

# Module 9: Lambda Assignment

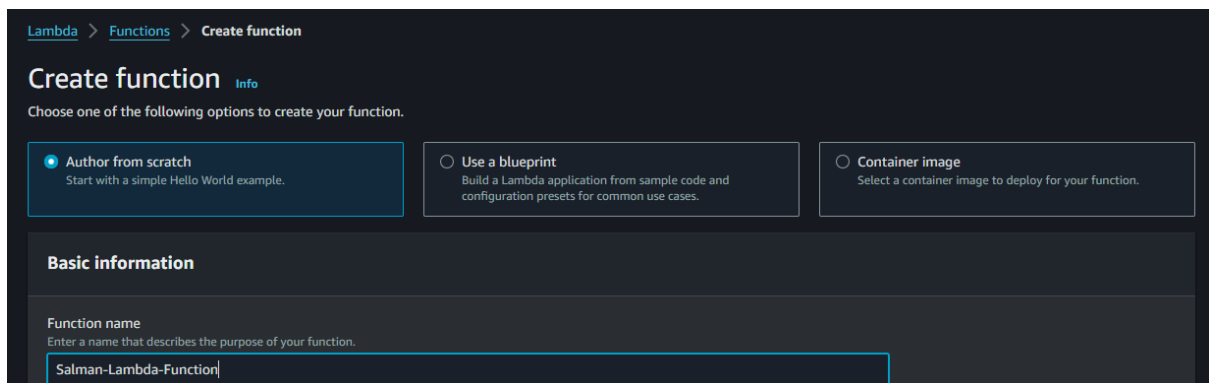
## Problem Statement:

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:

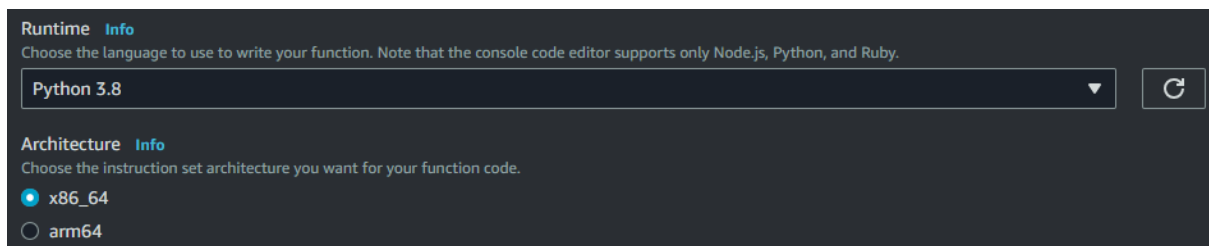
1. Create a sample Python Lambda function.
2. Set the Lambda Trigger as SQS and send a message to test invocations.

1. Go to **Lambda Service** and **Create Function**



The screenshot shows the 'Create function' page in the AWS Lambda console. The breadcrumb trail at the top is 'Lambda > Functions > Create function'. The main heading is 'Create function' with an 'Info' link. Below the heading, it says 'Choose one of the following options to create your function.' There are three radio button options: 'Author from scratch' (selected), 'Use a blueprint', and 'Container image'. The 'Author from scratch' option has a subtext 'Start with a simple Hello World example.' Below these options is a section titled 'Basic information'. Under 'Basic information', there is a 'Function name' label and a text input field containing 'Salman-Lambda-Function'.

2. Select **Runtime Version** and **Architecture**



The screenshot shows the 'Runtime' and 'Architecture' configuration section of the AWS Lambda console. The 'Runtime' section has a label 'Runtime' and an 'Info' link. It says 'Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.' Below this is a dropdown menu showing 'Python 3.8' and a refresh button. The 'Architecture' section has a label 'Architecture' and an 'Info' link. It says 'Choose the instruction set architecture you want for your function code.' Below this are two radio button options: 'x86\_64' (selected) and 'arm64'.

3. Before **Selecting Role** So we create a **Role for Lambda** and **GO to IAM** > Services **Lambda**

The screenshot shows the 'Create role' wizard in the AWS IAM console. The breadcrumb navigation is 'IAM > Roles > Create role'. The left sidebar shows the progress: Step 1 'Select trusted entity' is completed, Step 2 'Add permissions' is next, and Step 3 'Name, review, and create' is the current step. The main heading is 'Name, review, and create'. Under 'Role details', the 'Role name' field contains 'Salman-Lambda-Role' and the 'Description' field contains 'Allows Lambda functions to call AWS services on your behalf.'

