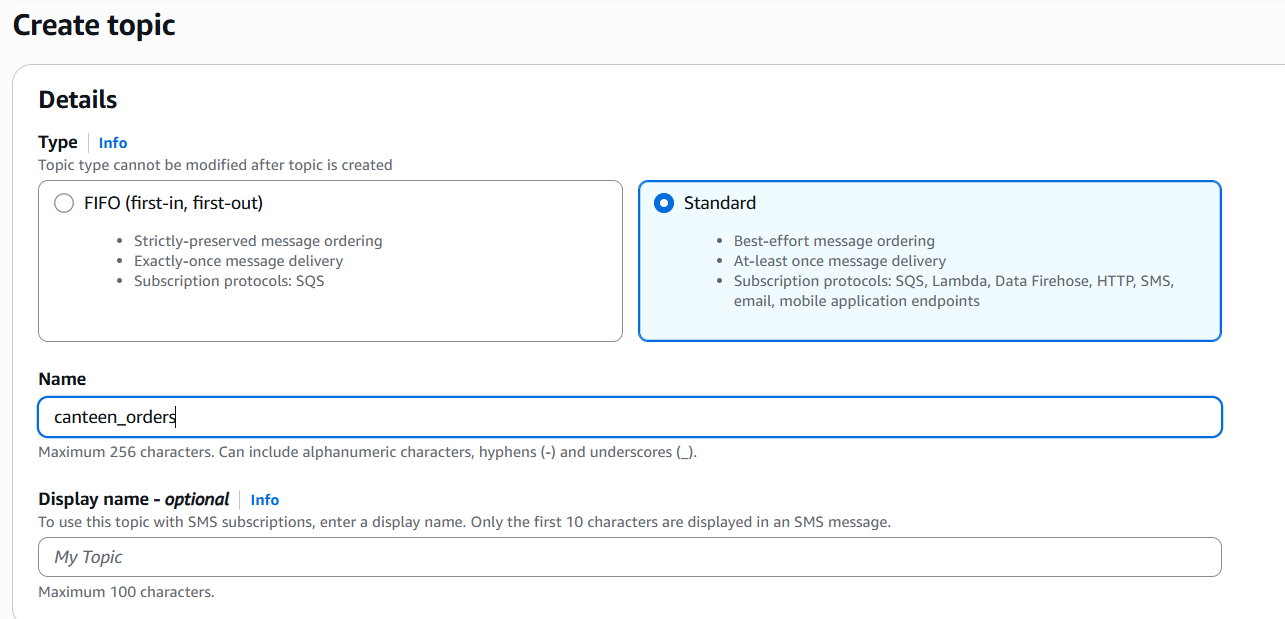
**Lab Experiment 10:SNS (Simple Notification Service)**

