# Message queue service using AWS SNS and SQS

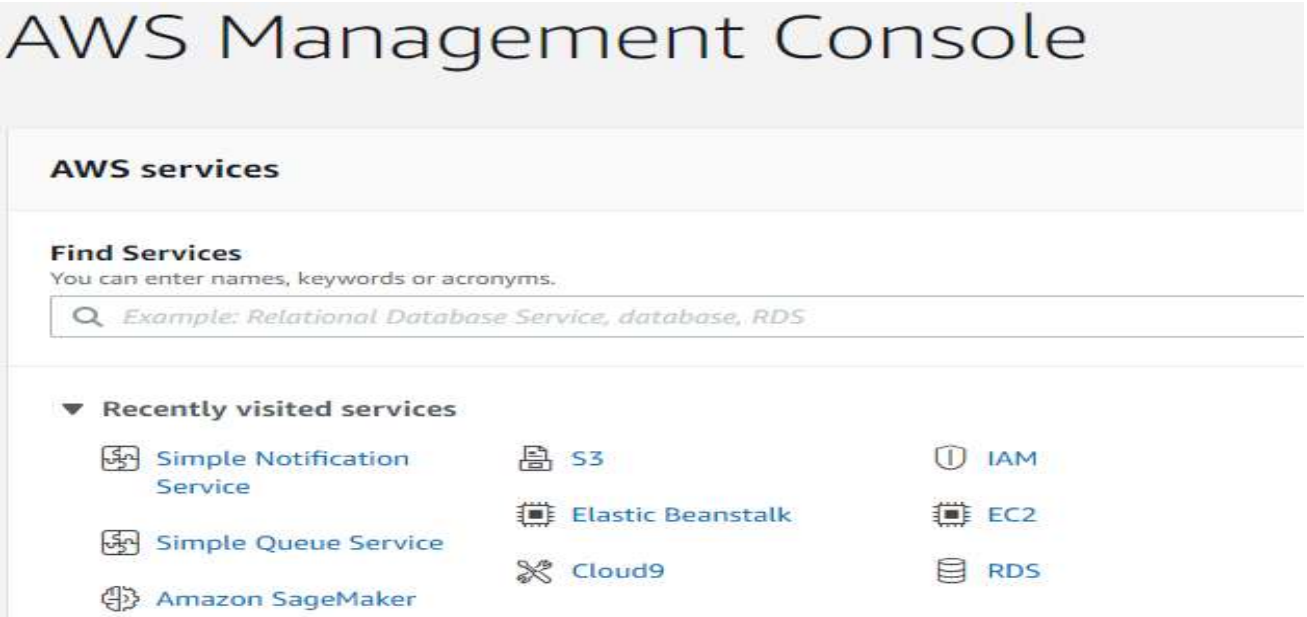
## Theory of SNS:

Amazon Simple Notification Service (Amazon SNS) is a web service that coordinates and manages the delivery or sending of messages to subscribing endpoints or clients. In Amazon SNS, there are two types of clients—publishers and subscribers—also referred to as producers and consumers. Publishers communicate asynchronously with subscribers by producing and sending a message to a topic, which is a logical access point and communication channel. Subscribers (that is, web servers, email addresses, Amazon SQS queues, AWS Lambda functions) consume or receive the message or notification over one of the supported protocols (that is, Amazon SQS, HTTP/S, email, SMS, Lambda) when they are subscribed to the topic.

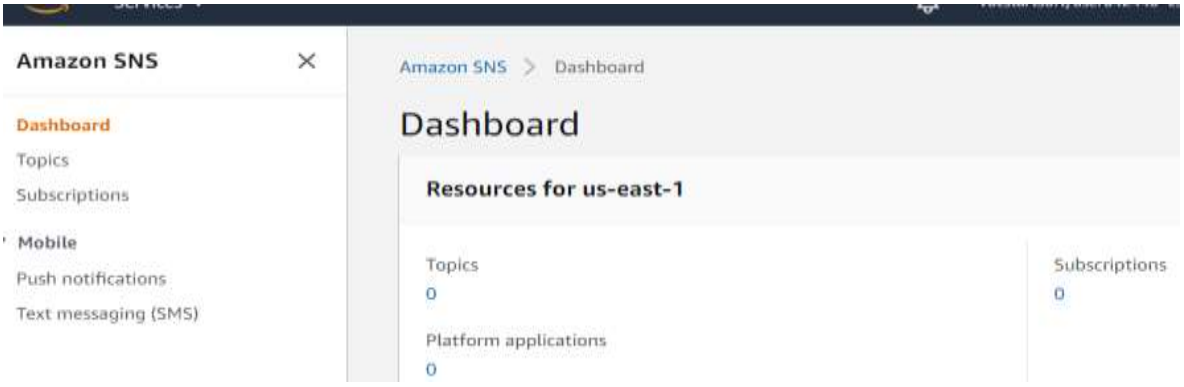
When using Amazon SNS, you (as the owner) create a topic and control access to it by defining policies that determine which publishers and subscribers can communicate with the topic. A publisher sends messages to topics that they have created or to topics they have permission to publish to. Instead of including a specific destination address in each message, a publisher sends a message to the topic. Amazon SNS matches the topic to a list of subscribers who have subscribed to that topic, and delivers the message to each of those subscribers. Each topic has a unique name that identifies the Amazon SNS endpoint for publishers to post messages and subscribers to register for notifications. Subscribers receive all messages published to the topics to which they subscribe, and all subscribers to a topic receive the same messages.

### Procedure:

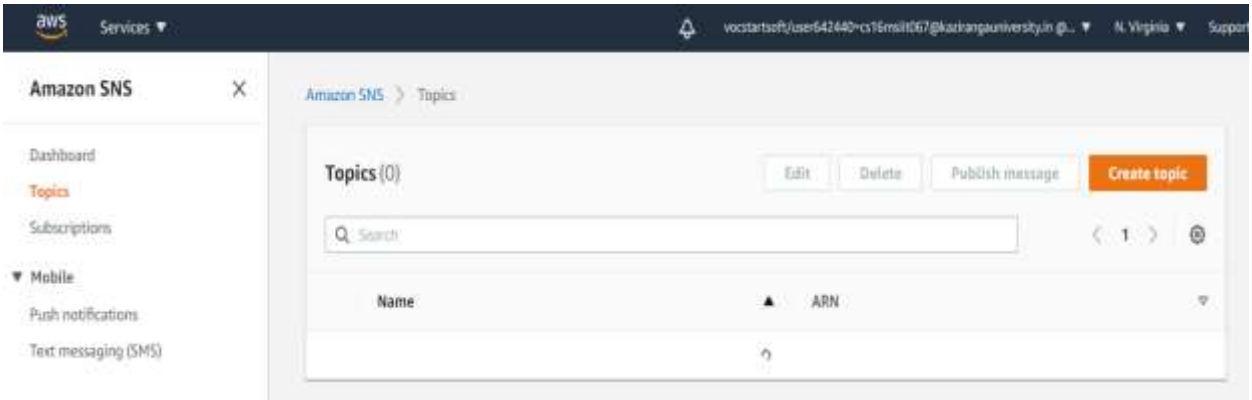
1: From the service catalogue select **“SNS and SQS” (Simple Notification Service and Simple Queue Service).**



