PART – C

Strategy for the Problem

- Create 3 SNS and 3 SQS architecture to ease the complexity by removing the usage of filter policy.
- Using very basic configurations even for the security to keep it simple and achieve goal.

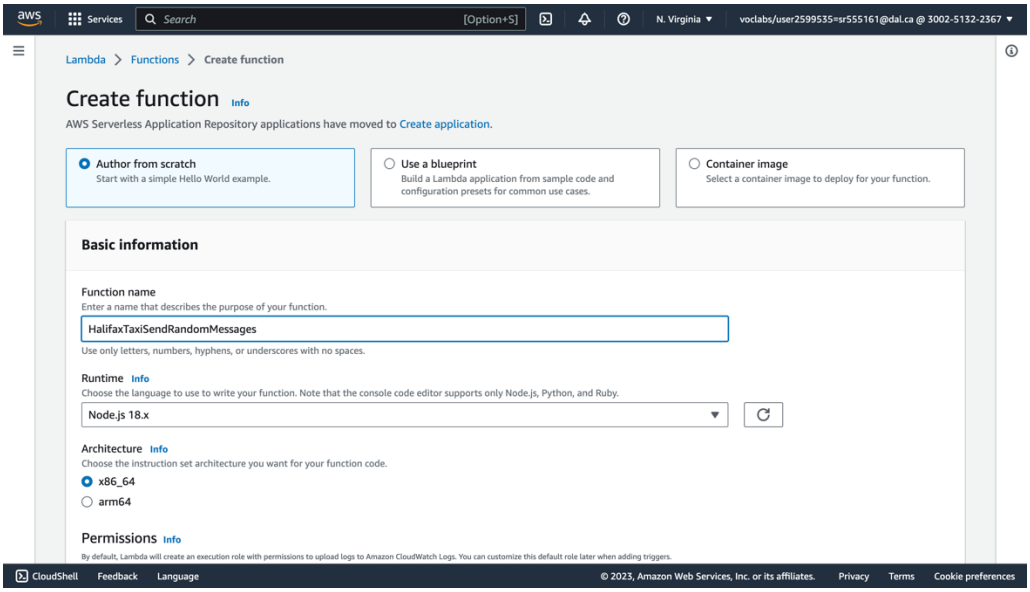

Fig. 1: Creating Lambda Function to send message to SNS.

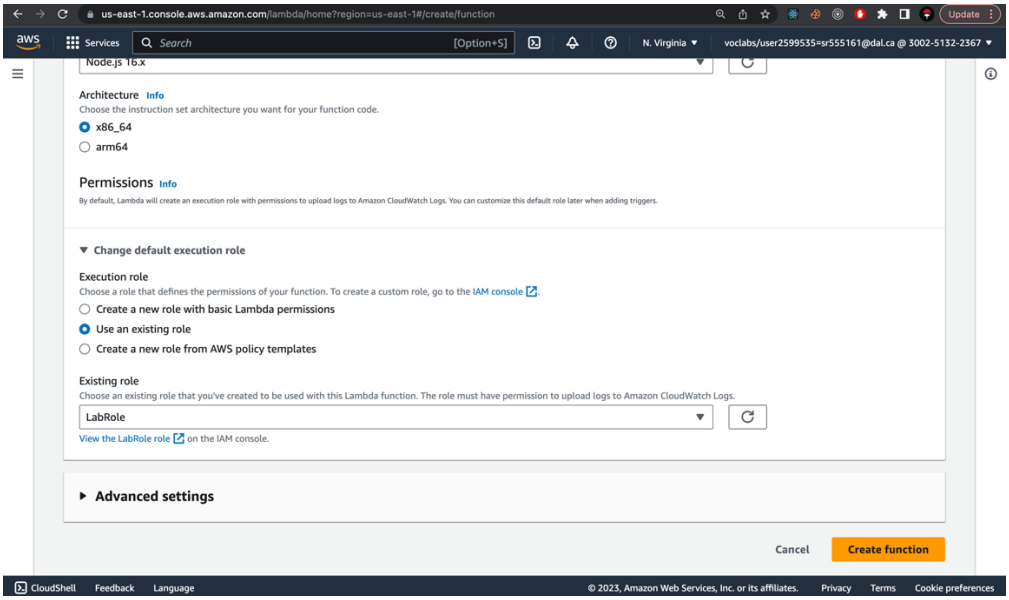

Fig. 2: Selecting LabRole as Existing Role

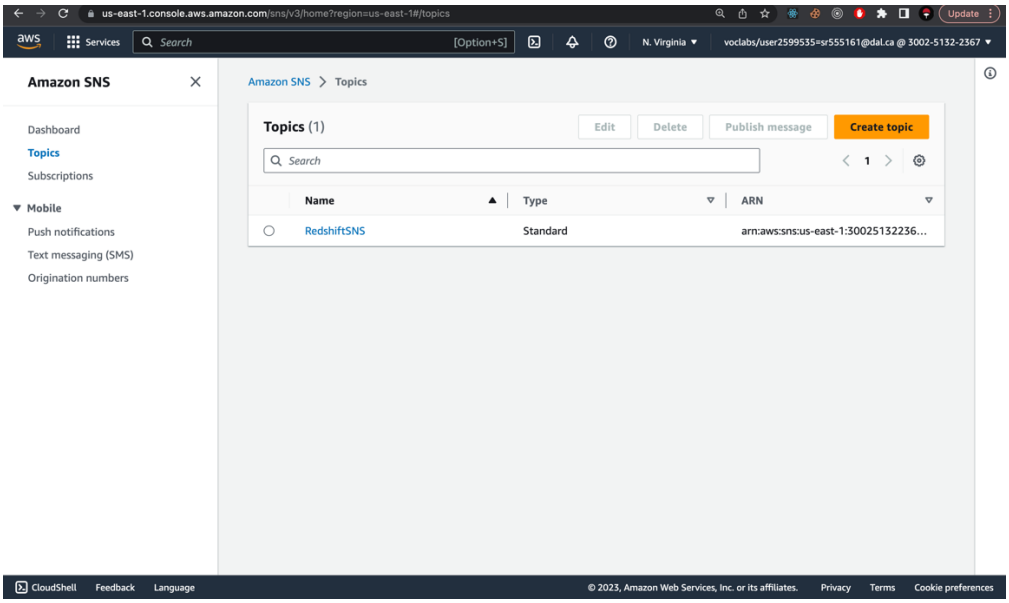

Fig. 3: SNS Dashboard

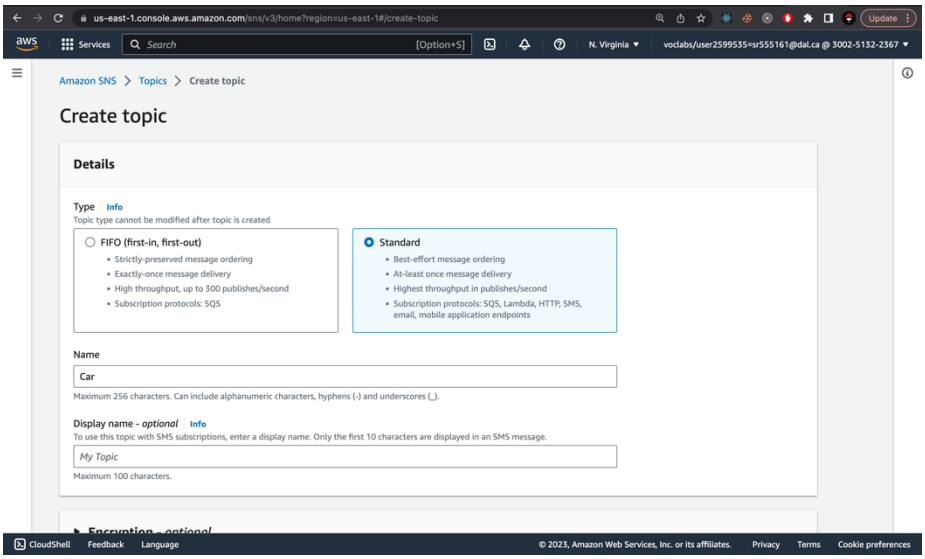

Fig. 4: Creating Standard SNS for Car

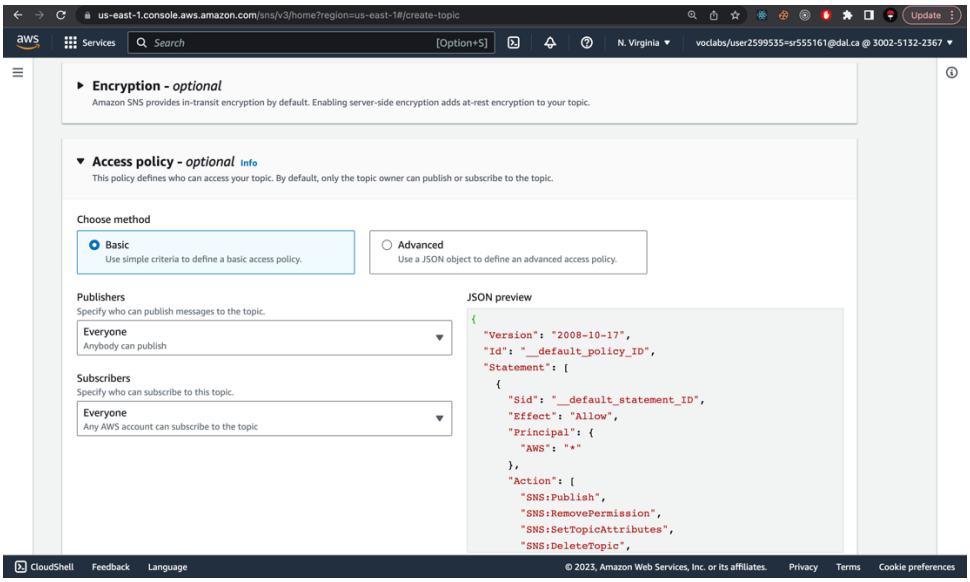

Fig. 5: Using very basic Access Policy for Car SNS.

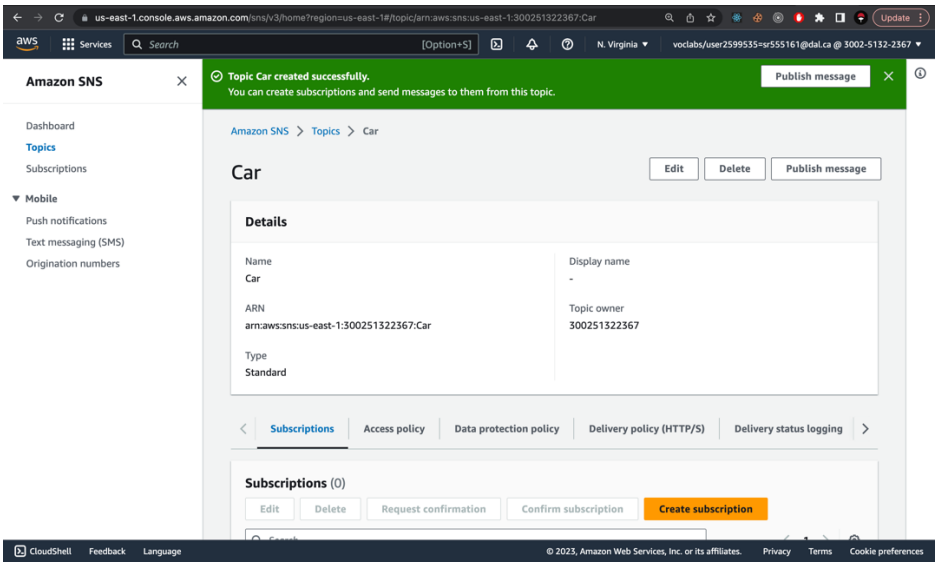

Fig. 6: SNS Car Created.

