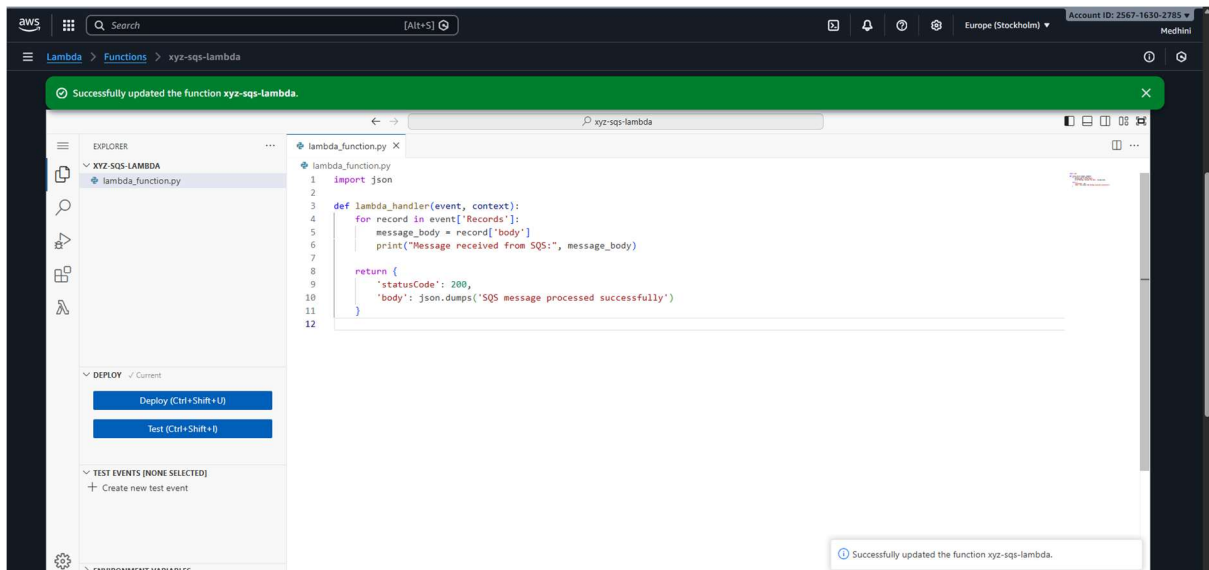
        'body': json.dumps('SQS message processed successfully')

    }
```

Click **Deploy**



## Assignment – 1 - Lambda



### STEP 5: Add SQS as Lambda Trigger

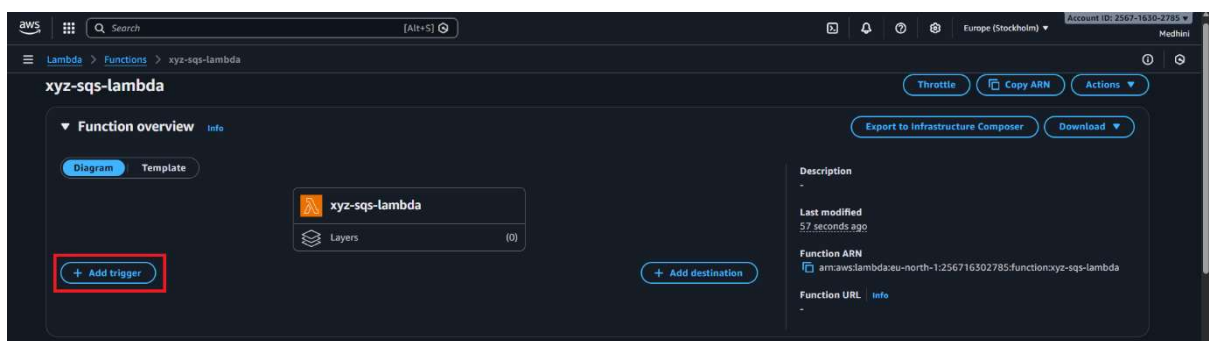
#### Add Trigger

- In Lambda → Click **Add trigger**
- Select **SQS**

#### Configure Trigger

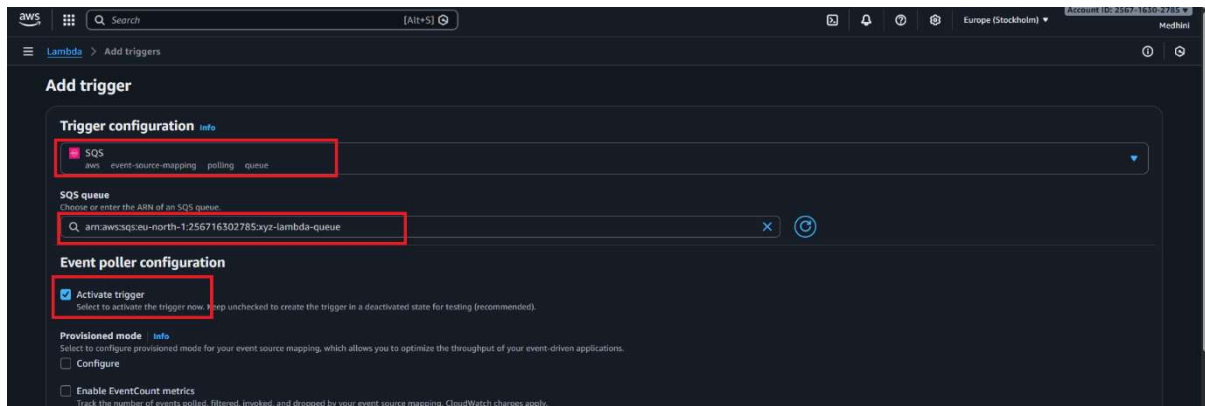
- Choose queue: xyz-lambda-queue
- Enable trigger
- Click **Add**

Now SQS is connected to Lambda