4. Example I **will Allow Permissions S3** and **Cloudwatch Full Access** and Hit the Button **Created**

The screenshot shows the 'Step 2: Add permissions' screen. It includes a 'Permissions policy summary' table with two entries: 'AmazonS3FullAccess' and 'CloudWatchFullAccess', both of type 'AWS managed' and attached as 'Permissions policy'. Below this is the 'Step 3: Add tags' section, which states 'No tags associated with the resource.' and has an 'Add new tag' button. At the bottom right are 'Cancel', 'Previous', and 'Create role' buttons.

Policy name	Type	Attached as
<a href="#">AmazonS3FullAccess</a>	AWS managed	Permissions policy
<a href="#">CloudWatchFullAccess</a>	AWS managed	Permissions policy

5. Role Created **Successfully**

The screenshot shows the 'Roles (8)' page in the AWS IAM console. The left sidebar shows the navigation menu with 'Roles' selected. The main area has a search bar and a table listing the roles. The 'Salman-Lambda-Role' is highlighted.

Role name	Trusted entities	Last activity
<a href="#">AWSServiceRoleForSupport</a>	AWS Service: support (Service-Linker)	20 days ago
<a href="#">AWSServiceRoleForTrustedAdvisor</a>	AWS Service: trustedadvisor (Service-Linker)	-
<a href="#">Salman-Lambda-Role</a>	AWS Service: lambda	-

## 6. And Now Go to Lambda Function Now Select **Existing Role** which we Created

**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

**Existing role**  
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

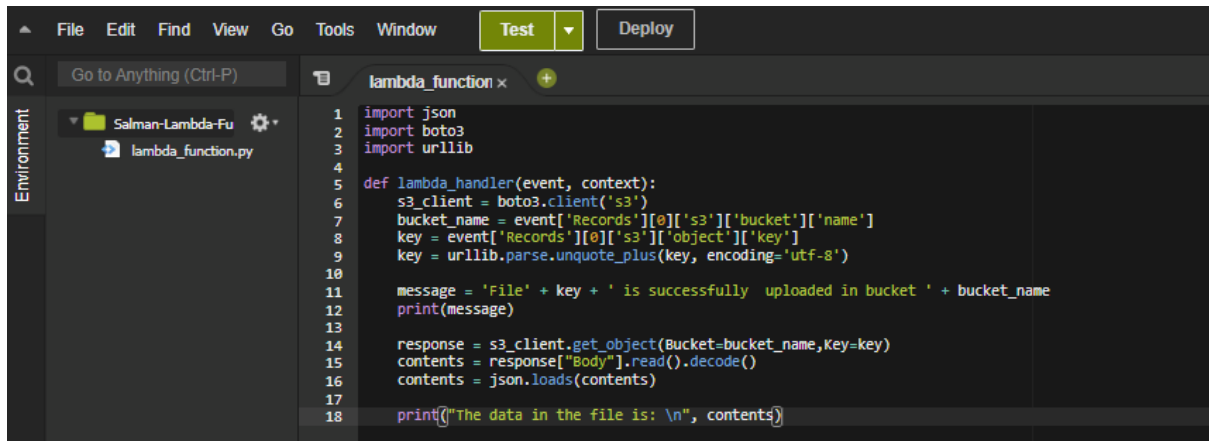
Salman-Lambda-Role ▼ 

[View the Salman-Lambda-Role role](#) on the IAM console.