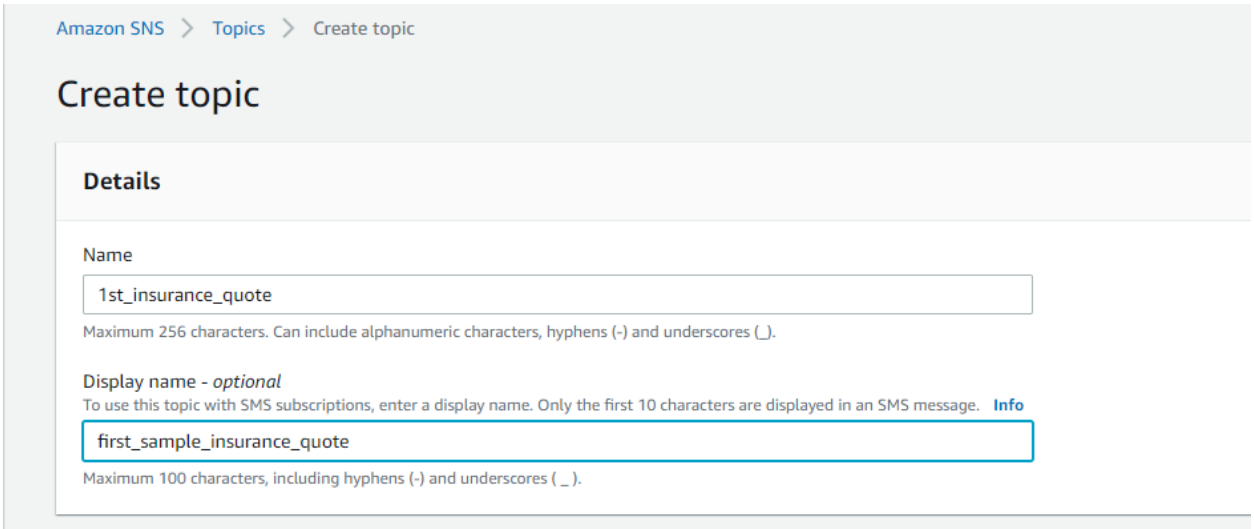

2: First we will go to the SNS dashboard.



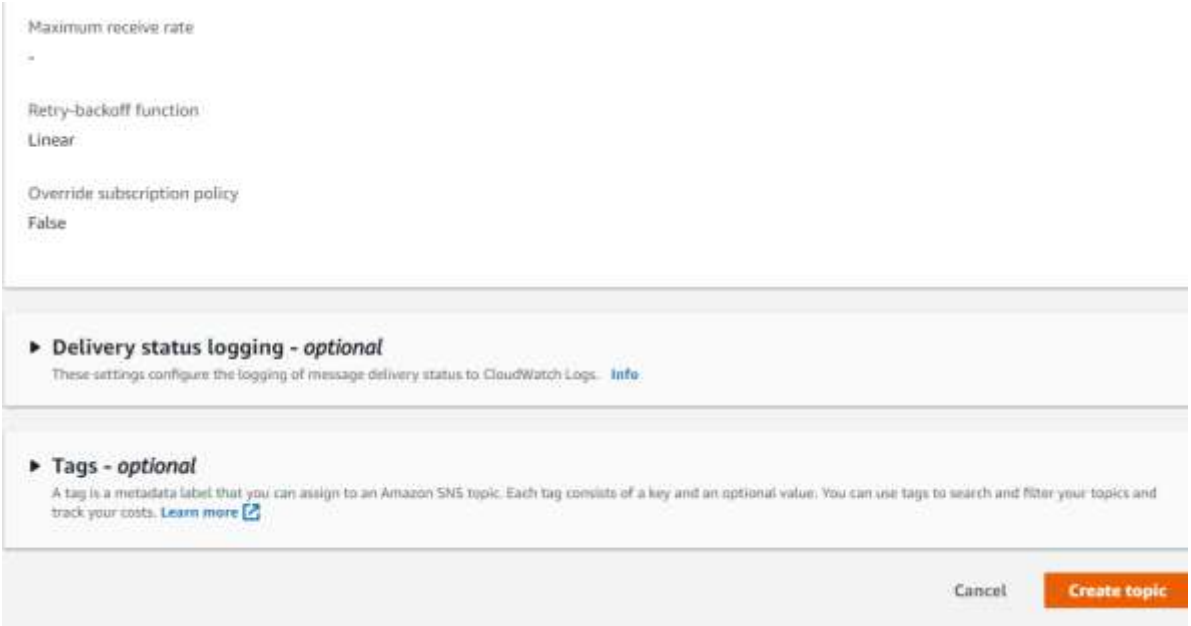
3: Click on the **“Create topic”**



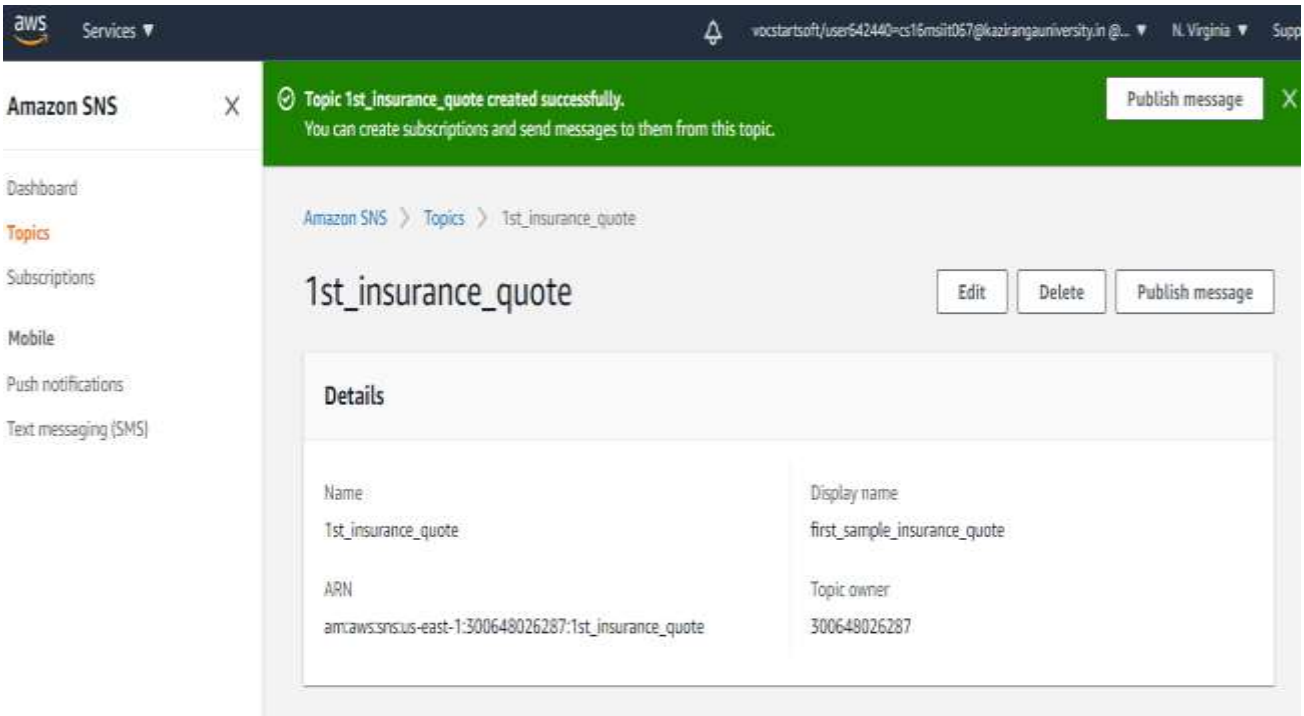

4: On the details we will give our “name” and “Display name”.



5: Click on the **Create topic.**




6: Our Topic is successfully created.