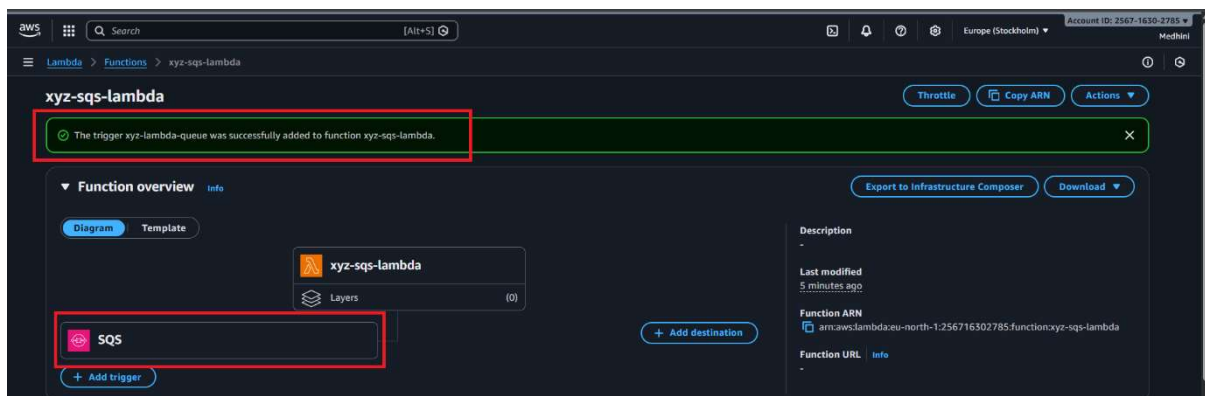




## Assignment – 1 - Lambda



Scroll down and click on “Add”



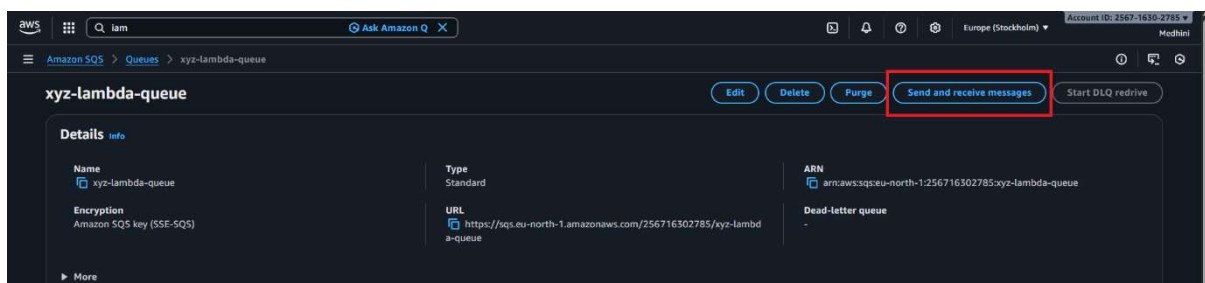
## STEP 6: Send Message to SQS (Test)

### Open SQS

- Click xyz-lambda-queue
- Click **Send and receive messages**

### Send Message

Message body example:



## Assignment – 1 - Lambda

**Send and receive messages**  
Use this page to send, retrieve and view messages, helping you experiment with various queue features.