► **Advanced settings**

Cancel **Create function**

## 7. Deploy the Python **Program and Test it**



The screenshot shows an IDE with a menu bar (File, Edit, Find, View, Go, Tools, Window) and buttons for 'Test' and 'Deploy'. The left sidebar shows the 'Environment' with a project named 'Salman-Lambda-Fu' and a file 'lambda\_function.py'. The main editor window shows the following Python code:


```
1 import json
2 import boto3
3 import urllib
4
5 def lambda_handler(event, context):
6     s3_client = boto3.client('s3')
7     bucket_name = event['Records'][0]['s3']['bucket']['name']
8     key = event['Records'][0]['s3']['object']['key']
9     key = urllib.parse.unquote_plus(key, encoding='utf-8')
10
11     message = 'File' + key + ' is successfully uploaded in bucket ' + bucket_name
12     print(message)
13
14     response = s3_client.get_object(Bucket=bucket_name, Key=key)
15     contents = response["Body"].read().decode()
16     contents = json.loads(contents)
17
18     print("The data in the file is: \n", contents)
```

8. And Need to be **Add the Trigger**(We Here Select S3 as a Trigger) and **Select Here Your S3 Bucket** and before that need to create a Bucket which **I already created**

Lambda > Add trigger

## Add trigger

**Trigger configuration** Info

 **S3**  
aws asynchronous storage

**Bucket**  
Choose or enter the ARN of an S3 bucket that serves as the event source. The bucket must be in the same region as the function.

Q s3/salman-mar28 X ↻

Bucket region: us-east-1

**Event types**  
Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.

▼

All object create events X

9. **Acknowledge** and Add

☒ I acknowledge that using the same S3 bucket for both input and output is not recommended and that this configuration can cause recursive invocations, increased Lambda usage, and increased costs.

Lambda will add the necessary permissions for AWS S3 to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.

Cancel Add

10. Go **to S3 selected Bucket** Make sure u have to create a Bucket **which u selected region for Lambda** and Upload a Json File which we written in the **code Allow Messages to S3** and the **Destination is SQS**

Objects Properties Permissions Metrics Management Access Points

**Objects (0)** Info

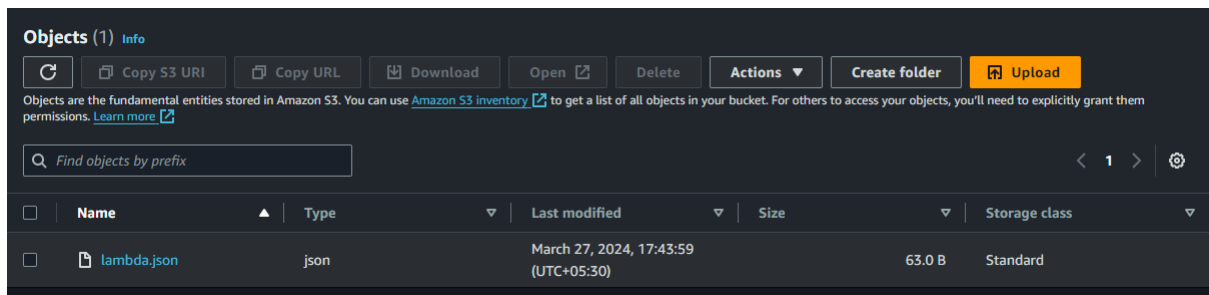
↻ Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

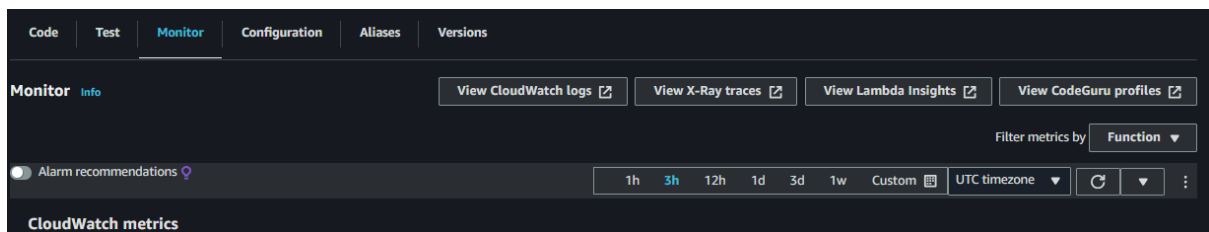
Find objects by prefix

Name	Type	Last modified	Size	Storage class
No objects				
You don't have any objects in this bucket.				
Upload				