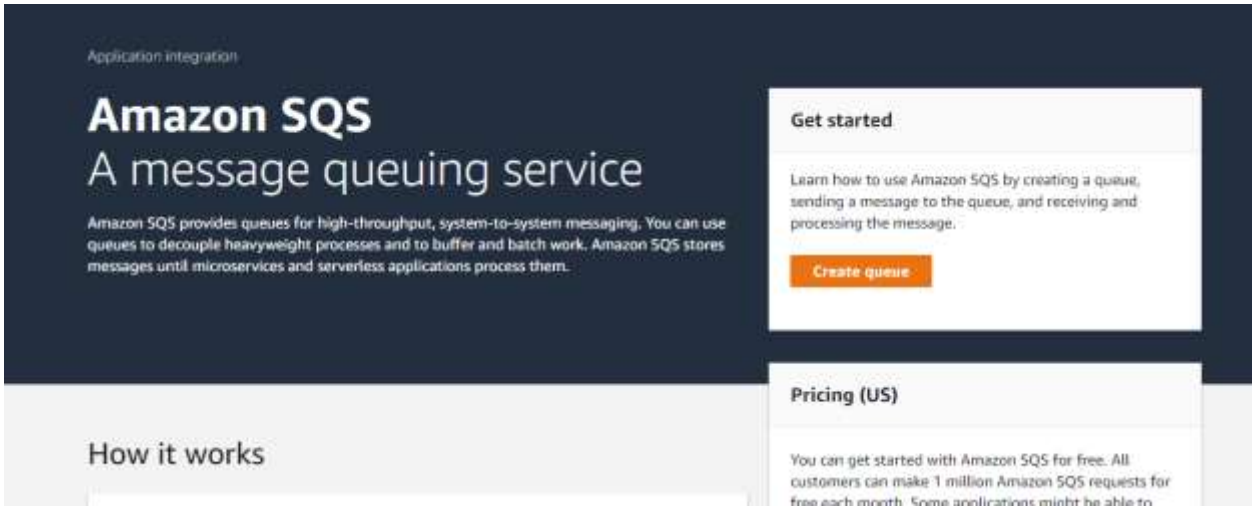


## Theory of SQS:

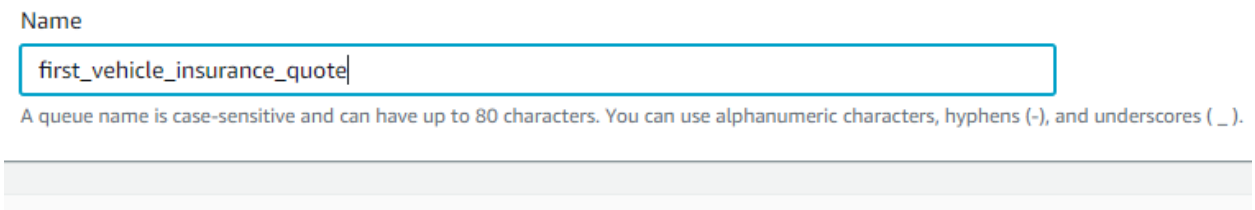
Amazon Simple Queue Service (SQS) is a fully managed message queuing service that enables us to decouple and scale micro services, distributed systems, and serverless applications. SQS eliminates the complexity and overhead associated with managing and operating message oriented middleware, and empowers developers to focus on differentiating work. Using SQS, we can send, store, and receive messages between software components at any volume, without losing messages or requiring other services to be available. Get started with SQS in minutes using the AWS console, Command Line Interface or SDK of our choice, and three simple commands.

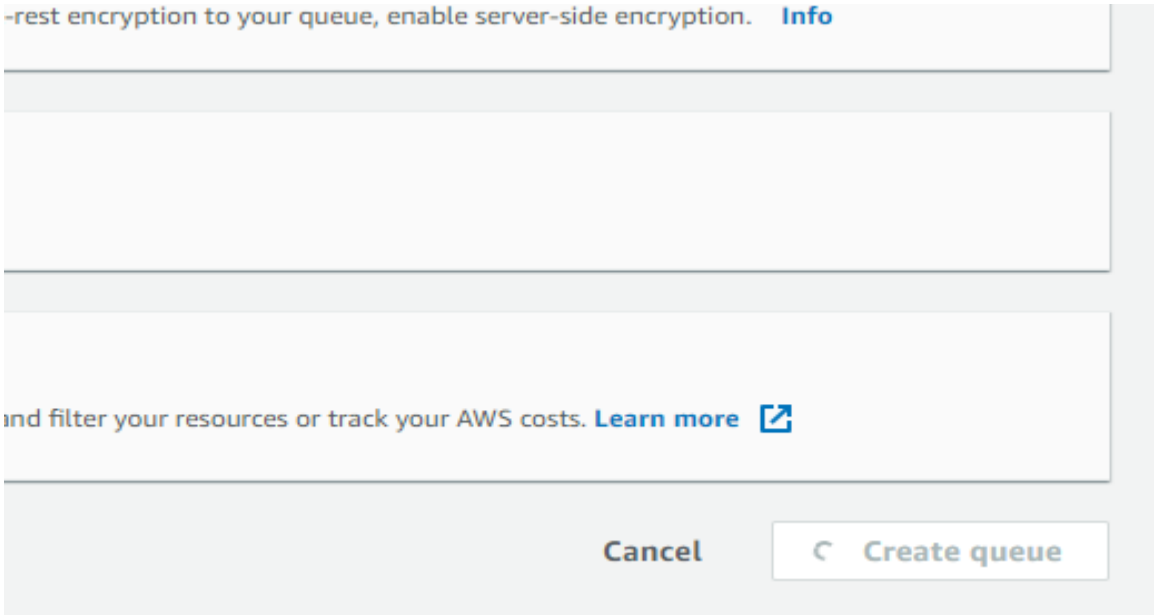
SQS offers two types of message queues. Standard queues offer maximum throughput, best-effort ordering, and at-least-once delivery. SQS FIFO queues are designed to guarantee that messages are processed exactly once, in the exact order that they are sent

7: Select on the SQS and click on the **“Create queue”.**

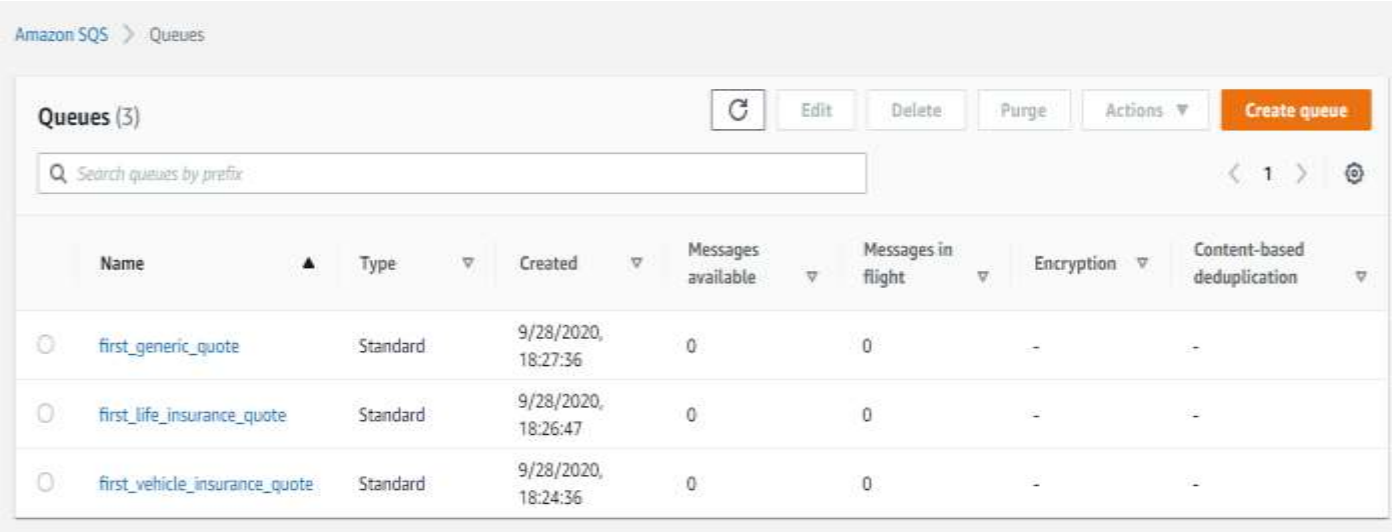



8: We will give the name for the queue and click on the create queue.