**Step 1: Create an SNS Topic**

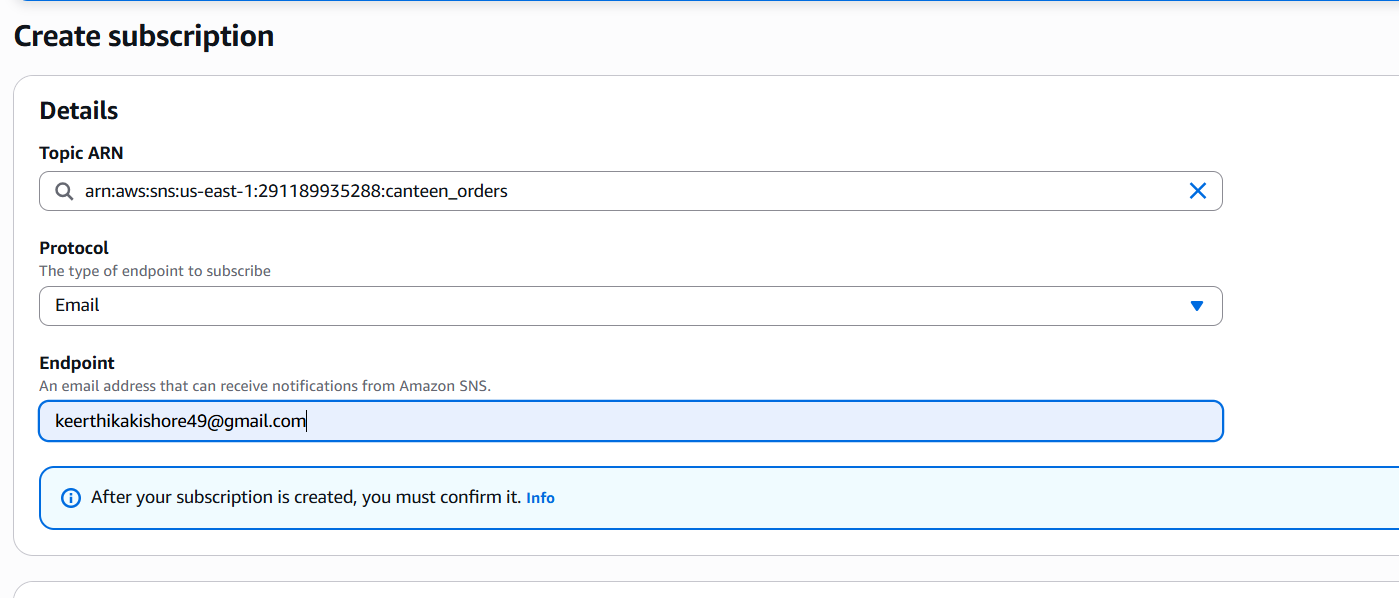
* Go to **AWS SNS** Console.
* Click on **Topics** → **Create topic**.
* Choose:
  + **Type**: Standard
  + **Name**: Orders
* Click **Create topic**.



**Fig-1: Create SNS Topic**

**Step 2: Create Subscriptions (Customer as Consumer)**

* In the Orders topic, click **Create subscription**.
* For **Protocol**, choose Email or SMS (e.g., Email for simulation).
* For **Endpoint**, enter the customer’s email address (e.g., customer@example.com).
* Click **Create subscription**.
* Go to the customer email inbox and **confirm the subscription**.