**Send message** [info](#)

**Message body**  
Enter the message to send to the queue.

**Message group ID - optional, new** [info](#)  
A group identifier for the message to allow fair processing across message groups in a standard queue.

Message group ID must be 1 to 128 characters. Valid characters are a-z, A-Z, 0-9, and punctuation ("#\$%&'()\*+,-./:;<=>?@[\]^\_`{|}~").

**Delivery delay** [info](#)  
The duration (in seconds) that SQS will postpone the initial delivery of the message. During this delay period, the message is not visible to consumers, allowing you to create a wait time before the message becomes available for processing.  
 Seconds  
Should be between 0 seconds and 15 minutes.

► **Message attributes - optional** [info](#)

[Clear content](#) [Send message](#)

**Send and receive messages**  
Use this page to send, retrieve and view messages, helping you experiment with various queue features.

**Send message** [info](#)

✔ Your message has been sent and is ready to be received. [View sent message details](#) ✕

**Message body**  
Enter the message to send to the queue.

**Message group ID - optional, new** [info](#)  
A group identifier for the message to allow fair processing across message groups in a standard queue.

Message group ID must be 1 to 128 characters. Valid characters are a-z, A-Z, 0-9, and punctuation ("#\$%&'()\*+,-./:;<=>?@[\]^\_`{|}~").

**Delivery delay** [info](#)  
The duration (in seconds) that SQS will postpone the initial delivery of the message. During this delay period, the message is not visible to consumers, allowing you to create a wait time before the message becomes available for processing.  
 Seconds  
Should be between 0 seconds and 15 minutes.

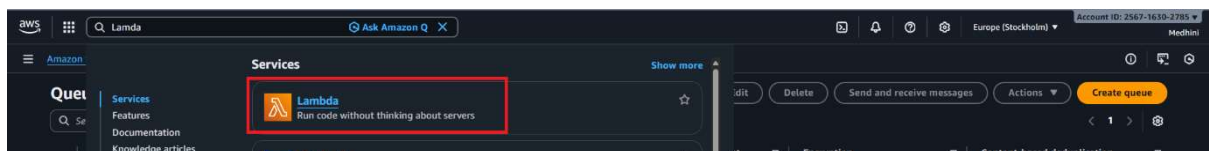
► **Message attributes - optional** [info](#)

[Clear content](#) [Send message](#)

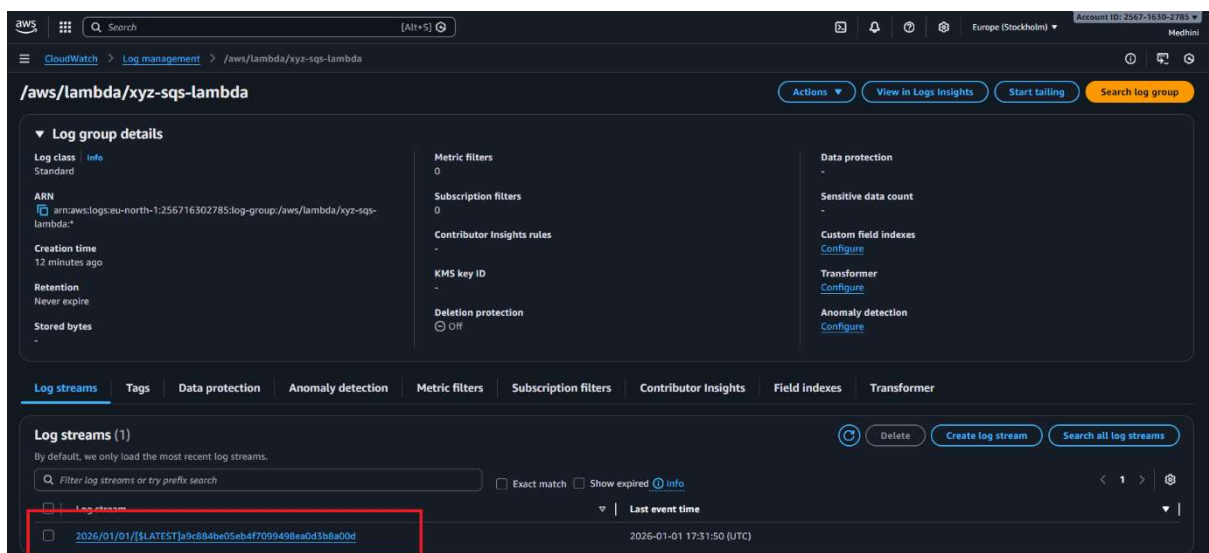
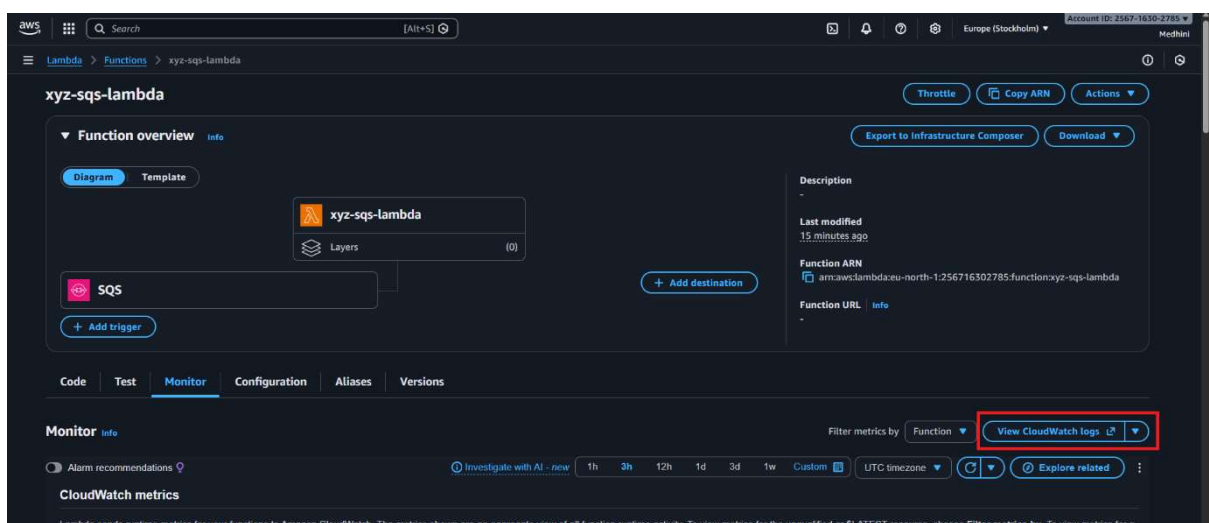
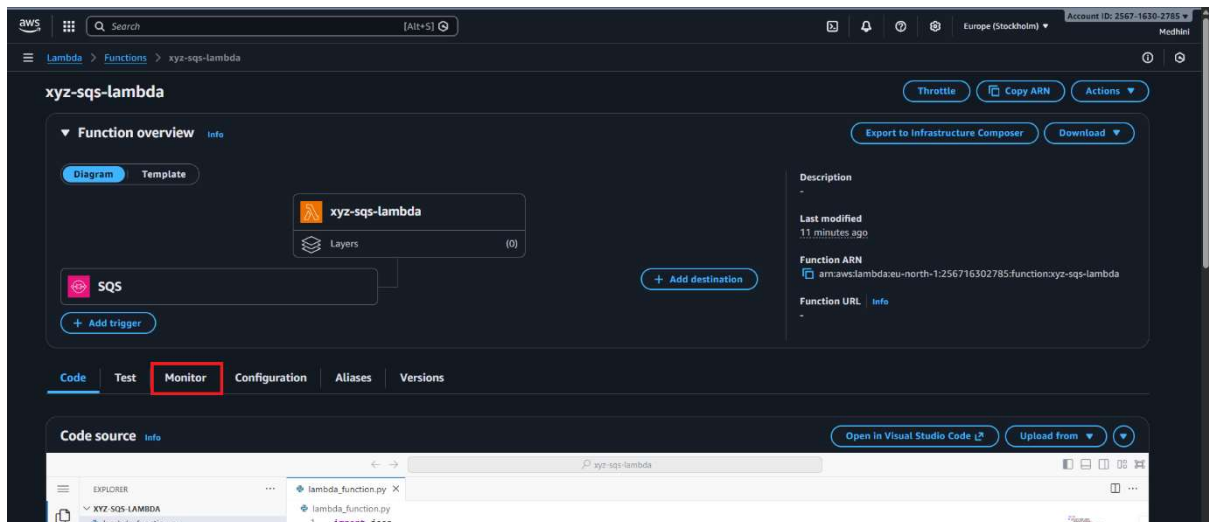
## STEP 7: Verify Lambda Execution

### Go to Lambda

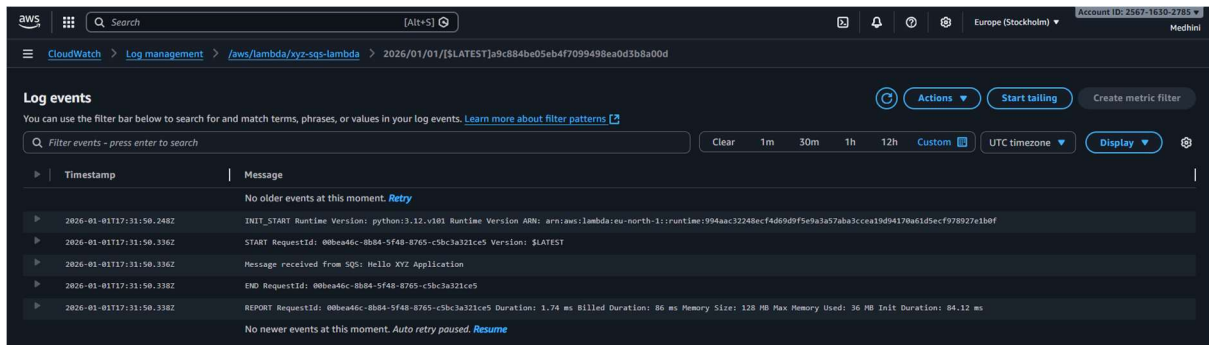
- Open xyz-sqs-lambda
- Click **Monitor**
- Click **View logs in CloudWatch**



## Assignment – 1 - Lambda



## Assignment – 1 - Lambda



The screenshot shows the AWS CloudWatch console interface for a Lambda function named 'xyz-sqs-lambda'. The breadcrumb navigation indicates the