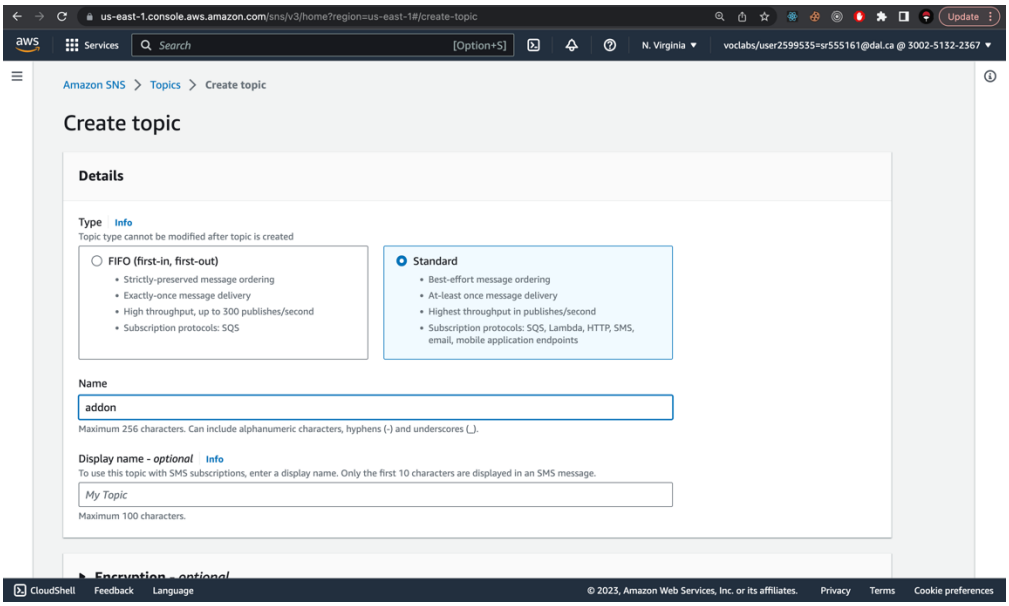

Fig. 7: Creating Standard SNS for addon

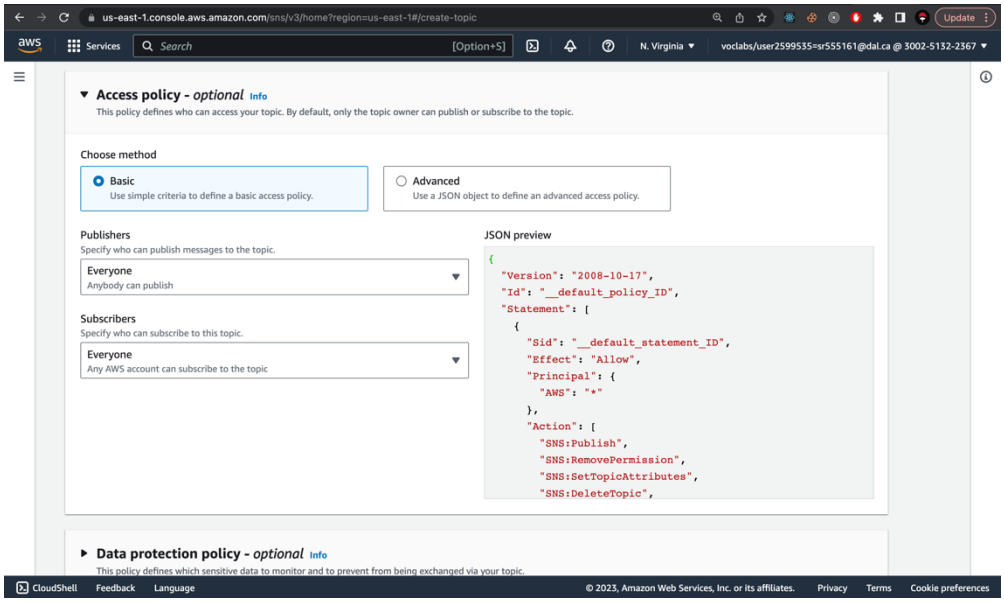

Fig. 8: Creating basic access policy for addon SNS.

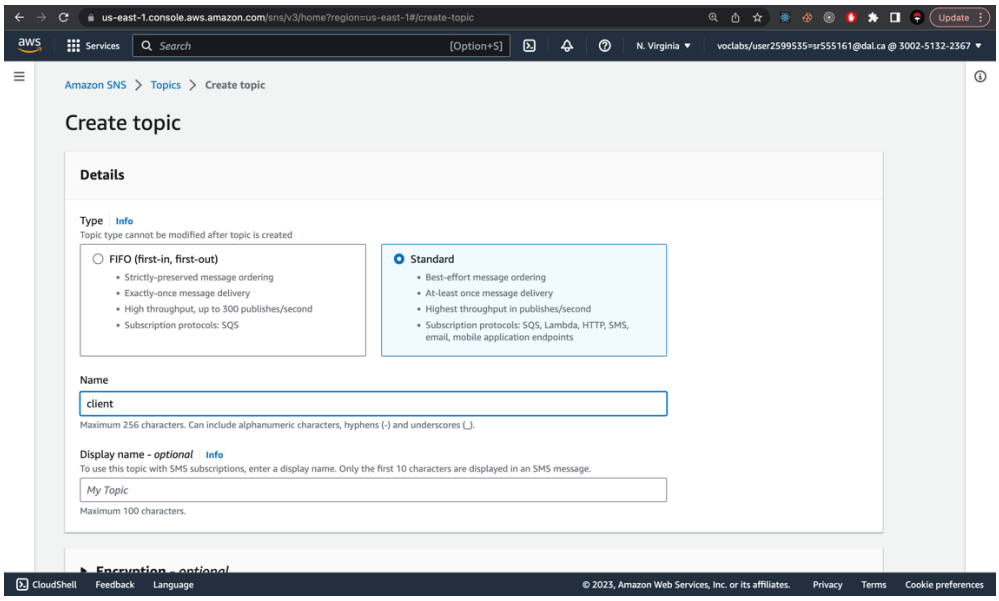

Fig. 9: Creating Standard SNS for client.

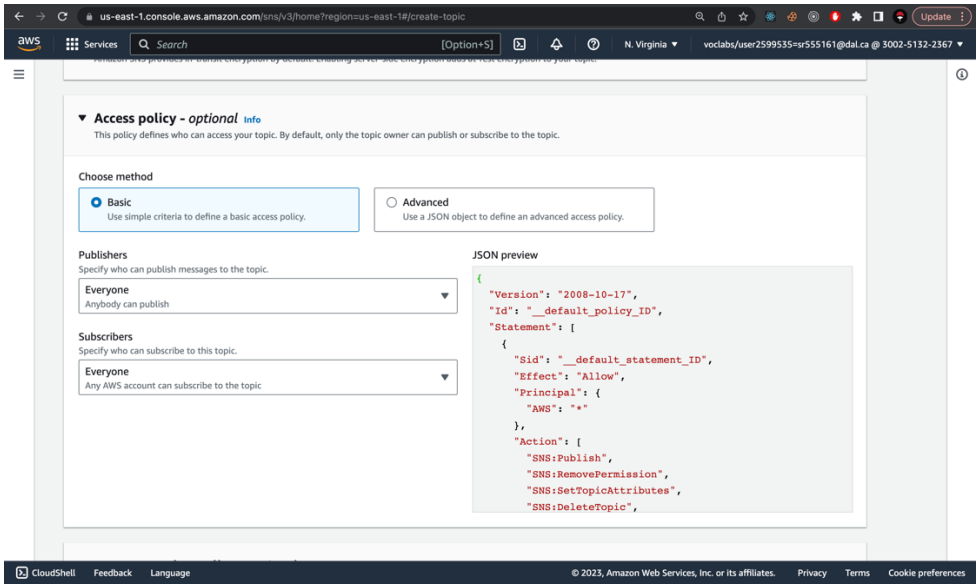

Fig. 10: Creating basic access policy for SNS client.

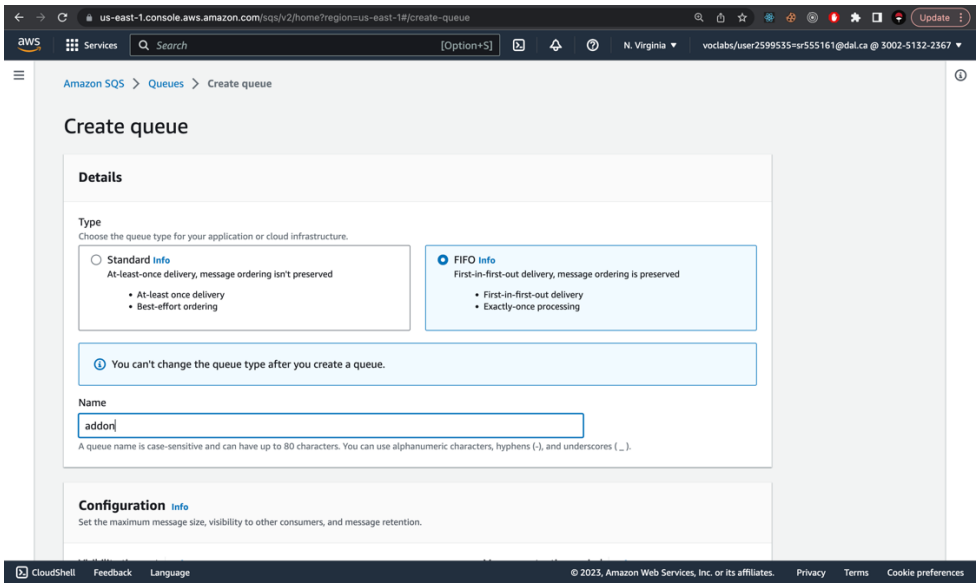

Fig. 11: Creating Standard SQS queue for addon.

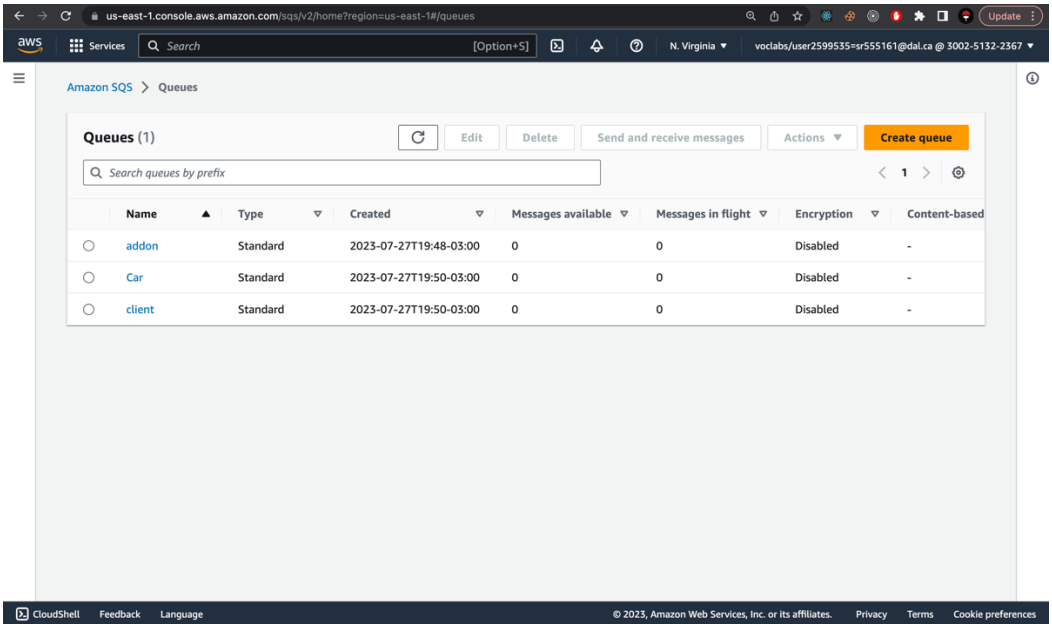

Fig. 12: Similarly create three SQS for three entities.