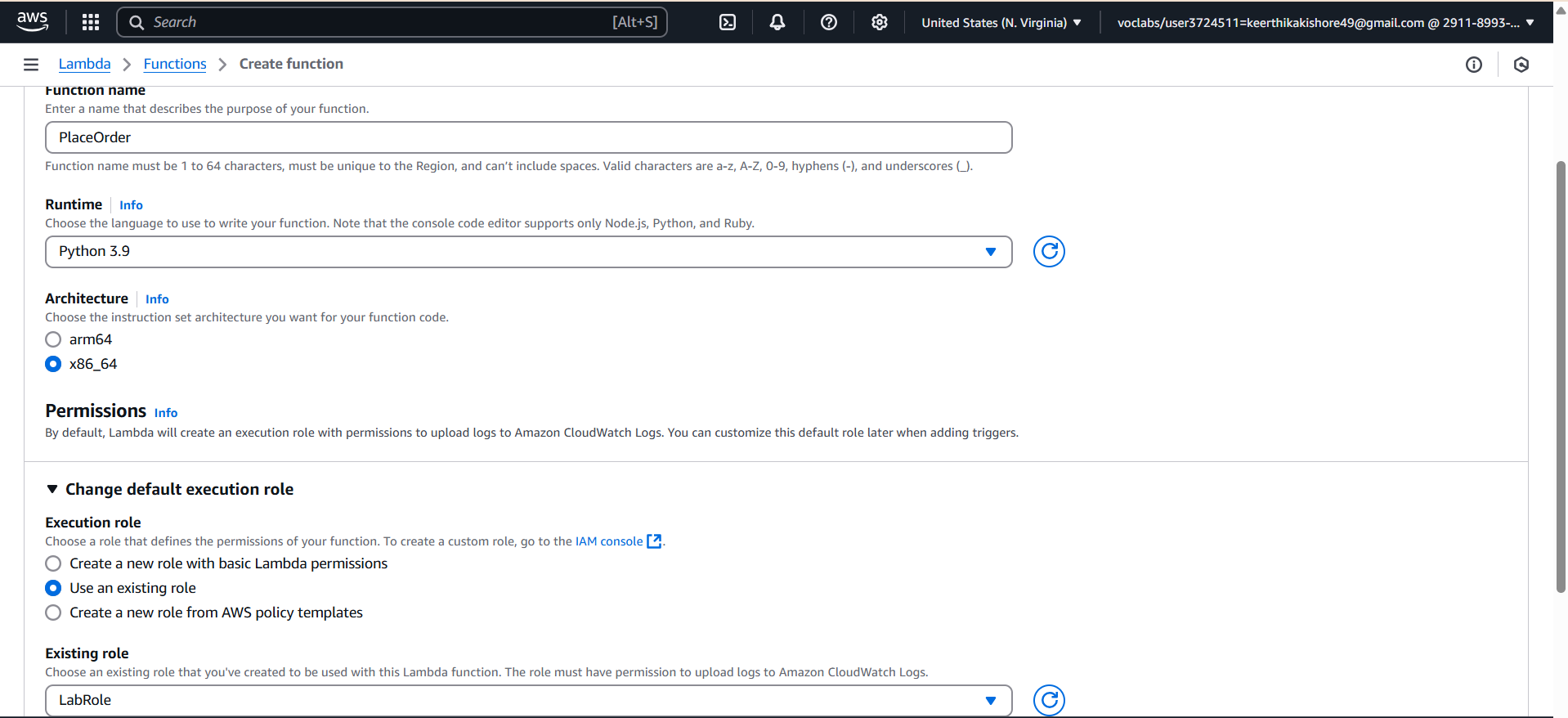
**Fig-2: Create Subscription**

**Step 3: Create AWS Lambda Function**

This Lambda will simulate the full order lifecycle after the customer initiates the order.

**a. Go to AWS Lambda → Create function**

* Name: PlaceOrder
* Runtime: Python 3.9
* Permissions: Create new role with basic Lambda permissions