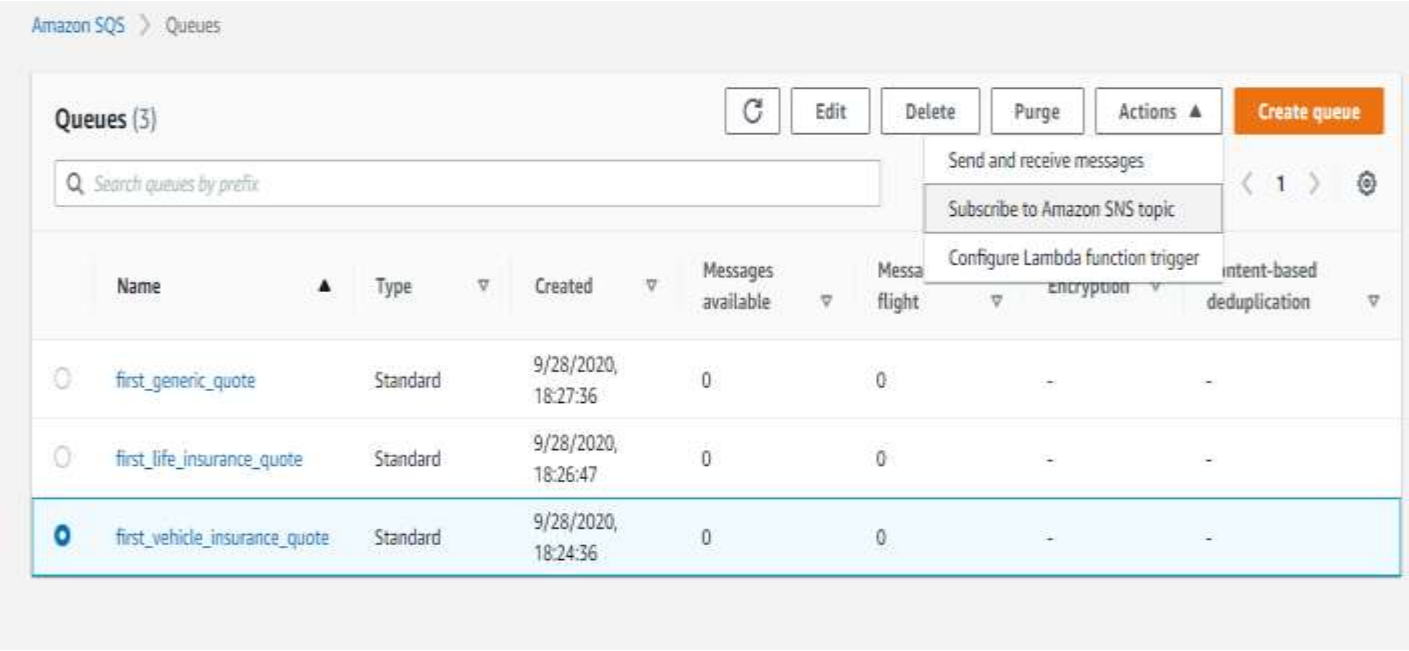




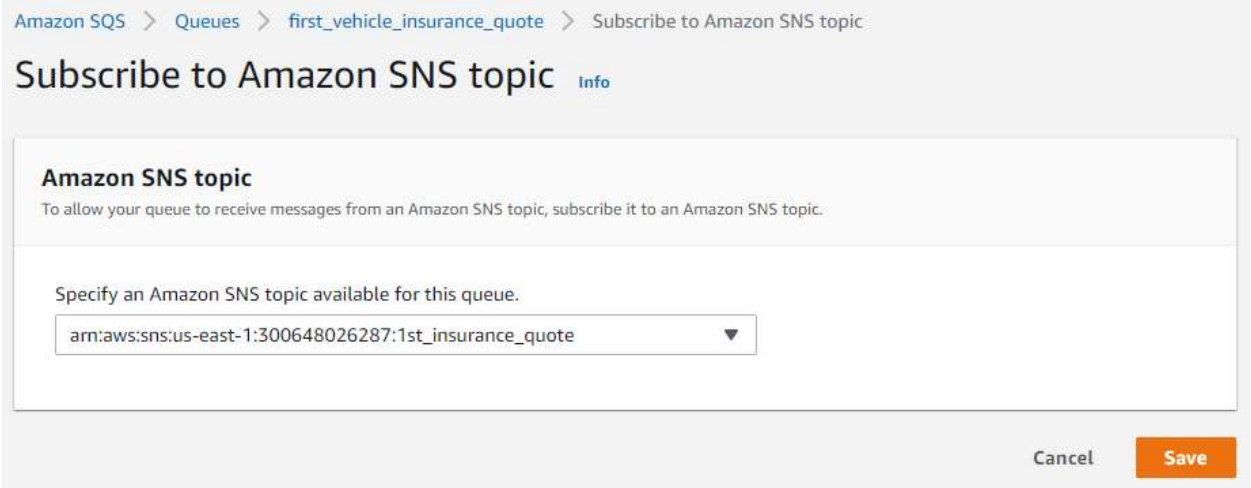

9: Like this we will Create  queue (here we have created 3 queues)



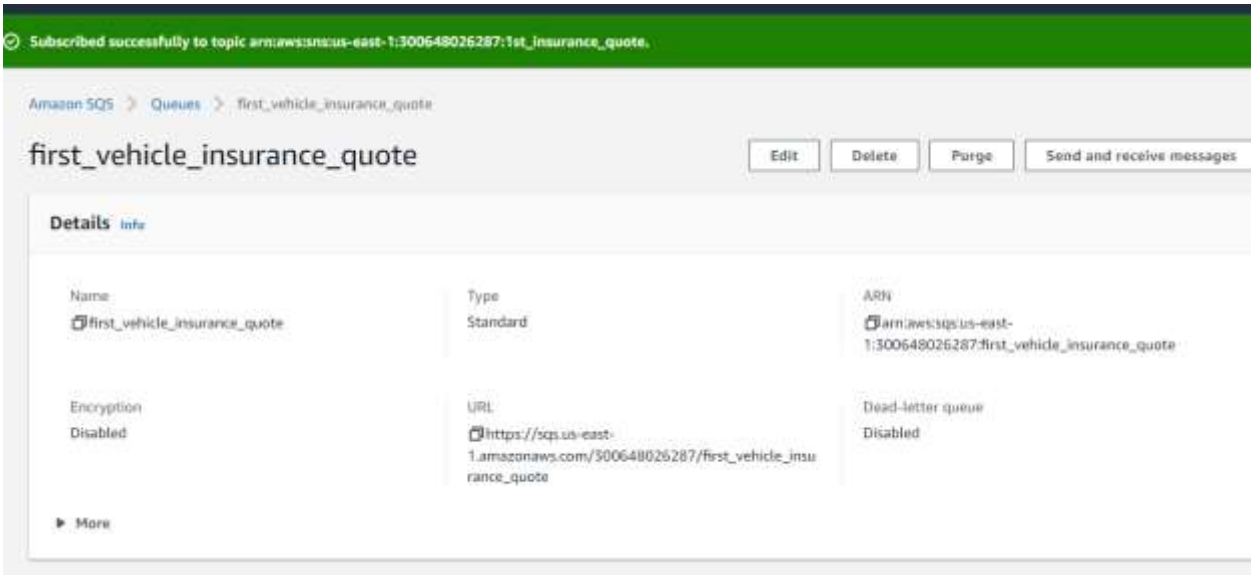

10: Now we will select one queue and from actions select **“Subscribe to Amazon SNS topic”.**



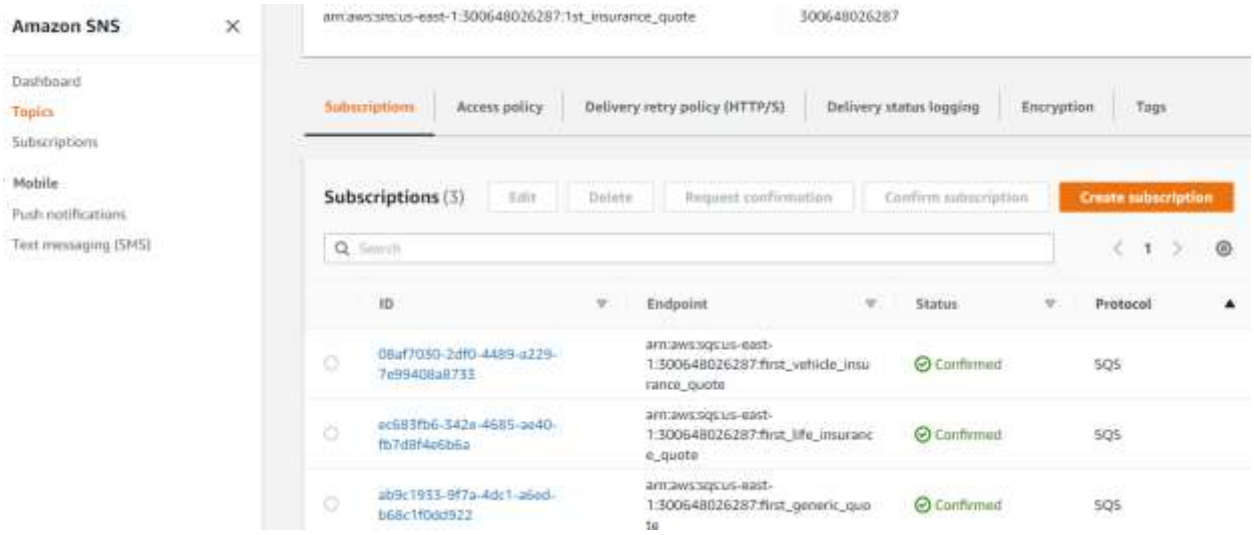

11: We will **Specify an Amazon SNS topic available for this queue** and click **Save**