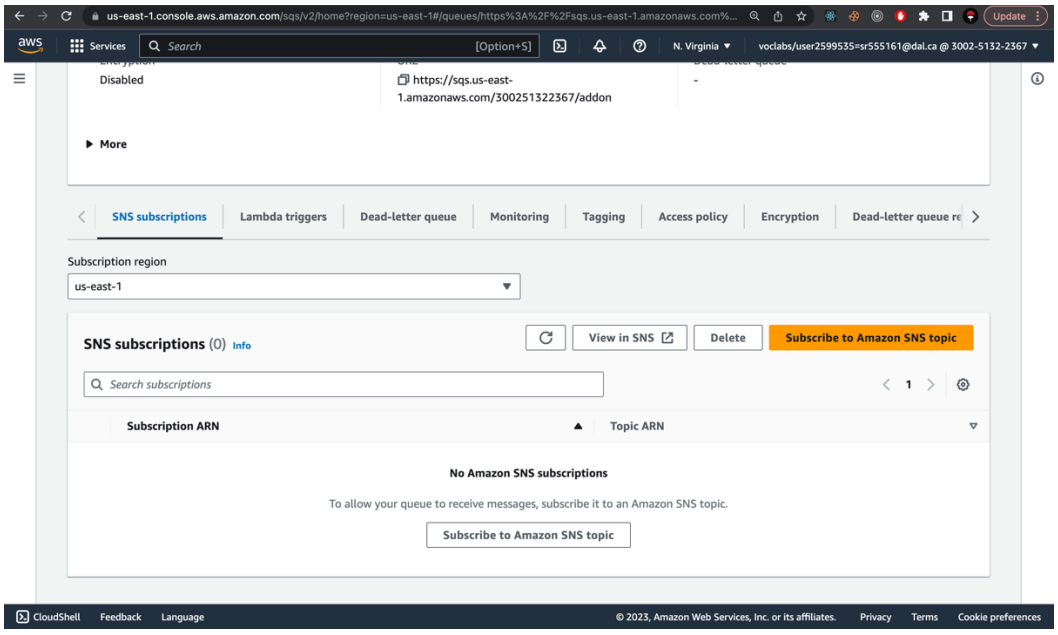

Fig. 13: Creating a SNS subscription for addon.

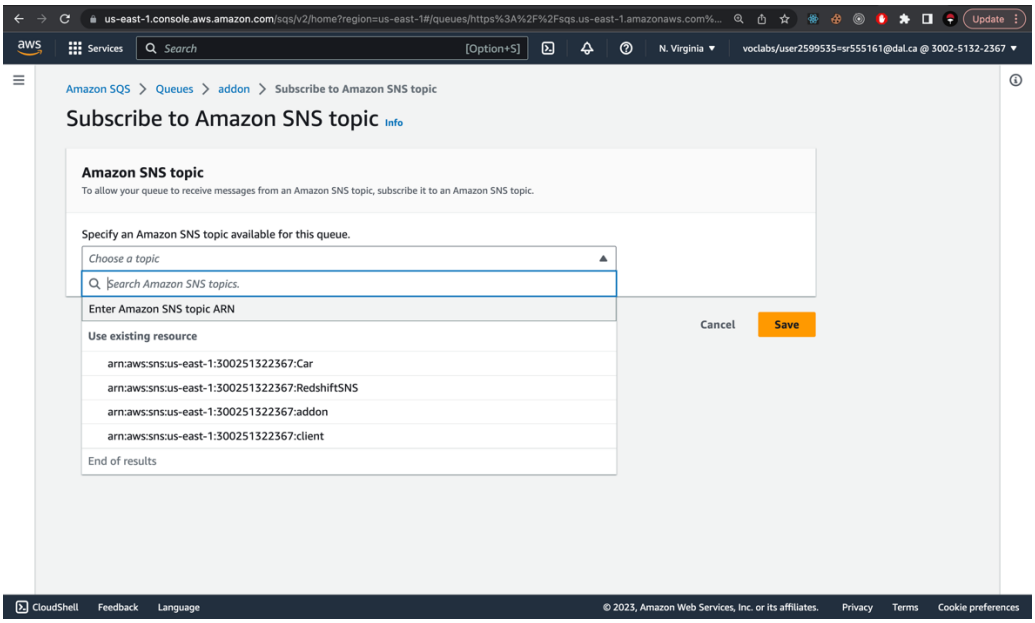

Fig. 14: Choosing addon SNS.

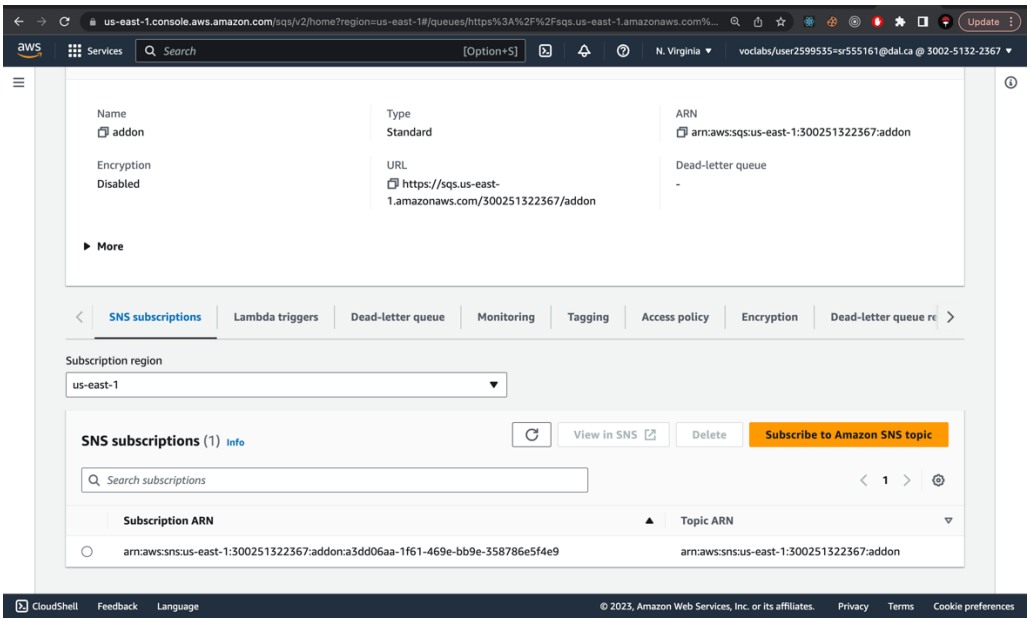

Fig. 15: Subscription for addon SQS created with addon SNS.

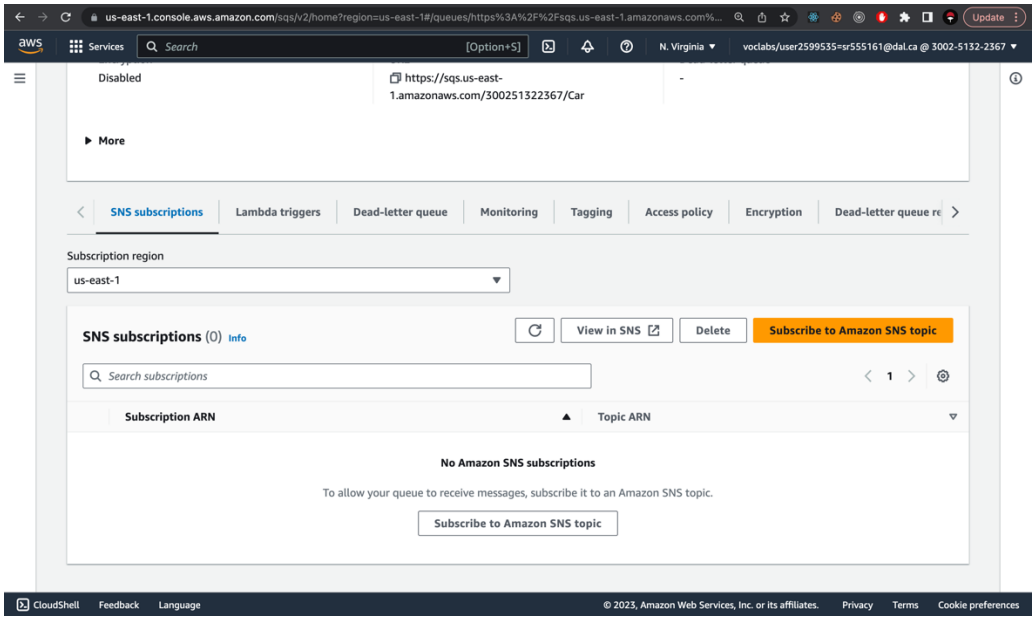

Fig. 16: Subscription for Car SQS.

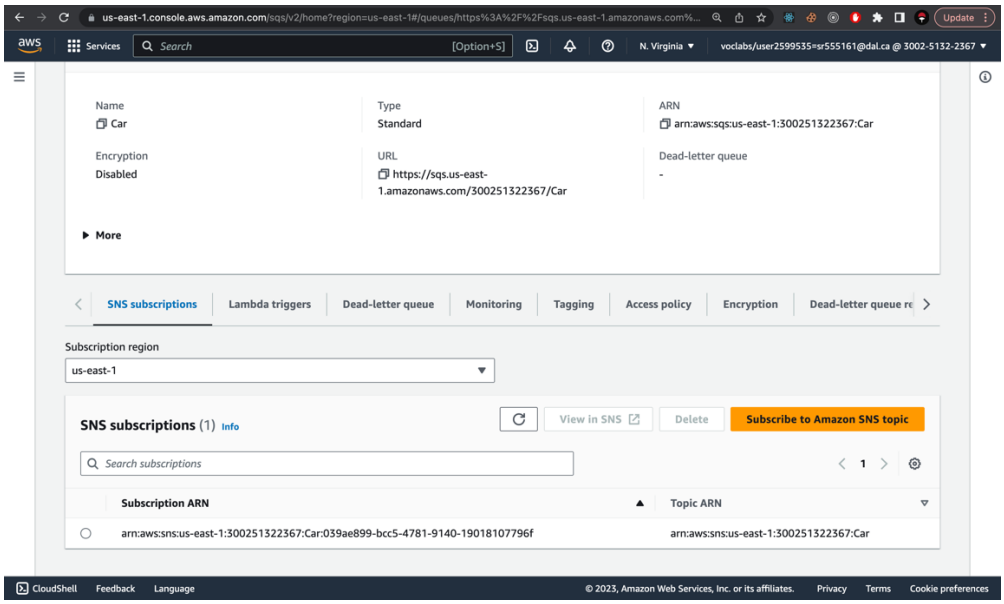

Fig. 17: Subscription for Car SQS created with Car SNS.

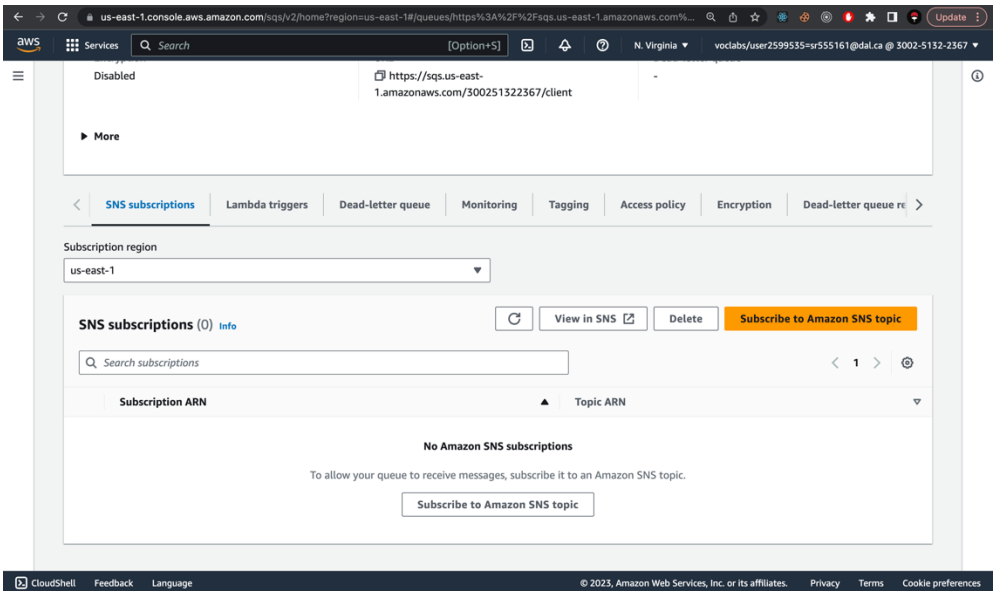

Fig. 18: Subscription for client SQS.