**Fig-4: Setup AWS Lambda Function**

**b. Add this code:**

***import boto3***

***import time***

***sns = boto3.client('sns')***

***TOPIC\_ARN = 'arn:aws:sns:REGION:ACCOUNT\_ID:Orders' # Replace with actual ARN***

***# Ordered messages with delay (in seconds)***

***order\_flow = [***

***("Order Placed", "Order placed successfully. Thank you for ordering from our canteen!", 2),***

***("Preparing", "Your food is being freshly prepared by our kitchen team.", 5),***

***("Ready", "Your food is ready for pickup. Our delivery agent will arrive soon.", 5),***

***("Out for Delivery", "Your food is out for delivery. Please be available to receive it.", 5),***

***("Delivered", "Your food has been delivered. Enjoy your meal!", 0),***

***]***

***def publish(subject, message):***

***sns.publish(***

***TopicArn=TOPIC\_ARN,***

***Subject=subject,***

***Message=message***

***)***

***def lambda\_handler(event, context):***

***for subject, message, delay in order\_flow:***

***publish(subject, message)***

***time.sleep(delay)***

***return {***

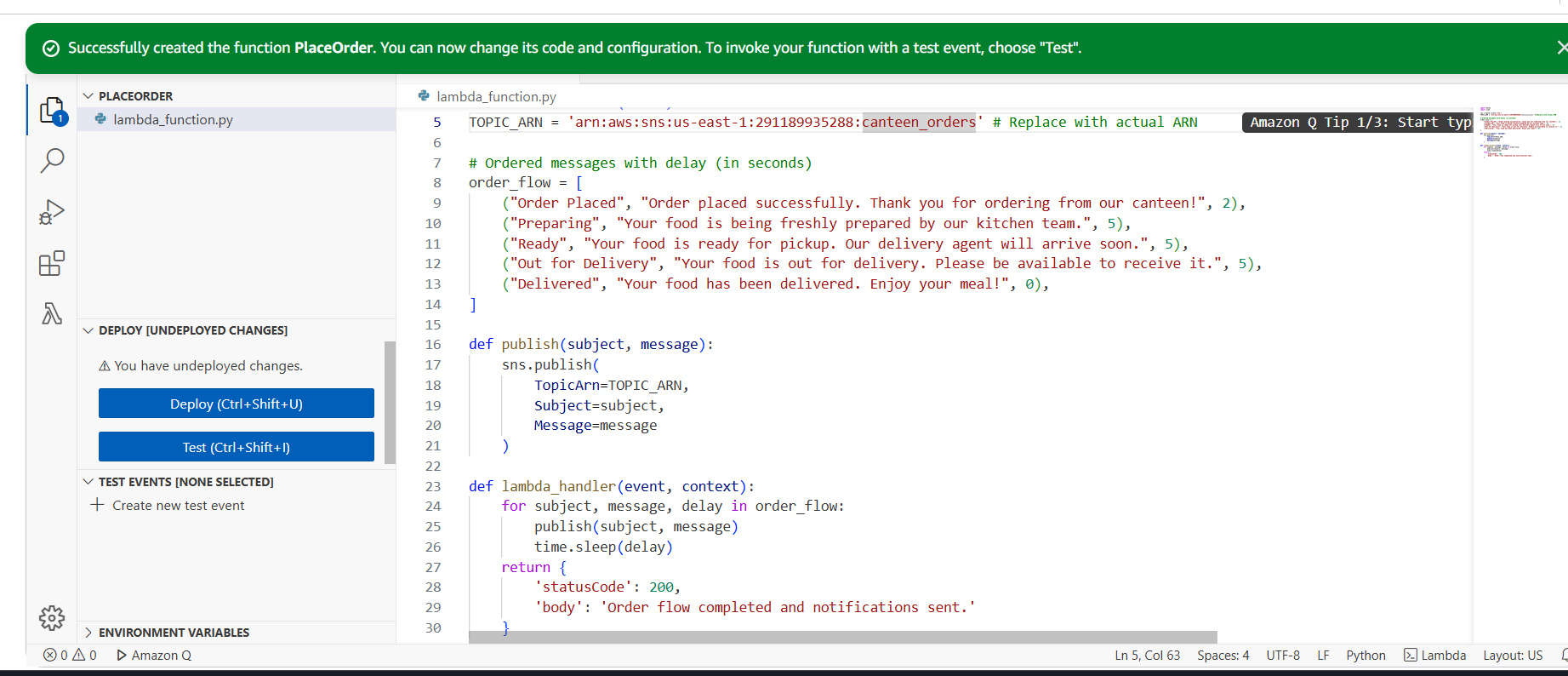
***'statusCode': 200,***

***'body': 'Order flow completed and notifications sent.'***

***}***

**Replace SNS Topic ARN**

* Replace TOPIC\_ARN = 'arn:aws:sns:REGION:ACCOUNT\_ID:Orders' with the actual **ARN** of your SNS topic.