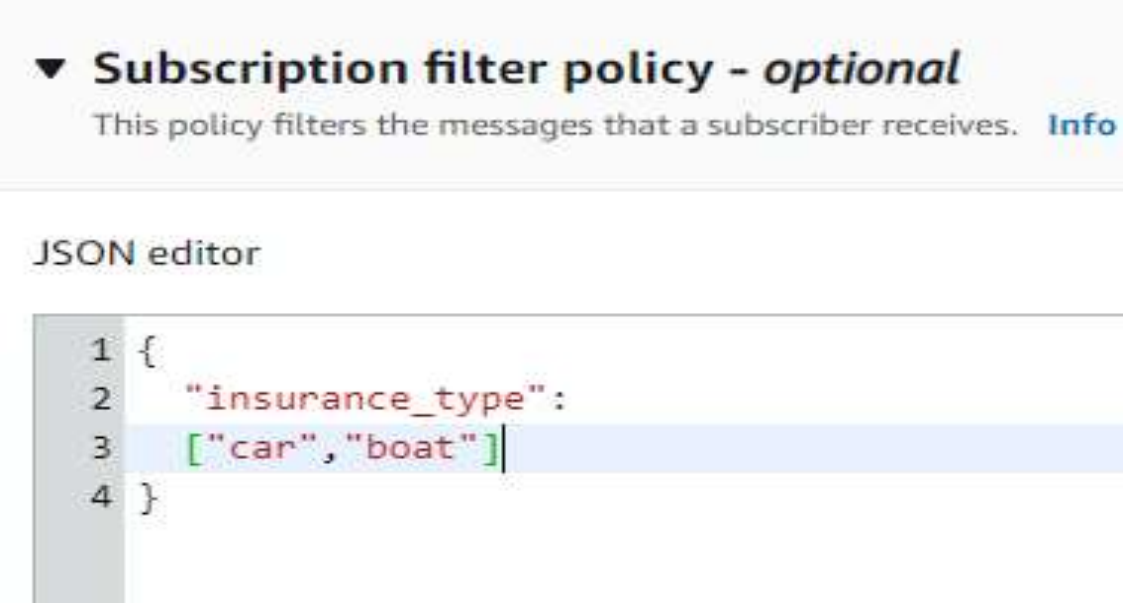

12: We have **subscribed** successfully to the topic.



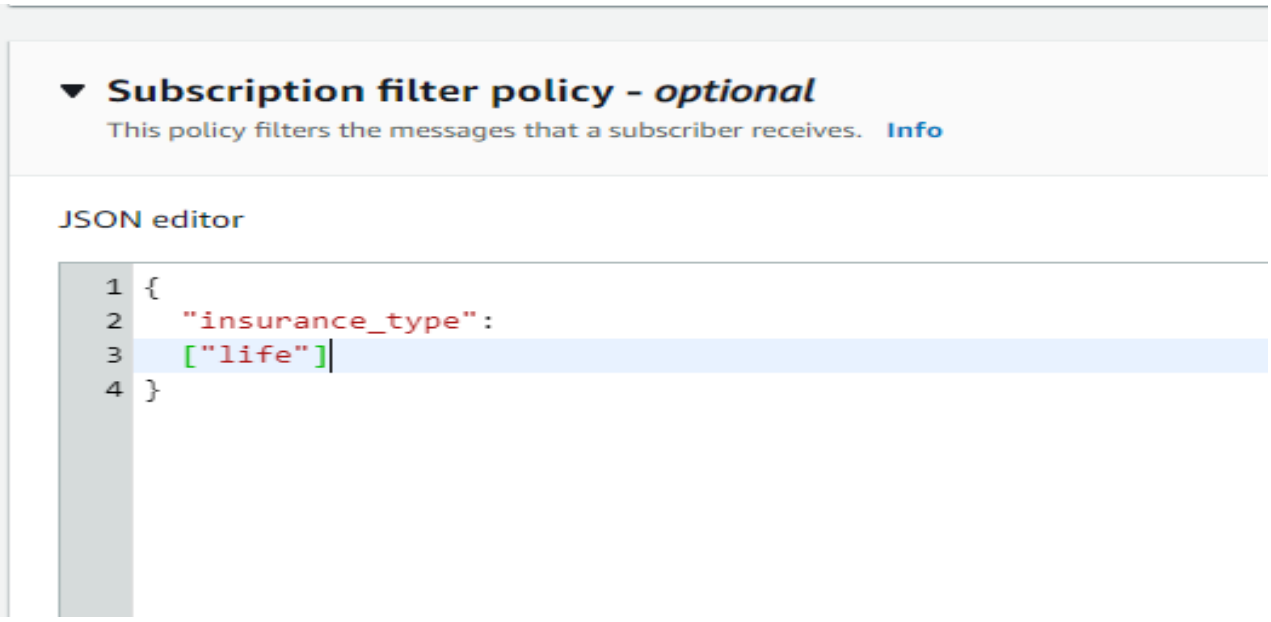

13: After connecting all the SNS topics we can see the subscription as below fig.



14: Now we will create **subscription filter policy** for each of the subscription . Click on edit and set the subscription policy with JASON code. (Below fig is for vehicle_insurance).