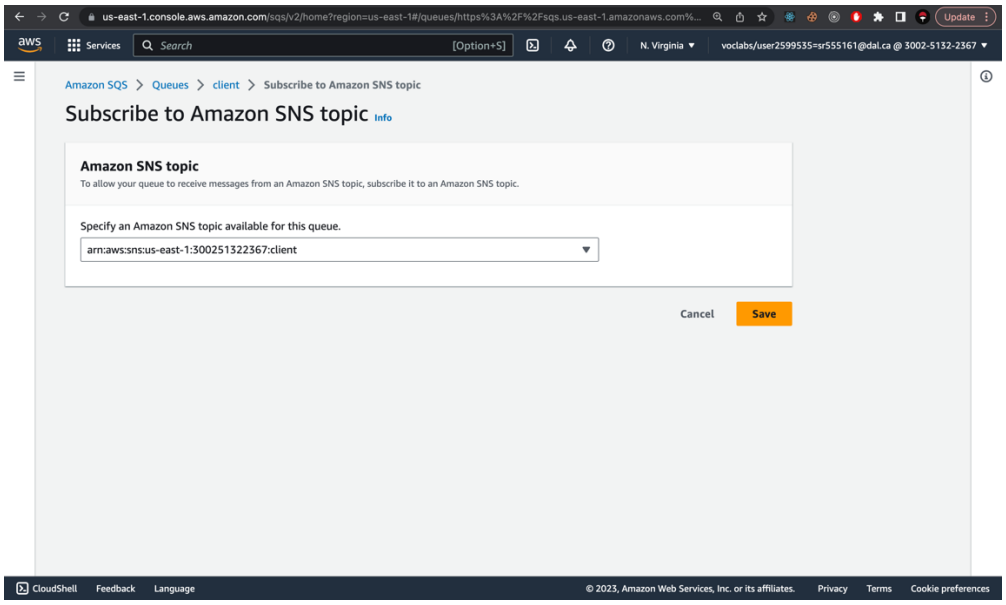

Fig. 19: Subscription for client SQS created with client SNS.

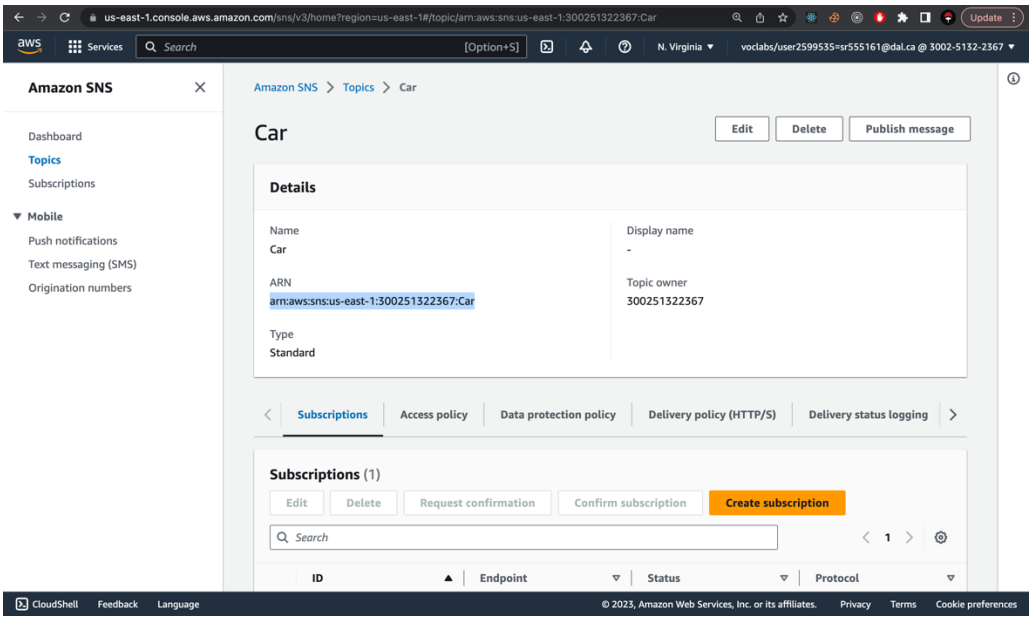

Fig. 20: Copying Car SNS ARN.

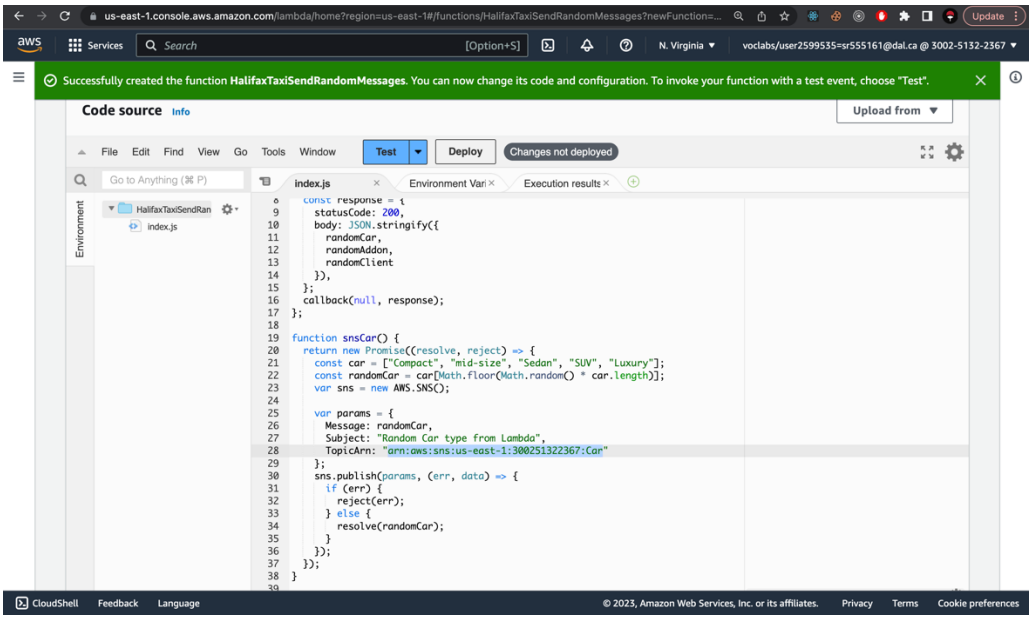

Fig. 21: Pasting the Car SNS ARN in lambda.

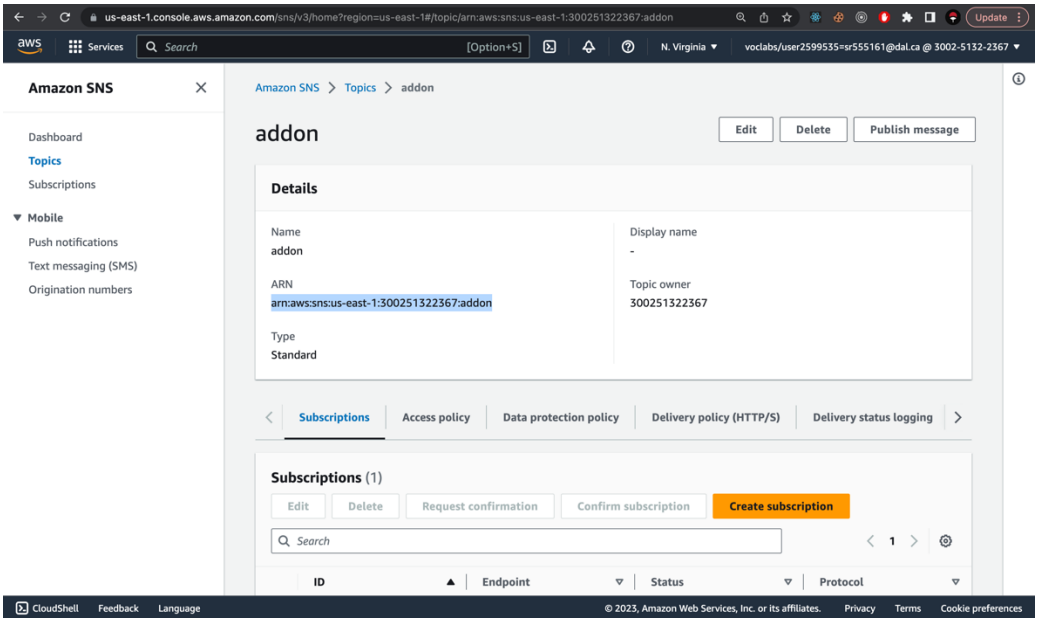

Fig. 22: Copying the addon SNS ARN in lambda.

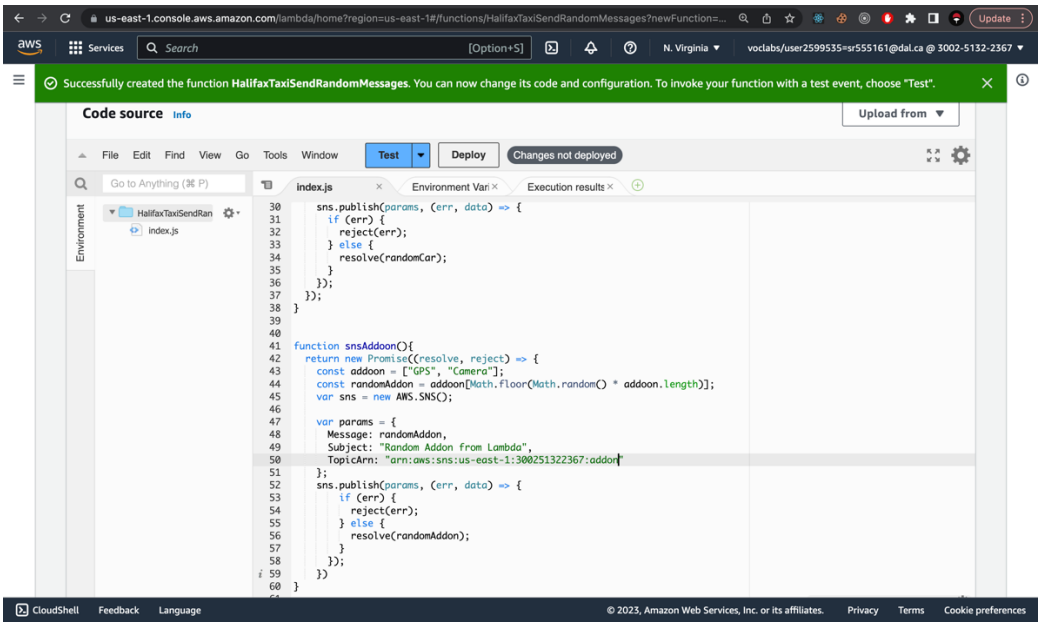

Fig. 23: Pasting the addon SNS ARN in lambda.

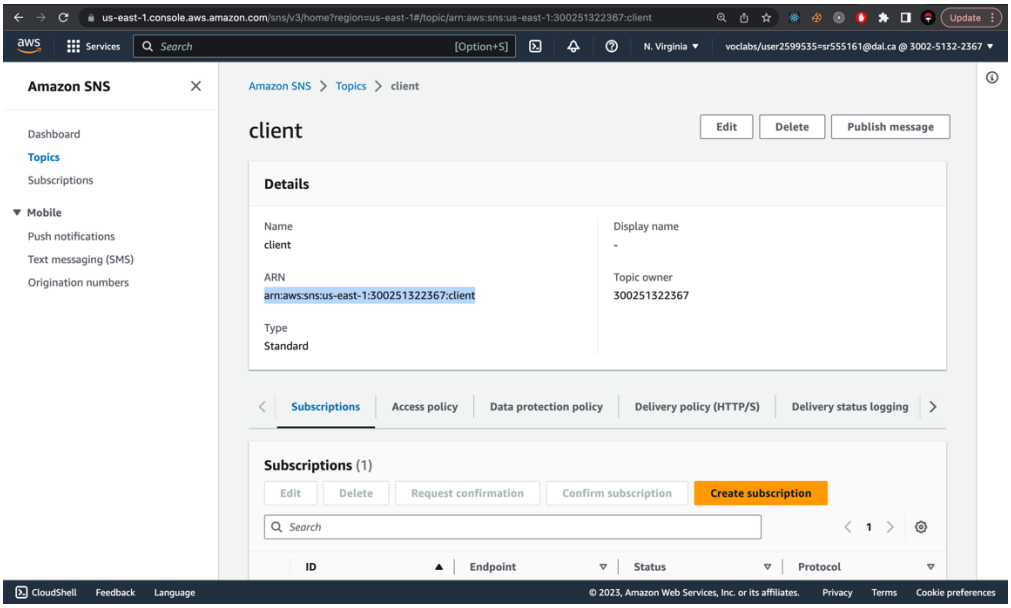

Fig. 24: Copying the client SNS ARN in lambda.