path: CloudWatch > Log management > /aws/lambda/xyz-sqs-lambda > 2026/01/01/[SLATEST]a9c884be05eb4f7099498ea0d3b8a00d. The 'Log events' section is active, displaying a list of log events. The interface includes a search bar, filter options (Clear, 1m, 30m, 1h, 12h, Custom), a 'Start tailing' button, and a 'Display' button. The log events are as follows:

Timestamp	Message
No older events at this moment. <a href="#">Retry</a>	
2026-01-01T17:31:50.248Z	INIT_START Runtime Version: python:3.12.v181 Runtime Version ARN: arn:aws:lambda:eu-north-1:runtime:994aac32248ecf4d69d9f5e9a3a57aba3ccea19d94178a61d5ecf978927e3b0f
2026-01-01T17:31:50.336Z	START RequestId: 00bea46c-8b84-5f48-8765-c5bc3a321ce5 Version: \$LATEST
2026-01-01T17:31:50.336Z	Message received from SQS: Hello XYZ Application
2026-01-01T17:31:50.338Z	END RequestId: 00bea46c-8b84-5f48-8765-c5bc3a321ce5
2026-01-01T17:31:50.338Z	REPORT RequestId: 00bea46c-8b84-5f48-8765-c5bc3a321ce5 Duration: 1.74 ms Billed Duration: 86 ms Memory Size: 128 MB Max Memory Used: 36 MB Init Duration: 84.12 ms
No newer events at this moment. <a href="#">Auto retry paused</a> . <a href="#">Resume</a>	

Lambda triggered automatically