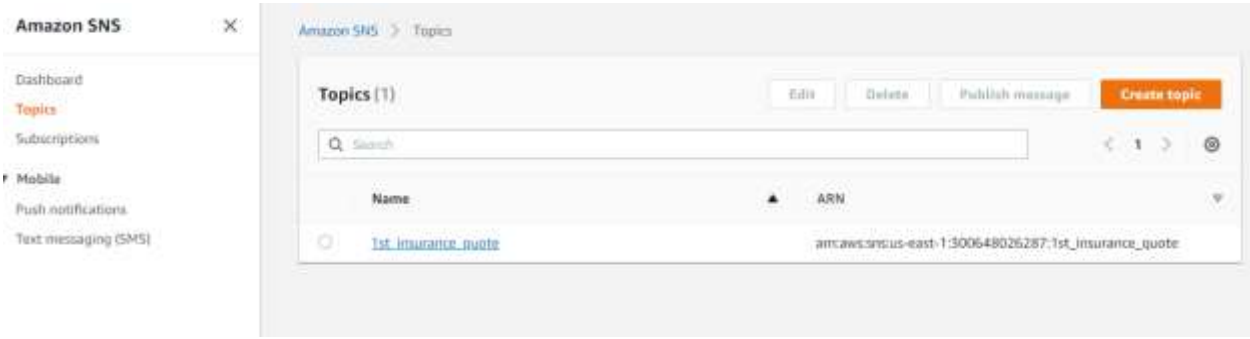


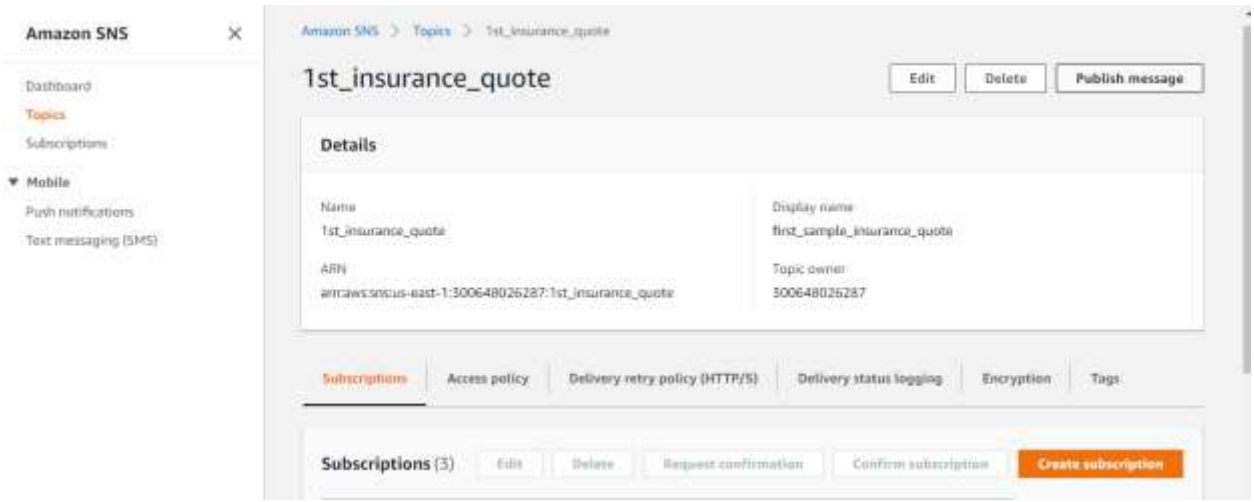

15: Similarly we created subscription policy for life_insurance.



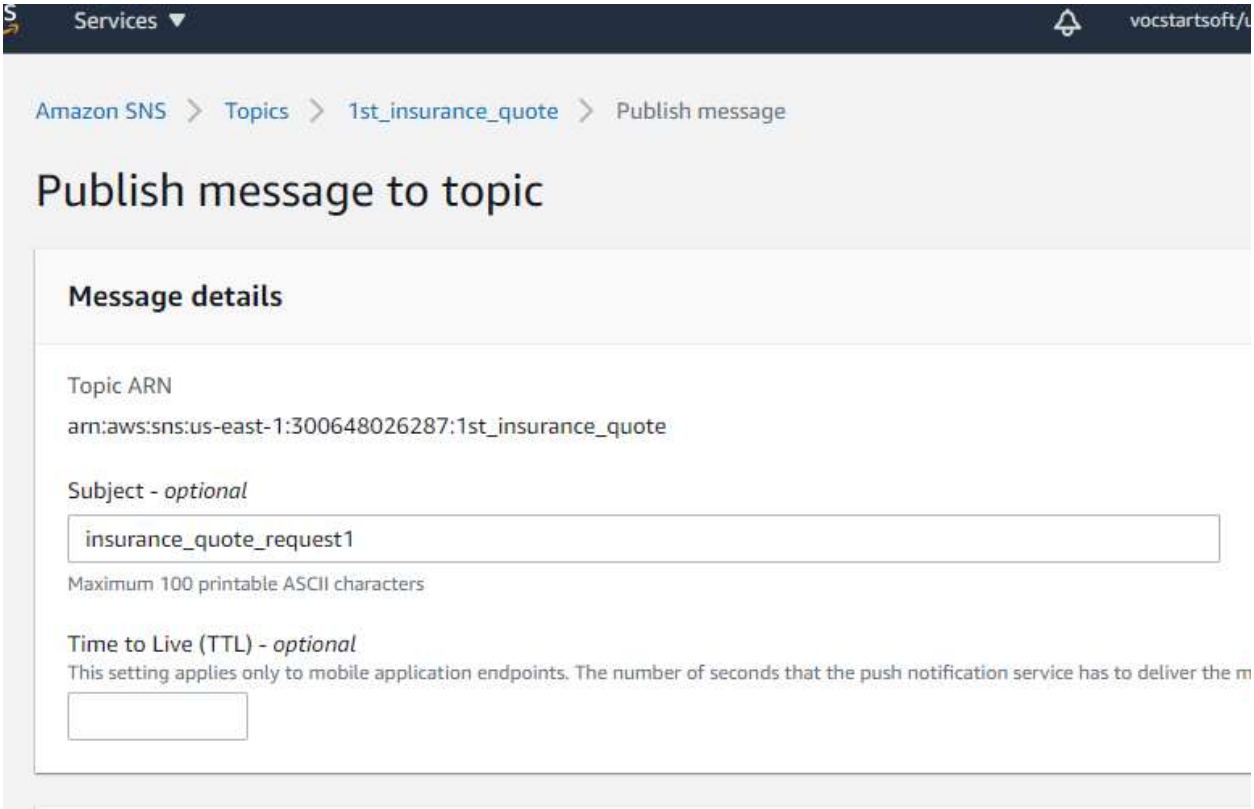

## Publishing Message:

16: For publishing message on the SQS page click on the **topic** and click on the **publish the message**.

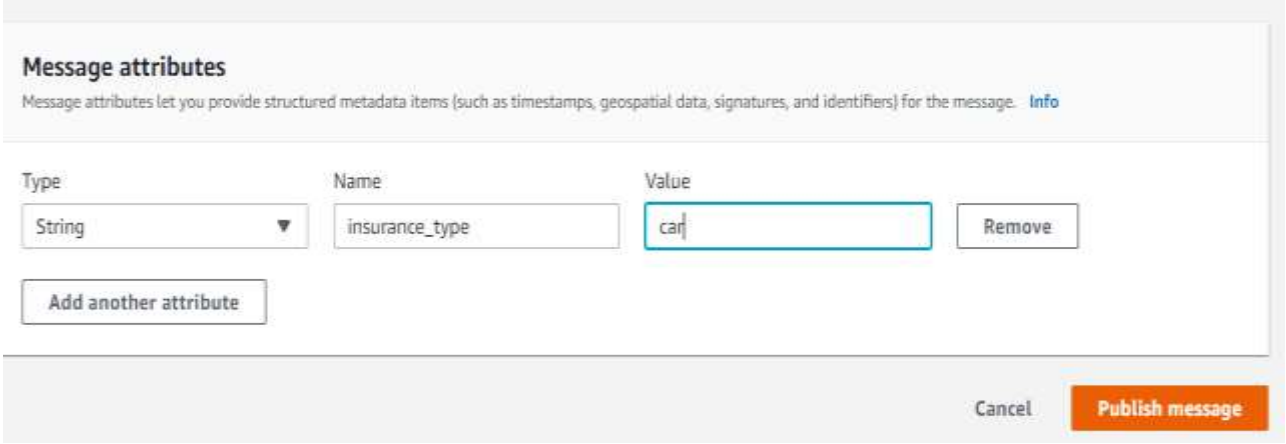





17: Then we have to provide the subject for the published message.

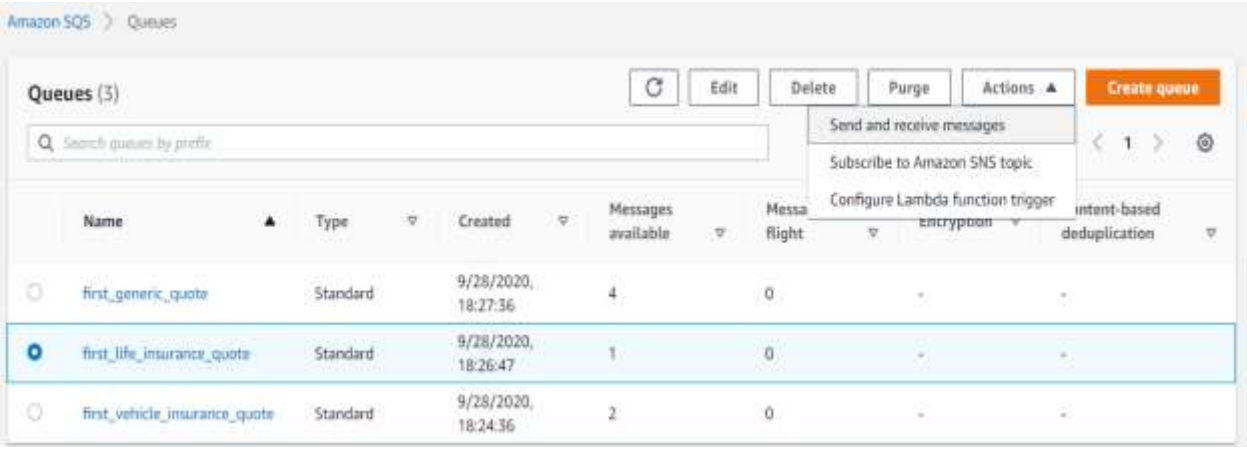


18: And we have to type the message and set the message attribute  and click **publish message**.