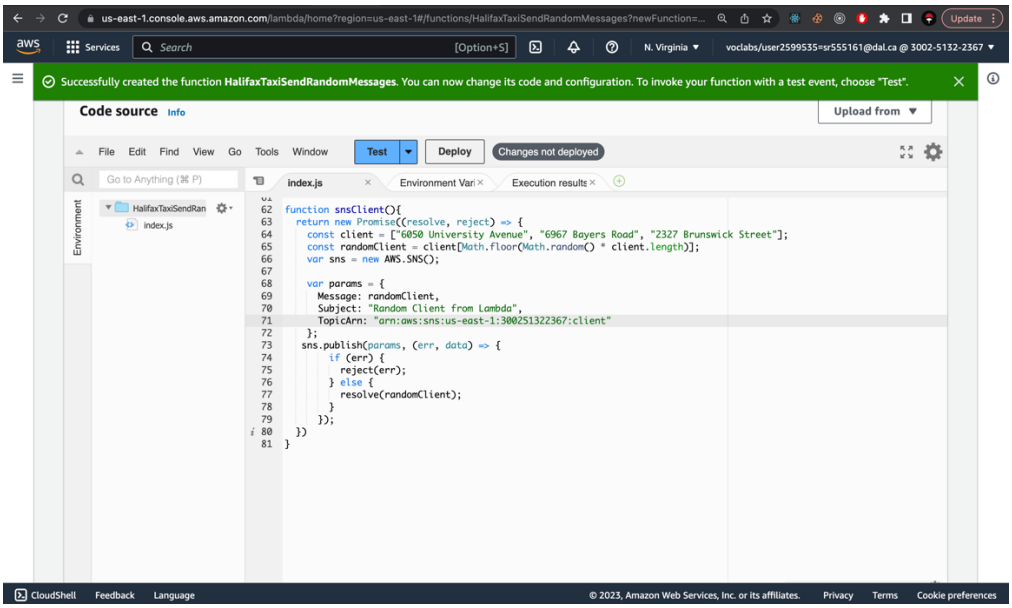

Fig. 25: Pasting the client SNS ARN in lambda.

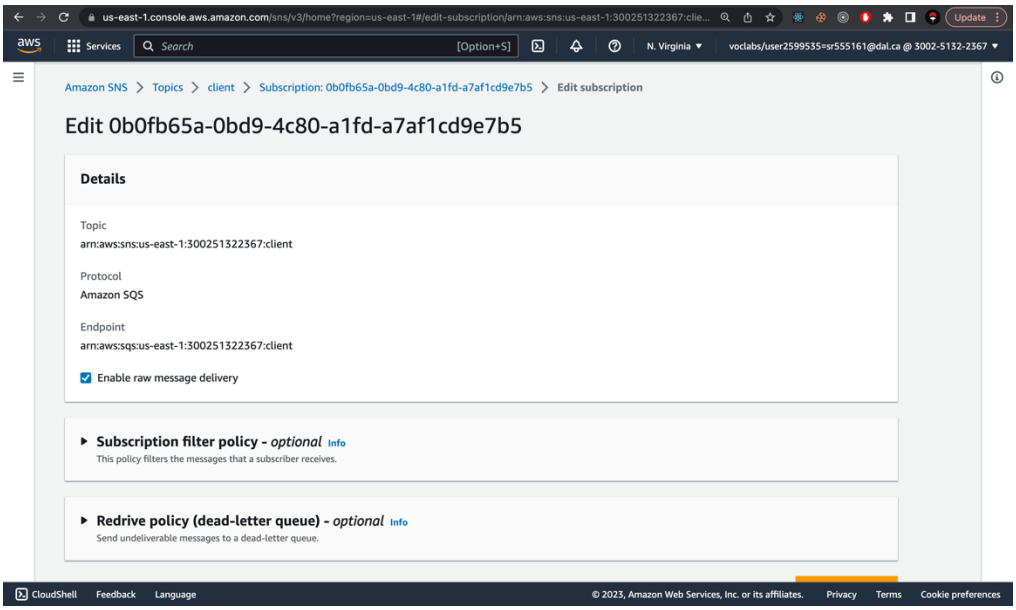

Fig. 26: Enabling Raw Message Delivery for client SNS.

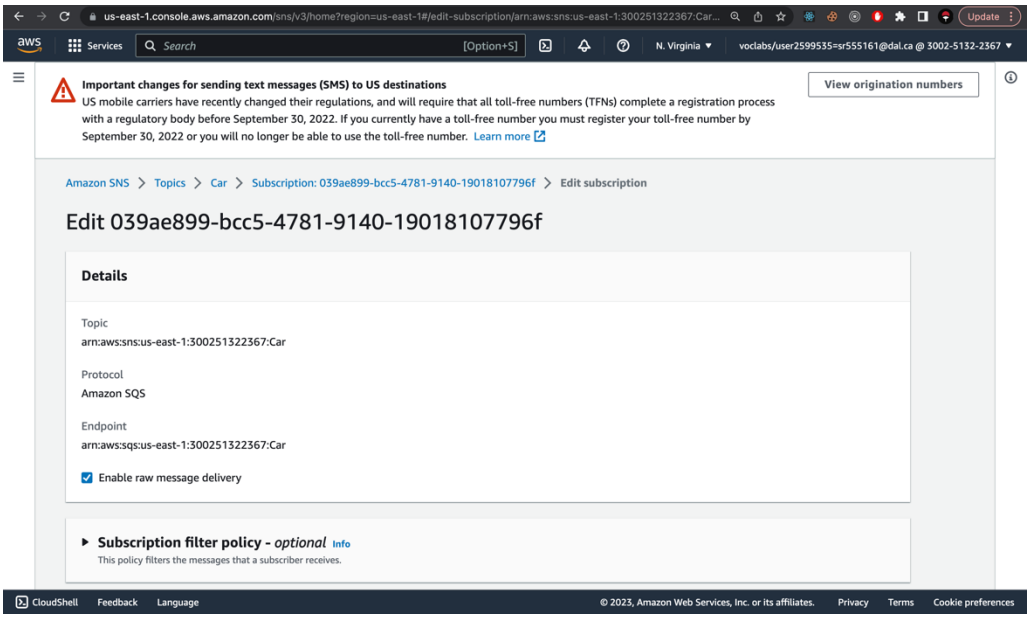

Fig. 27: Enabling Raw Message Delivery for Car SNS.

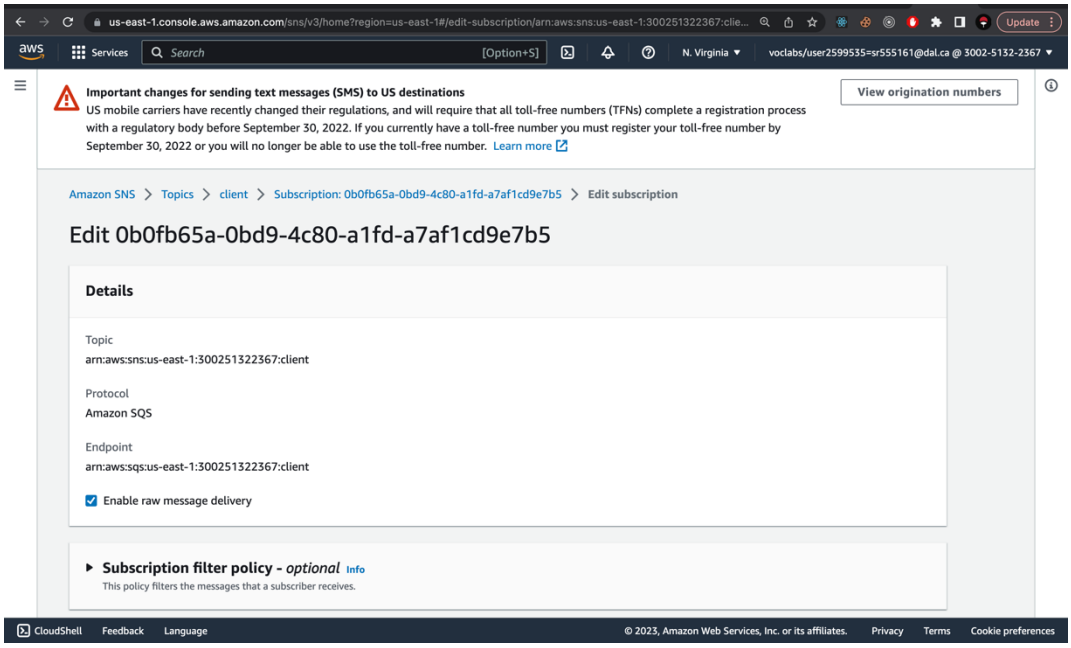

Fig. 28: Enabling Raw Message Delivery for addon.

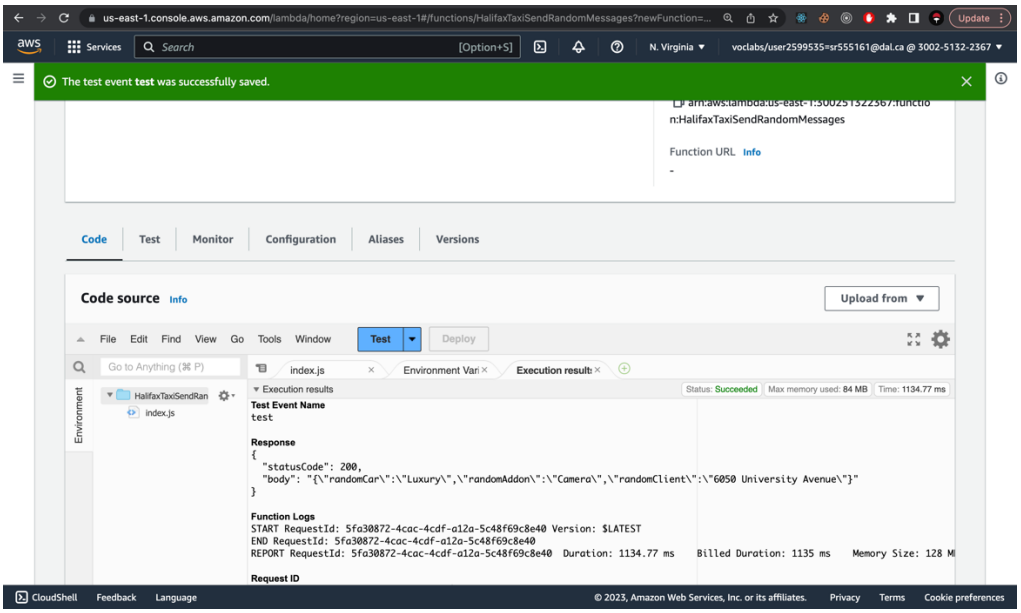

Fig. 29: Random Messages sent to AWS SQS through SNS.

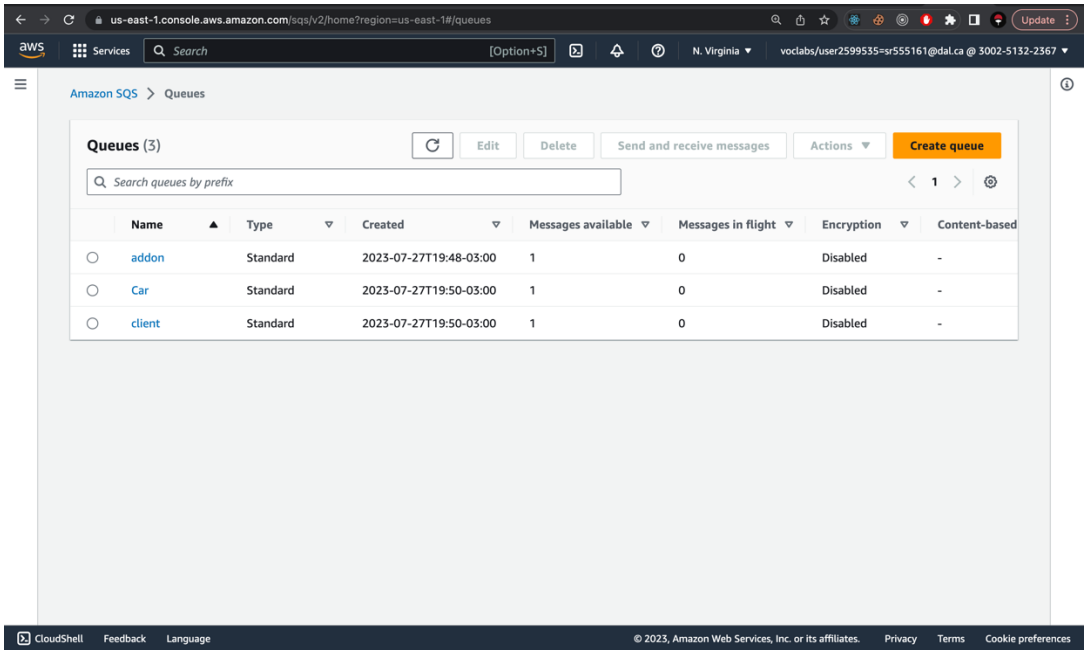

Fig. 30: Messages available in SQS Queue.