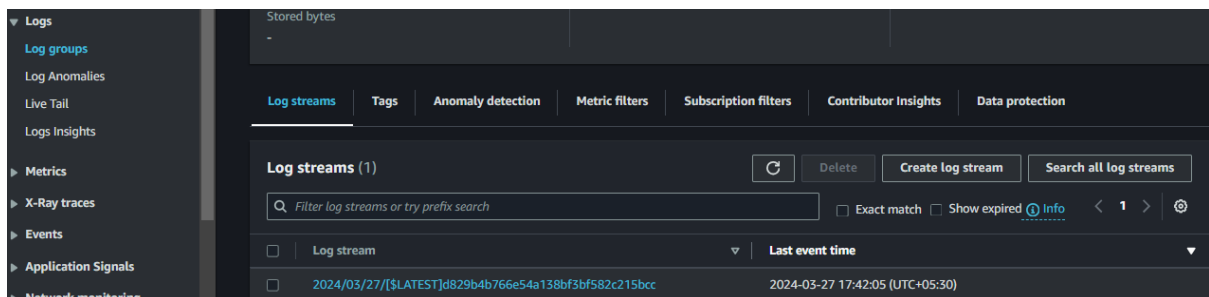
## 11. Uploaded Successfully



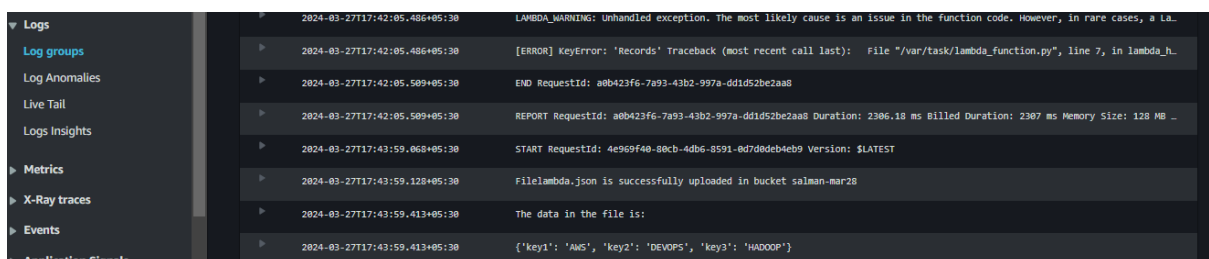
## 12. In Lambda we find a Monitor Option Click On it and it will open different tab and go Cloudwatch Dashboard and We Find Logs Groups



## 13. We Can see Log Stream



## 14. The messages Passes Successfully in cloudwatch and we can see in Logs which running and also we Allowed Permissions to Cloudwatch for Lambda



15. So Now our Destination is Pass the Message to SQS so Need to Attach SQS Permissions to Lambda Role

**Salman-Lambda-Role** [Info](#) [Delete](#)

Allows Lambda functions to call AWS services on your behalf.

---

**Summary** [Edit](#)

Creation date March 27, 2024, 17:39 (UTC+05:30)	ARN arn:aws:iam::211125783778:role/Salman-Lambda-Role
Last activity -	Maximum session duration 1 hour

---

[Permissions](#) | [Trust relationships](#) | [Tags](#) | [Access Advisor](#) | [Revoke sessions](#)

---

**Permissions policies (2)** [Info](#) [Refresh](#) [Simulate](#) [Remove](#) [Add permissions](#)

You can attach up to 10 managed policies.

Filter by Type [All types](#)

<input type="checkbox"/>	<a href="#">Policy name</a>	Type	Attached entities
<input type="checkbox"/>	<a href="#">AmazonS3FullAccess</a>	AWS managed	1
<input type="checkbox"/>	<a href="#">CloudWatchFullAccess</a>	AWS managed	1

16. See We Attached SQS Full Access Permission

<input type="checkbox"/>	<a href="#">Policy name</a>	Type	Attached entities
<input type="checkbox"/>	<a href="#">AmazonS3FullAccess</a>	AWS managed	1
<input type="checkbox"/>	<a href="#">AmazonSQSFullAccess</a>	AWS managed	1
<input type="checkbox"/>	<a href="#">CloudWatchFullAccess</a>	AWS managed	1

17. Go To SQS Create a Queue

[Amazon SQS](#) > [Queues](#) > [Create queue](#)

## 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

18. See Currently the **Poll Messages is Zero**

Receive messages <small>Info</small>			
<a>Edit poll settings</a> <a>Stop polling</a> <a>Poll for messages</a>			
Messages available	Polling duration	Maximum message count	Polling progress
0	30	10	 0 receives/second

19. Go to Lambda Function Click **Destination and Select Destination Type** and Create it

Lambda > Functions > Salman-Lambda-Function > **Add destination**

## Add destination

**Destination configuration** Info

Configure a destination to receive invocation records. Lambda can send records when your function is invoked asynchronously, or when your function processes records from an event source mapping.