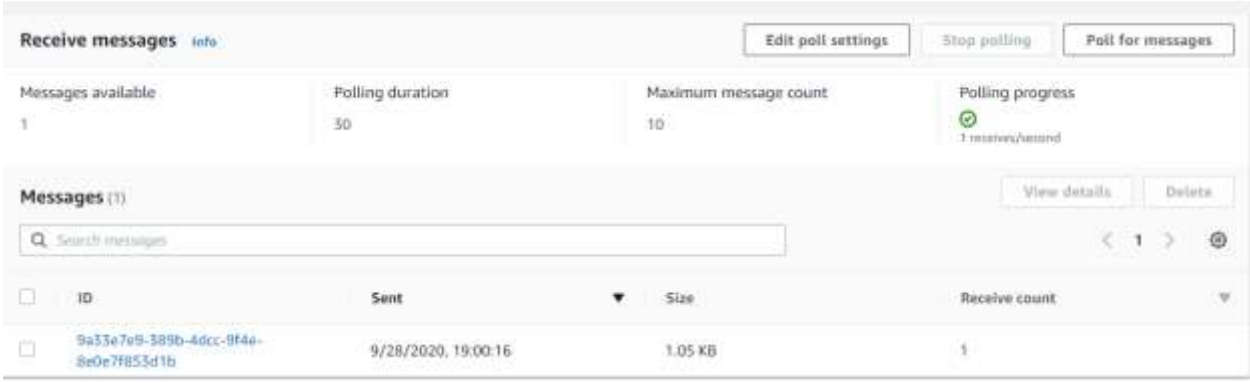




Similarly we have created for vehicle and generic insurance message with attribute and published the message.

19: Now we will go to the SQS console and verify if the messages are routed to the corresponding message. We can see on the below fig that there are values on the **“messages available”** , then we click on the action and select **“Send and receive messages**” .

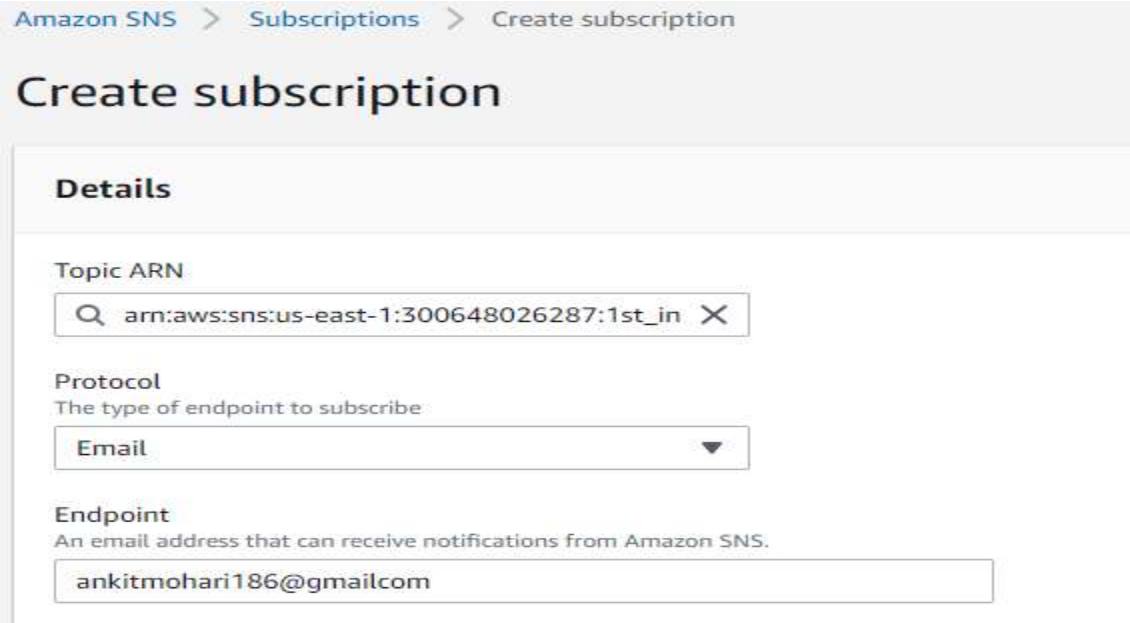


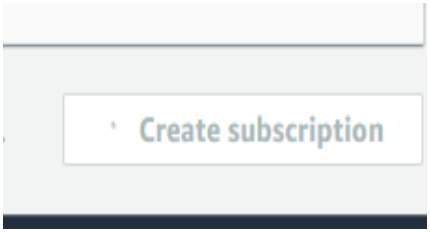

20: Click the **poll for messages**, we can see the receive count. Here we have shown that message come from the SNS to SQS.



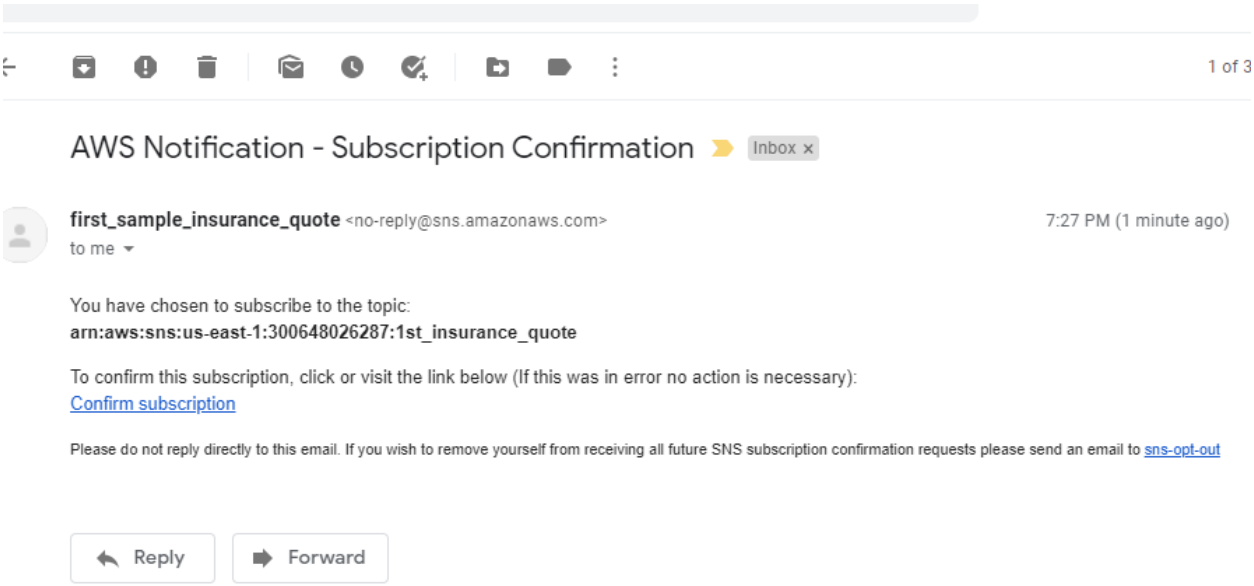
## Publishing message as a Notifications Services to E-mail:

21: To send notifications to the e-mail, first we have to create new subscription. Choose the Topic ARN(Amazon Resource Names) and select  the protocol as **E-mail,** provide the e-mail we want to send the notification on **the Endpoint.** And click on the **Create Subscription.**