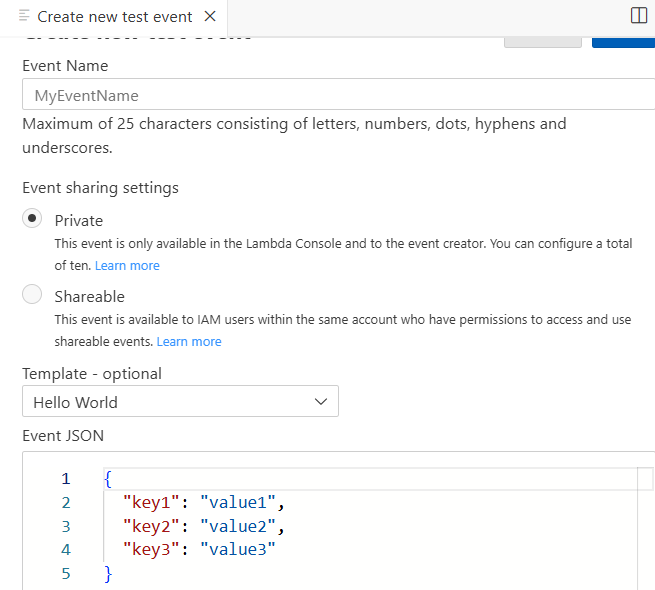
**Fig-5: Code to simulate the full order lifecycle**

**Step 4: Increase Lambda Timeout**

* Go to AWS Lambda Console.
* Open your function: PlaceOrder.
* In the Configuration tab, click on General configuration.
* Click Edit.
* Set the Timeout to 30 seconds (or more, if needed).

**Step 5: Save and Test the Lambda Function**

* After pasting the code and updating the TOPIC\_ARN, click **Deploy** to save the changes.
* **Test the Lambda** by clicking on **Test** and providing a test event (you can use an empty JSON {} as the test event).
* Once tested successfully, the function will be ready to execute automatically upon the customer placing an order.

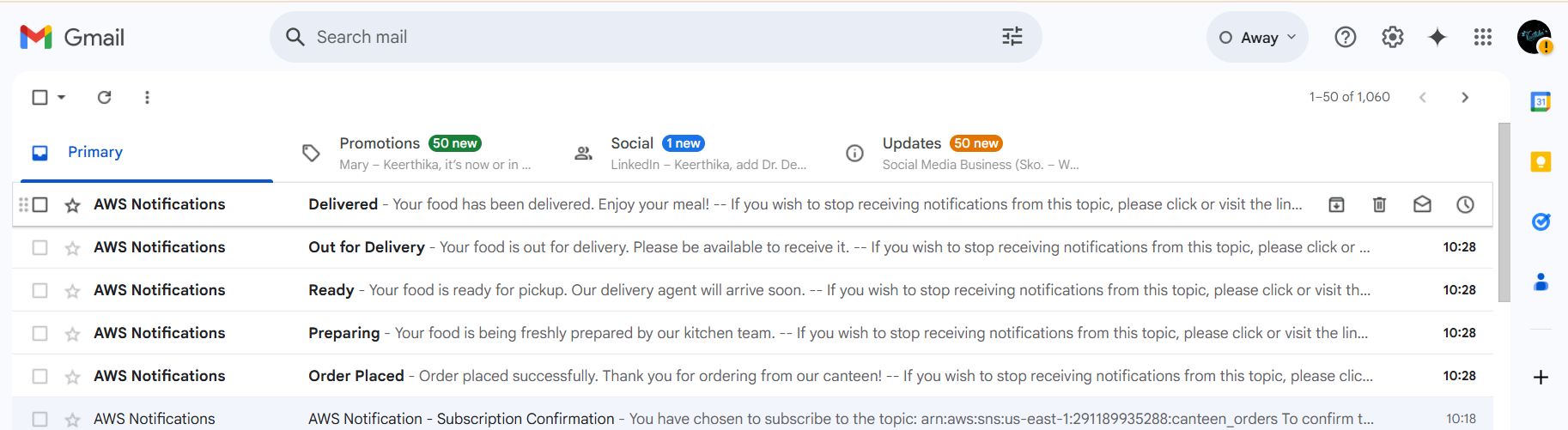


**Fig-6: Create new Test Event**

**Step 6: Customer Gets Automatic Notifications**

In their email, they will receive:

* Order placed successfully...
* After 5 sec: Your food is being prepared...
* After 5 sec: Food is ready...
* After 5 sec: Out for delivery...
* Delivered...



**Fig-7: Automatic Notifications received by consumer**