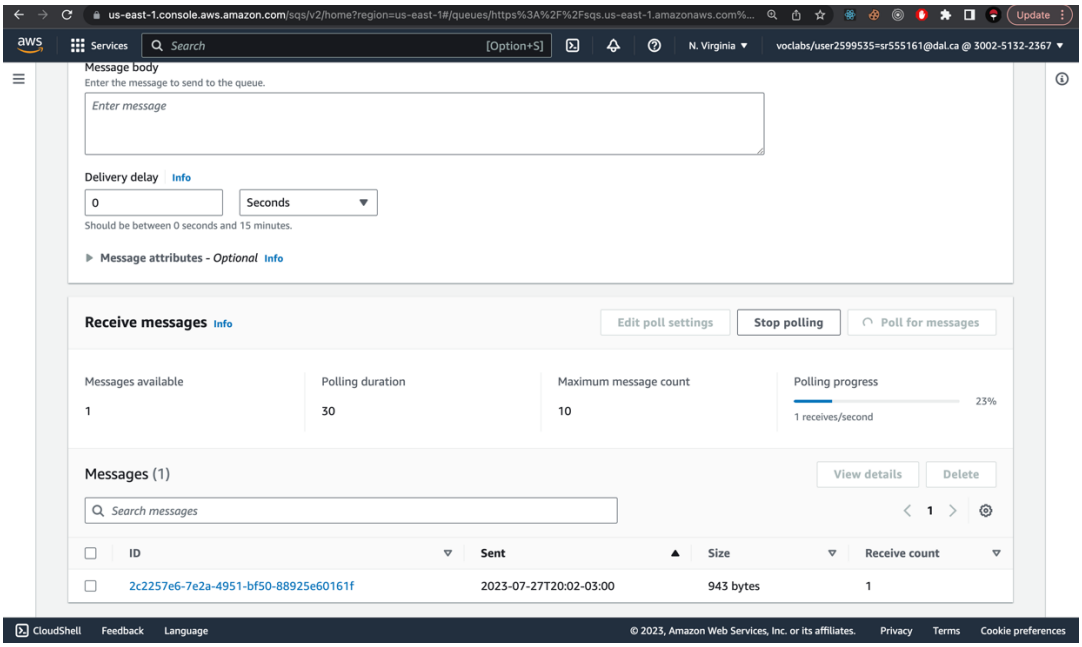

Fig. 31: Polling addon SQS for messages.

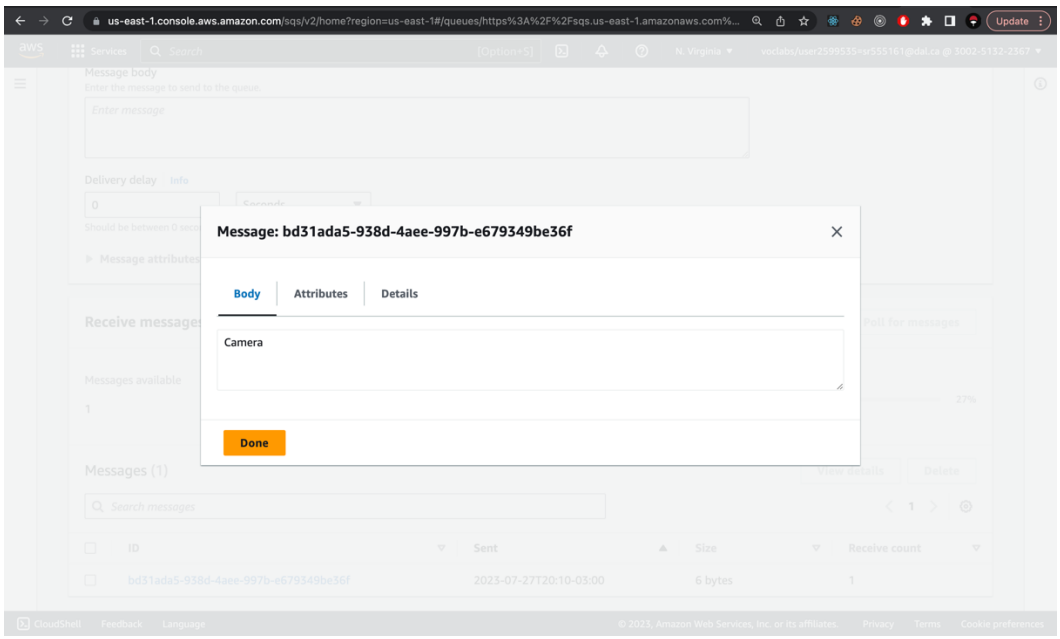

Fig. 32: Message from lambda to SNS to SQS.

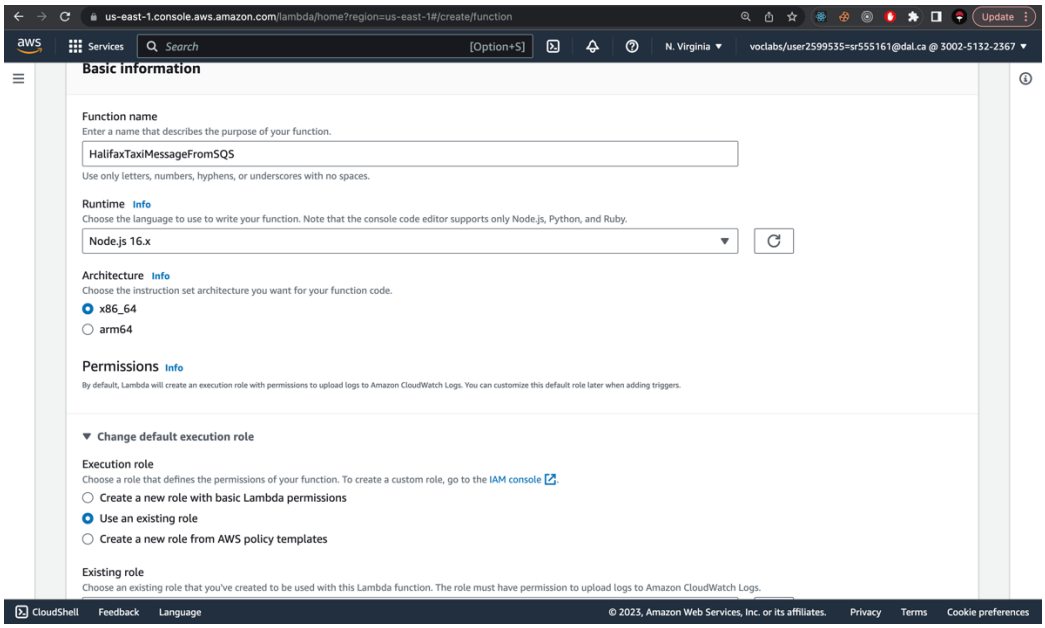

Fig. 33: Another lambda to retrieve all the messages from Queue.

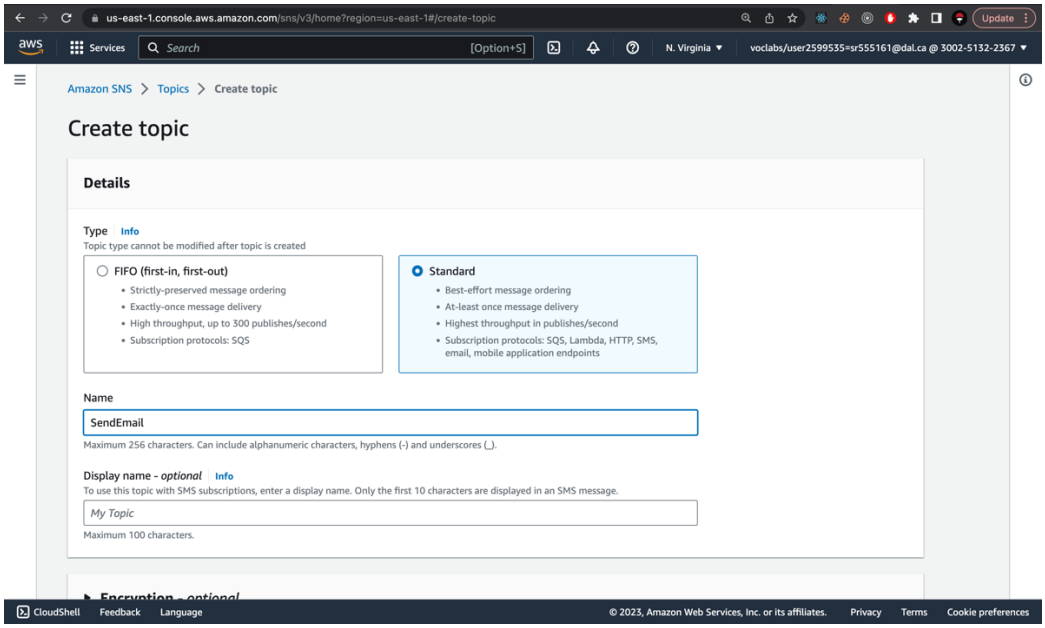

Fig. 34: Creating standard SNS to send email.

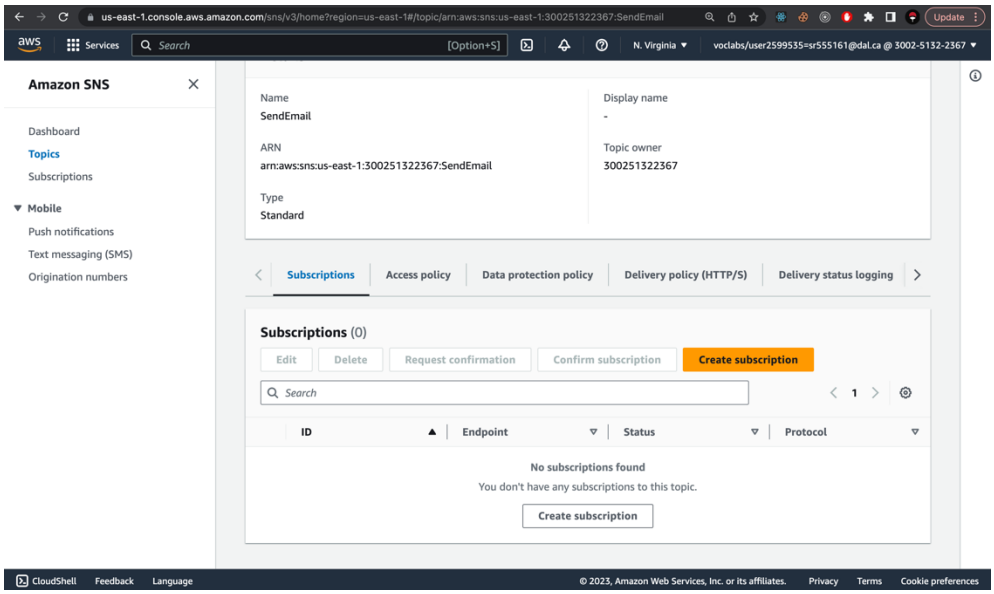

Fig. 35: Creating subscription for SendEmail SNS.

