# **Lambda Functions – Event Announcement System**

## **Overview**

In this part of the Event Announcement System, I implemented two key AWS Lambda functions that form the backbone of backend automation:

1. Subscription Lambda Function – to handle email subscriptions via SNS
2. Event Creation Lambda Function – to update event records in S3 and notify users via SNS

Both functions are integrated with the API Gateway and configured with IAM roles to access AWS services, such as SNS and S3, securely. Below, I have documented the steps I followed to build, test, and deploy both Lambda functions.

## **Create and Test the Subscription Lambda Function**

This function subscribes users to an SNS topic using the email address entered in the frontend form.

**Step 1: Create IAM Role**

I created a new IAM role called LambdaSubscribeRole with the following AWS-managed policies:

* AmazonSNSFullAccess
* AWSLambdaBasicExecutionRole

This role was attached to the Lambda function to grant necessary permissions.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 1: Creating an IAM role with required SNS and logging permissions for the subscription function.

**Step 2: Create the Lambda Function**

I created a new Lambda function named SubscribeToSNSFunction using Python 3.12. This function listens for HTTP POST requests containing the user's email, then subscribes that email to the SNS topic.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 2: Configuring the Subscribe Lambda function with Python 3.12 runtime.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 3: Creating the SubscribeToSNSFunction in the AWS Lambda console.

**Function Code:**

The full source code for the Subscription Lambda Function is available in the project folder. To view the complete implementation, click the link below:

[Open Subscription Lambda Function Code (subscribe function)](https://d.docs.live.net/f0b4b282846cba86/Desktop/AWS%20Projects/Event-announcement-system/lambda/subscribe_function.py)

**Step 3: Test the Function**

To verify, I tested the function with the following event:

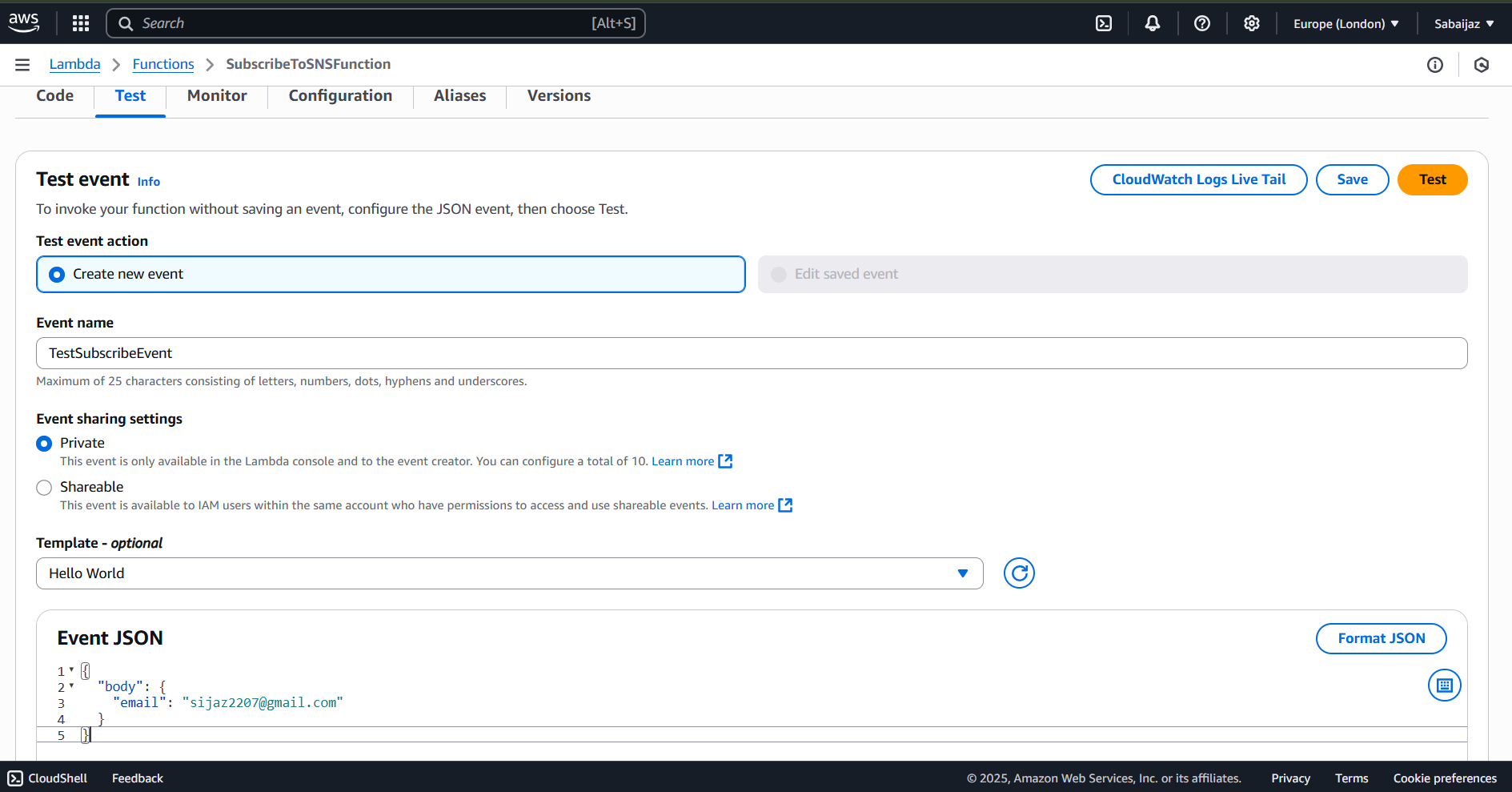


Figure 4: Testing the subscription Lambda function in the AWS console.

After executing the test, a confirmation email was sent. Upon confirmation, the email was added to the SNS topic.

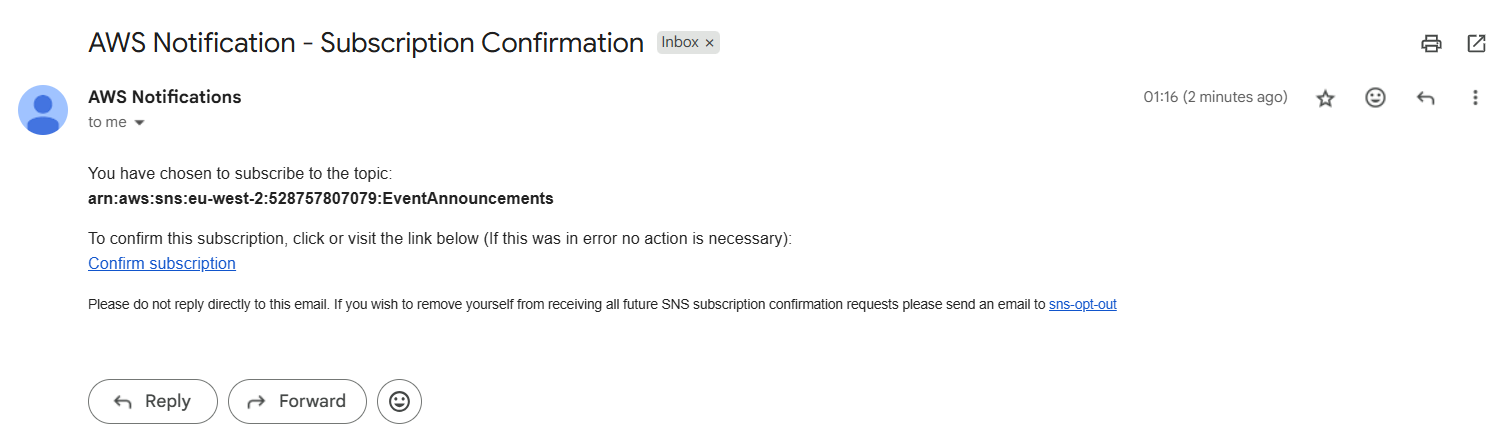


Figure 5: Confirmation email sent and subscription added to SNS topic after user confirmation.