**Source**  
Choose the invocation type that Lambda sends records for.

☒ Asynchronous invocation

☐ Event source mapping invocation

**Condition**  
Choose whether to send invocation records for event processing failures or for successful invocations.


☐ On failure

☒ On success

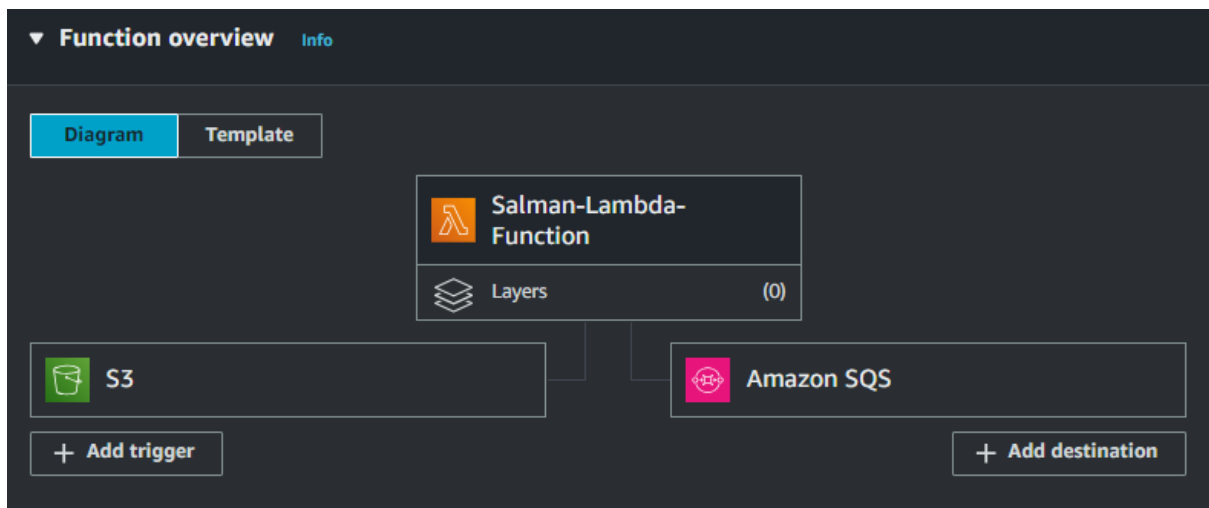
**Destination type**  
Choose the destination type that Lambda sends invocation records to.

SQS queue ▼

**Destination**  
Choose the ARN of the destination, or enter the ARN manually.

🔍  ✕ 

20. **Destination Also Added.**



21. Once Again **Re-upload the Json Doc** to Trigger from s3 and Passes the Message to SQS

Objects (1) Info

Buttons: Refresh, Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, Upload

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	lambda.json	json	March 27, 2024, 17:43:59 (UTC+05:30)	63.0 B	Standard

22. Reupload **Successfully**

Objects (1) Info

Buttons: Refresh, Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, Upload

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	lambda.json	json	March 27, 2024, 18:00:02 (UTC+05:30)	63.0 B	Standard

23. Now to **SQS Click Poll for Messages** and it shows **one Message** which is triggered from S3

Receive messages Info

Buttons: Edit poll settings, Stop polling, Poll for messages

Messages available: 0

Polling duration: 30

Maximum message count: 10

Polling progress: 0.5 receives/second

Messages (1)

Search messages

	ID	Sent	Size	Receive count
<input type="checkbox"/>	d02be2b3-d1e4-4ad9-a2ba-58795887fd6a	2024-03-27T18:00:05:30	1.09 KB	1



#### 24. Message Received.

Message Triggered from S3 and Particularly we written the code for Allowing only Json File and It Passes the Message to Destination and We Add Permissions to Cloudwatch So Logs Also Generated