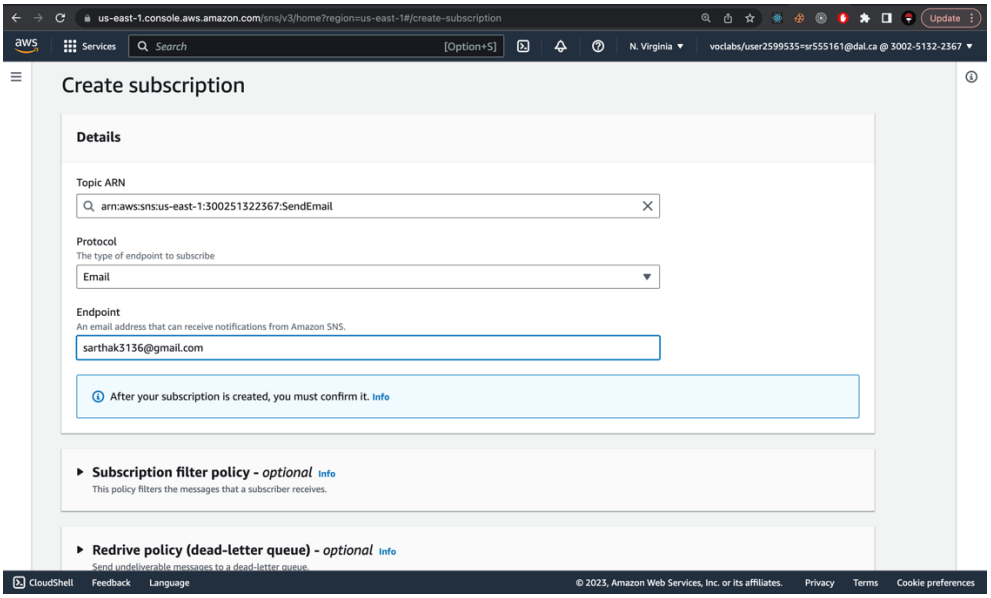

Fig. 36: Choosing Protocol as email and providing email ID.

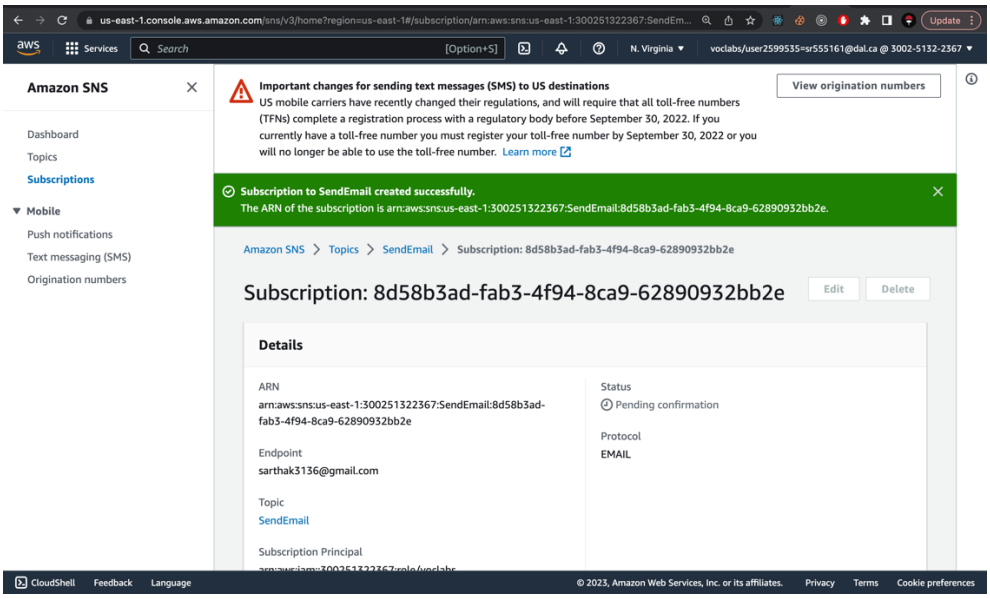

Fig. 37: Subscription for email created.

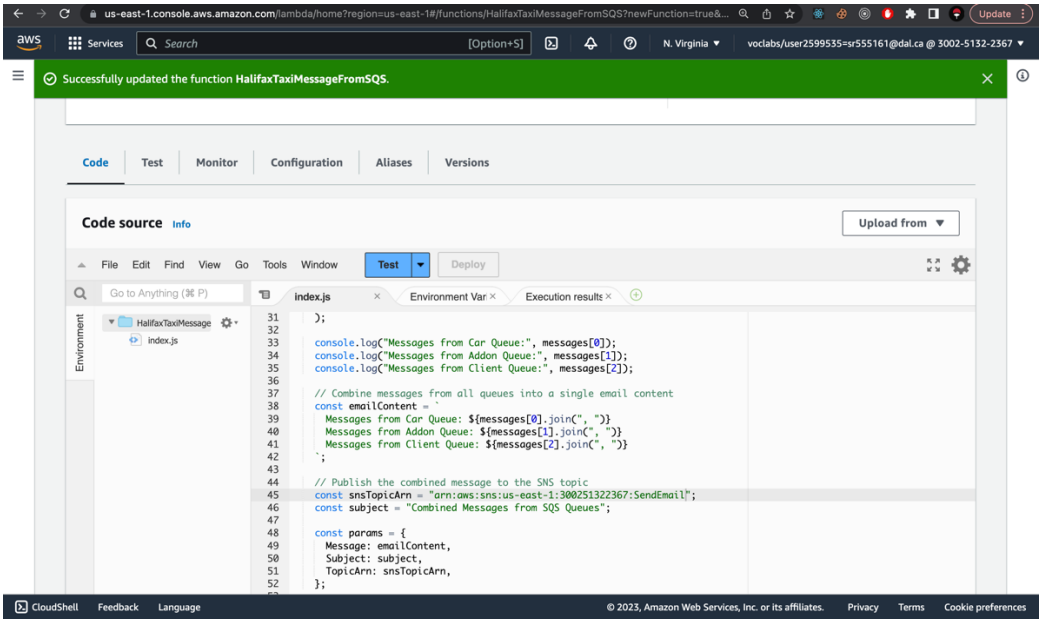

Fig. 38: Putting the SendEmail ARN in the second lambda.

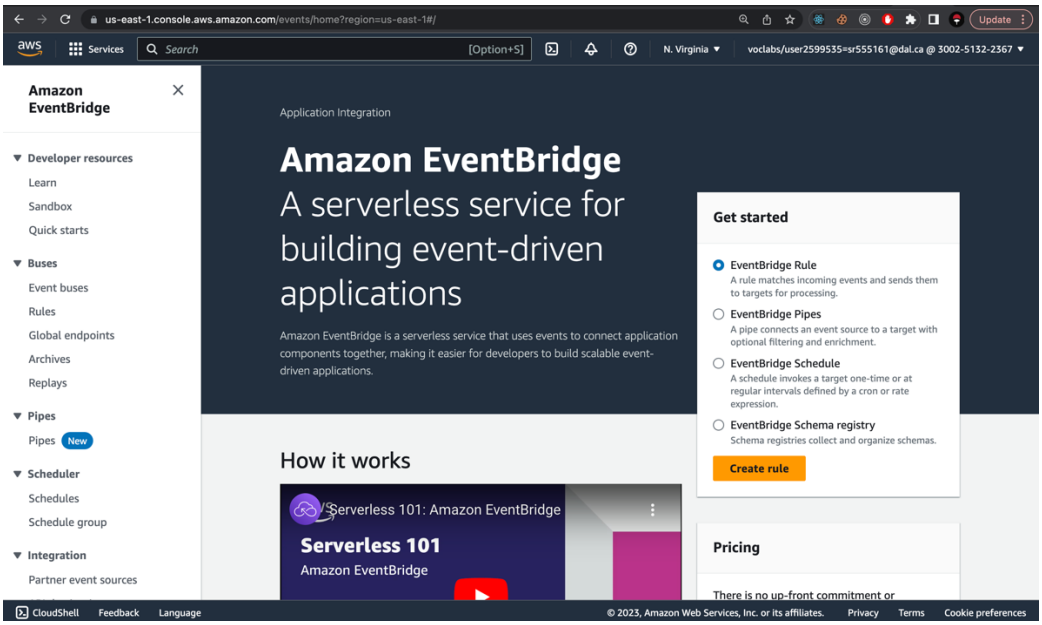

Fig. 39: Creating an event bridge to trigger lambda every 2 minutes.

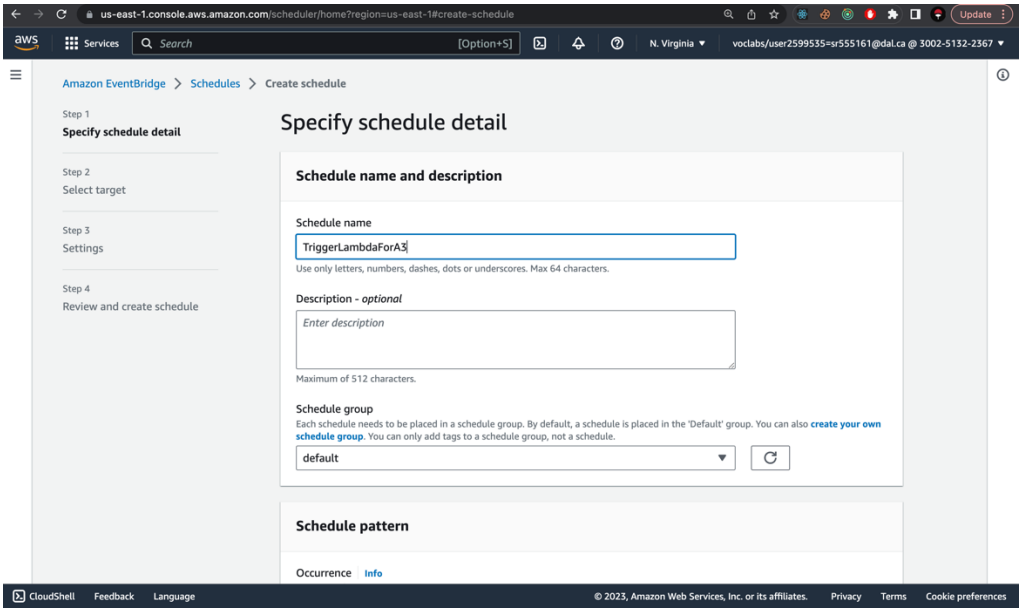

Fig. 40: Naming Event Bridge Schedule.

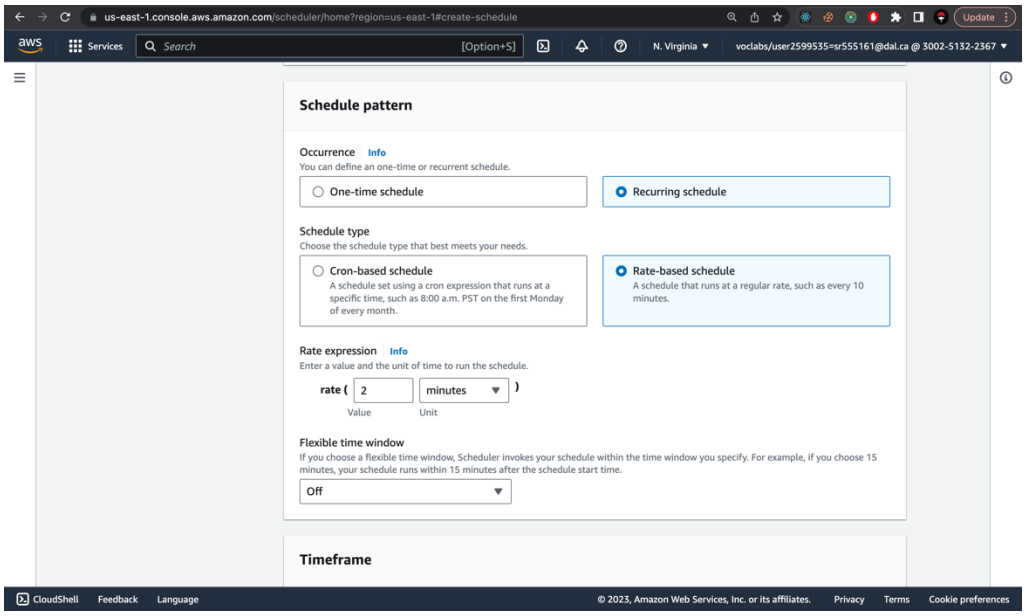

Fig. 41: Recurring schedule selected for every 2 minutes invocation.