**Create and Test the Event Creation Lambda Function**

This function updates the events.json file in S3 with new events and notifies subscribers via SNS.

**Step 1: Create IAM Role**

I created an IAM role named EventCreationLambdaRole and attached:

* AmazonS3FullAccess
* AmazonSNSFullAccess

This role allows the Lambda to read/write from S3 and publish to the SNS topic.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 6: IAM role creation for EventCreationLambdaRole with S3 and SNS permissions.

**Step 2: Create the Lambda Function**

I created a Lambda function named createEventFunction using Python 3.12 and attached the above IAM role. This function is triggered via API Gateway when users submit an event through the form.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 7: Configuring the create event Lambda function with Python 3.12 runtime.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 8: Setting up and creating createEventFunction in the AWS Lambda console.

**Function code:**

The full source code for the Event Creation Lambda Function is included in the project folder. To review the code, click the link below:

[Open Event Creation Lambda Function Code (createEventFunction)](https://d.docs.live.net/f0b4b282846cba86/Desktop/AWS%20Projects/Event-announcement-system/lambda/create_event_function.py)

**Step 3: Test the Function**

Test input used in Lambda console:

A screenshot of a computer

AI-generated content may be incorrect.

Figure 9: Testing the create event Lambda function.

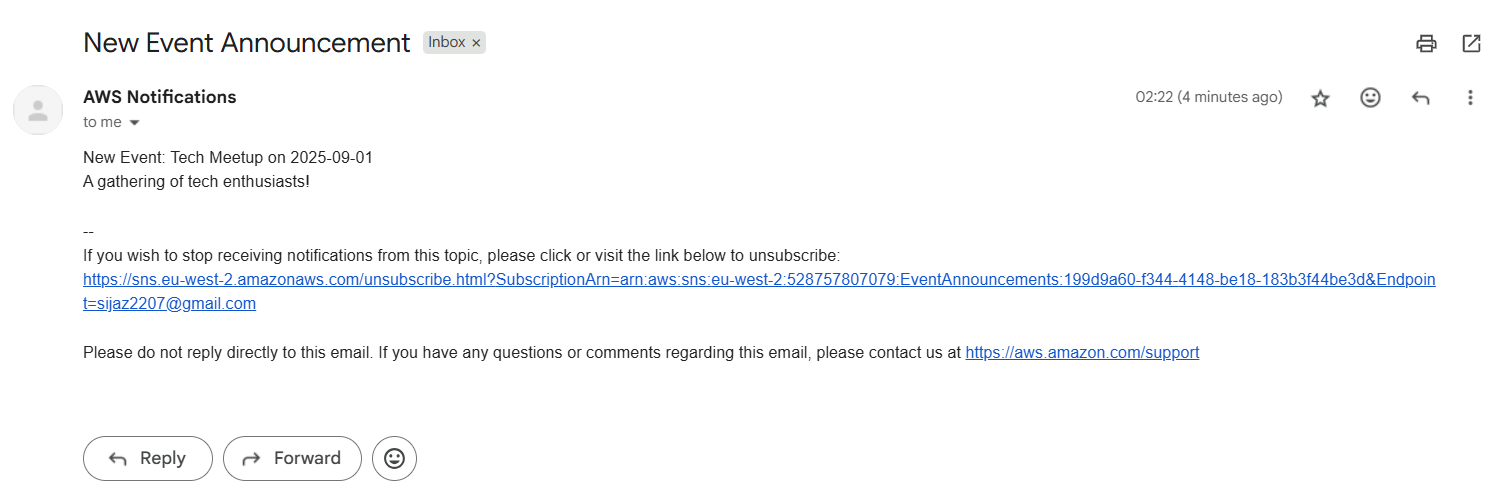


Figure 10: Email confirmation received for new event announcement subscription.