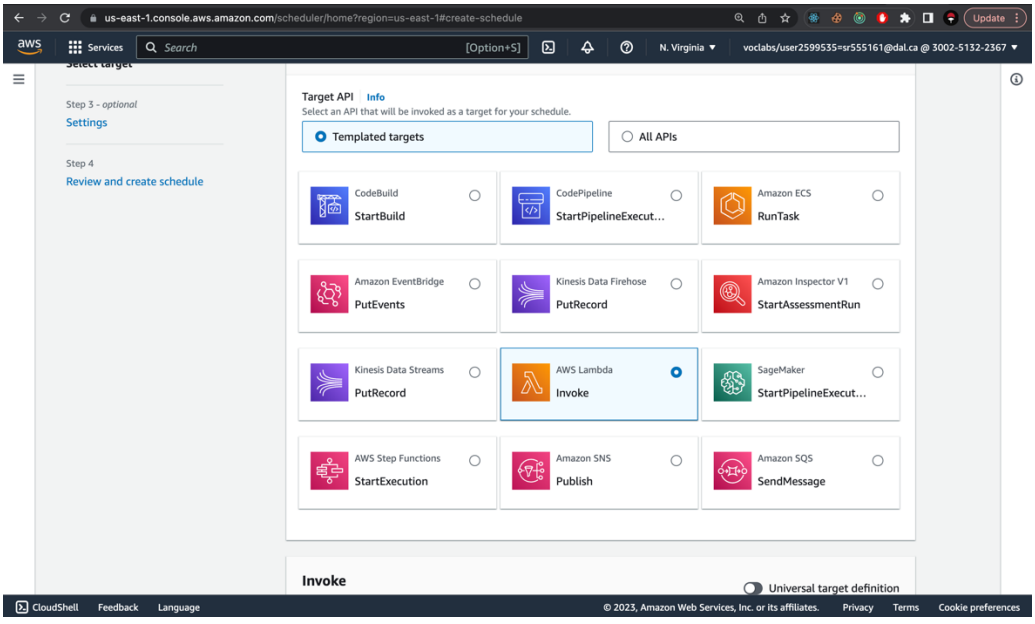

Fig. 42: Lambda selected for invocation.

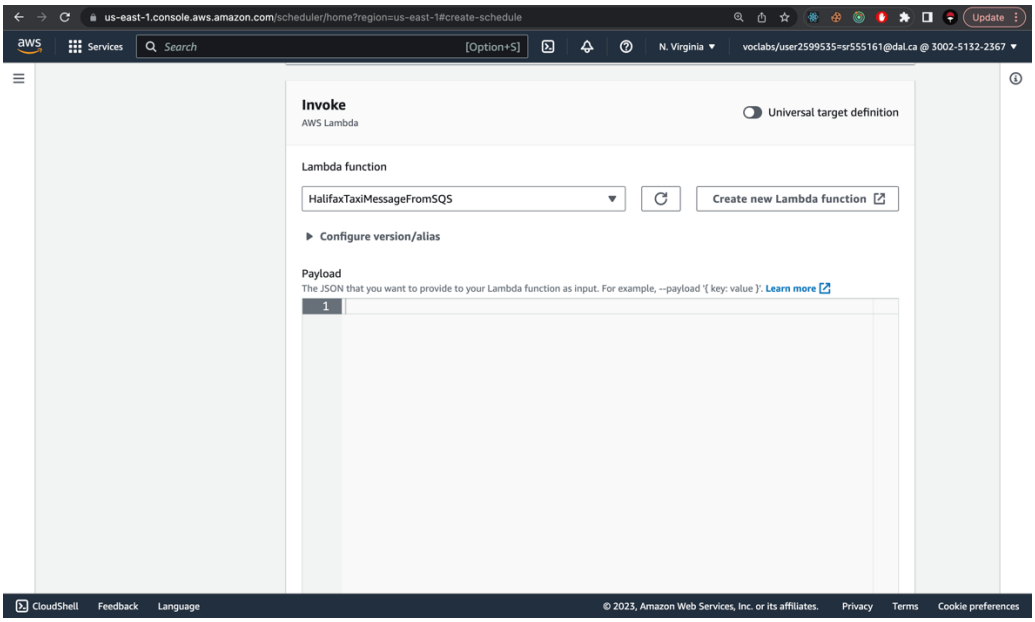

Fig. 43: Selecting the Lambda function to invoke.

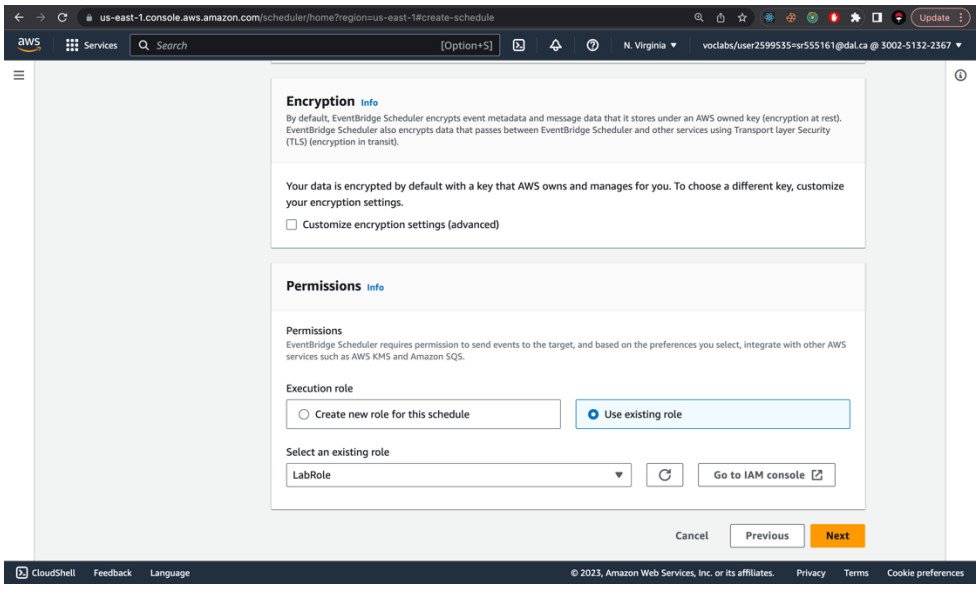

Fig. 44: Keeping the same IAM role as Lambda.

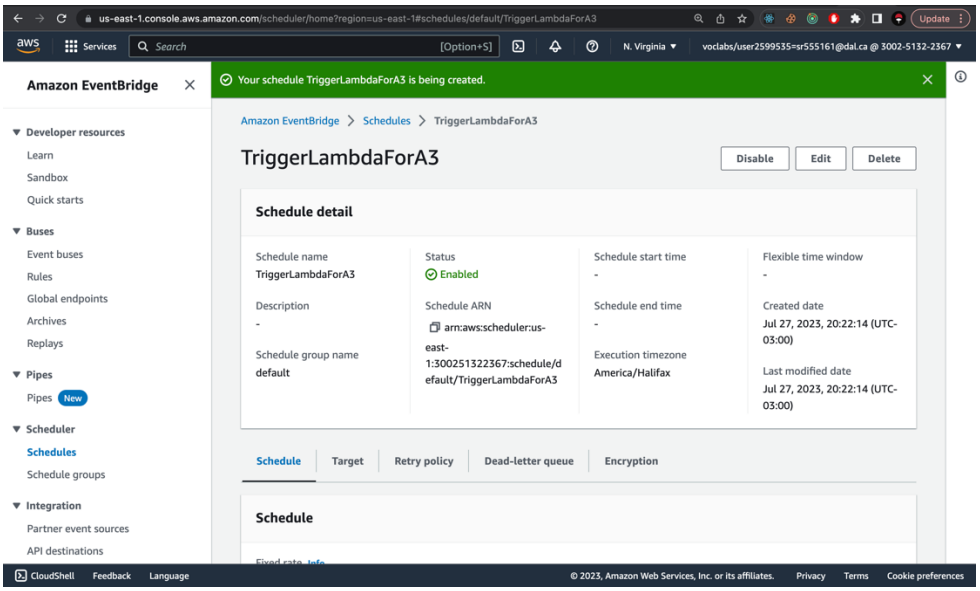

Fig. 45: Event bridge for lambda created.

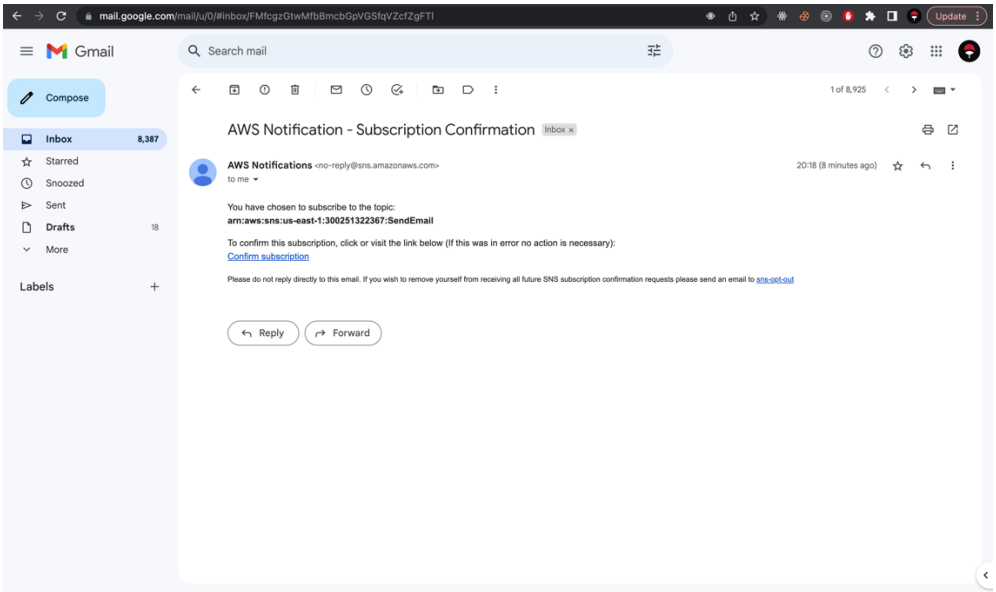

Fig. 46: Subscription permission email received.

Fig. 47: Subscription accepted.

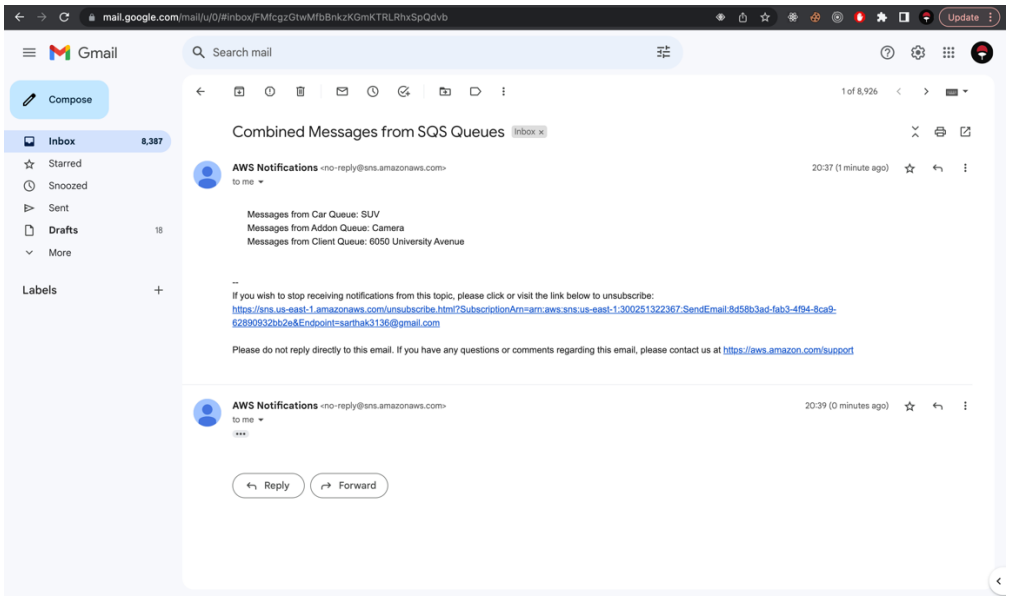

Fig. 48: After 2 minutes event bridge invoked lambda which sent email.