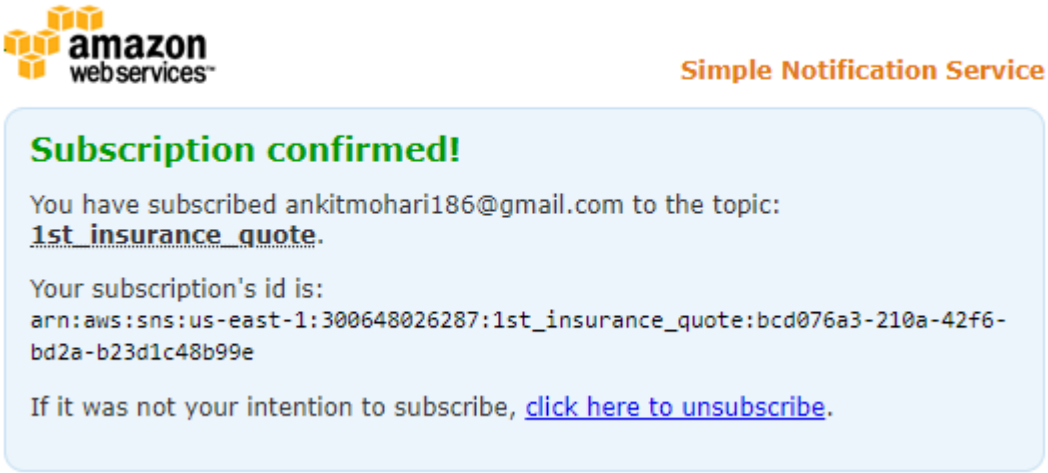


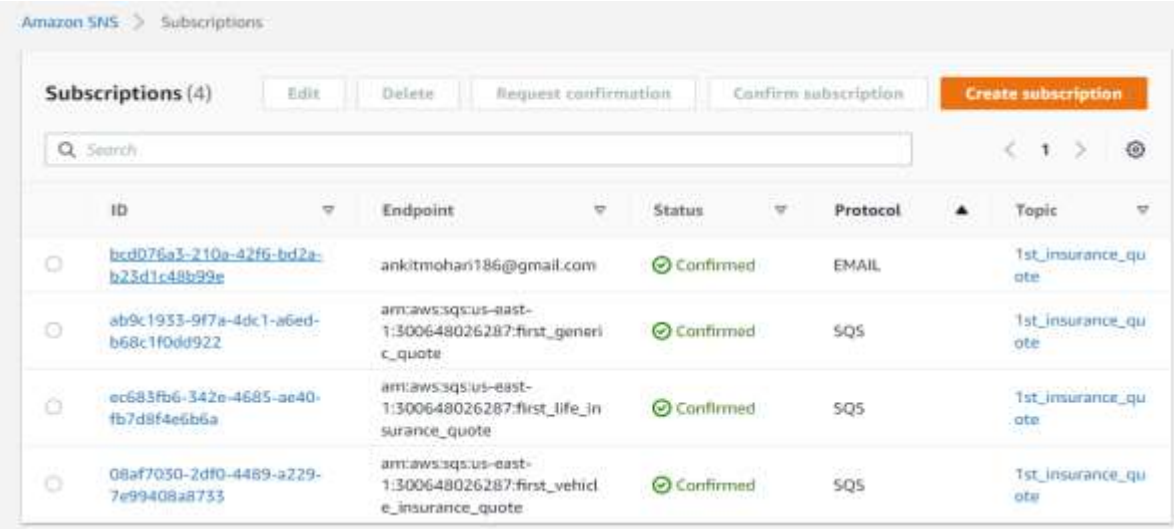
22: Now we will get a subscription request message.



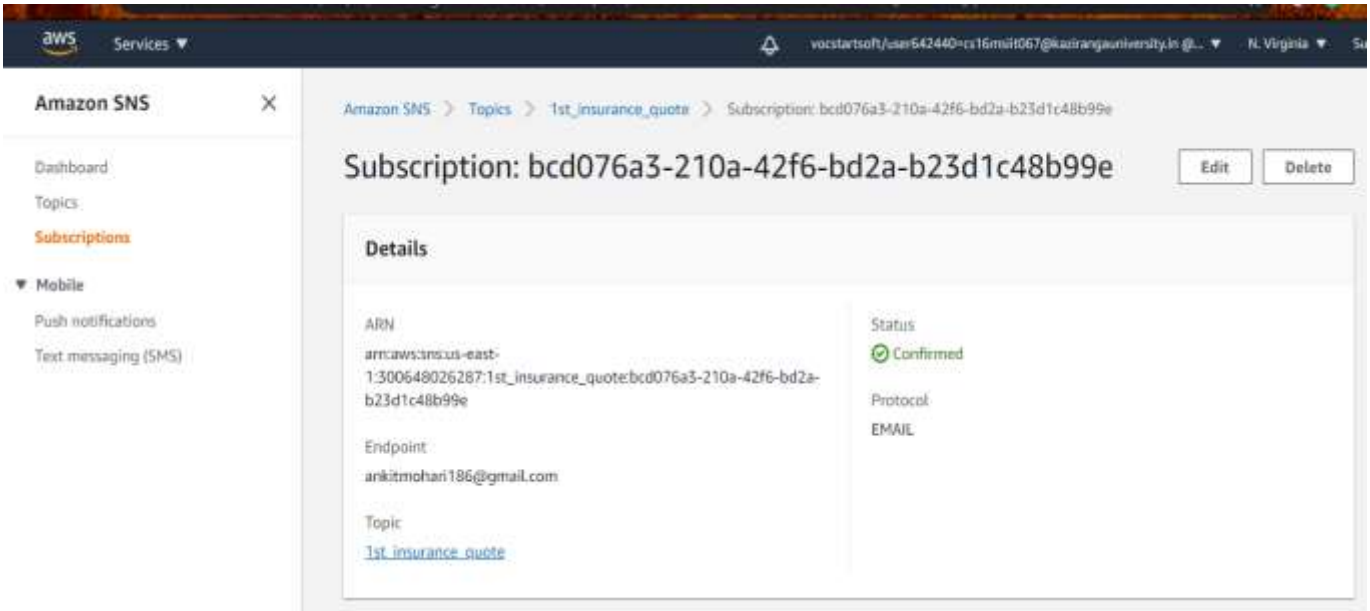

22: After we click on the Confirm subscription we will get the window displaying that “Subscription confirmed”.



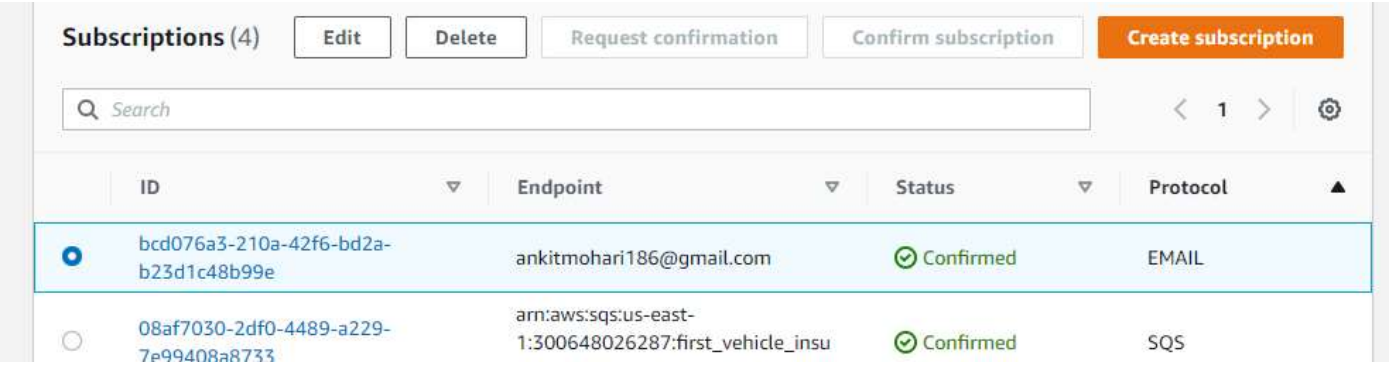
23: Now we can publish a message from the confirm subscription. Click on the **ID** of the confirm subscription **.**



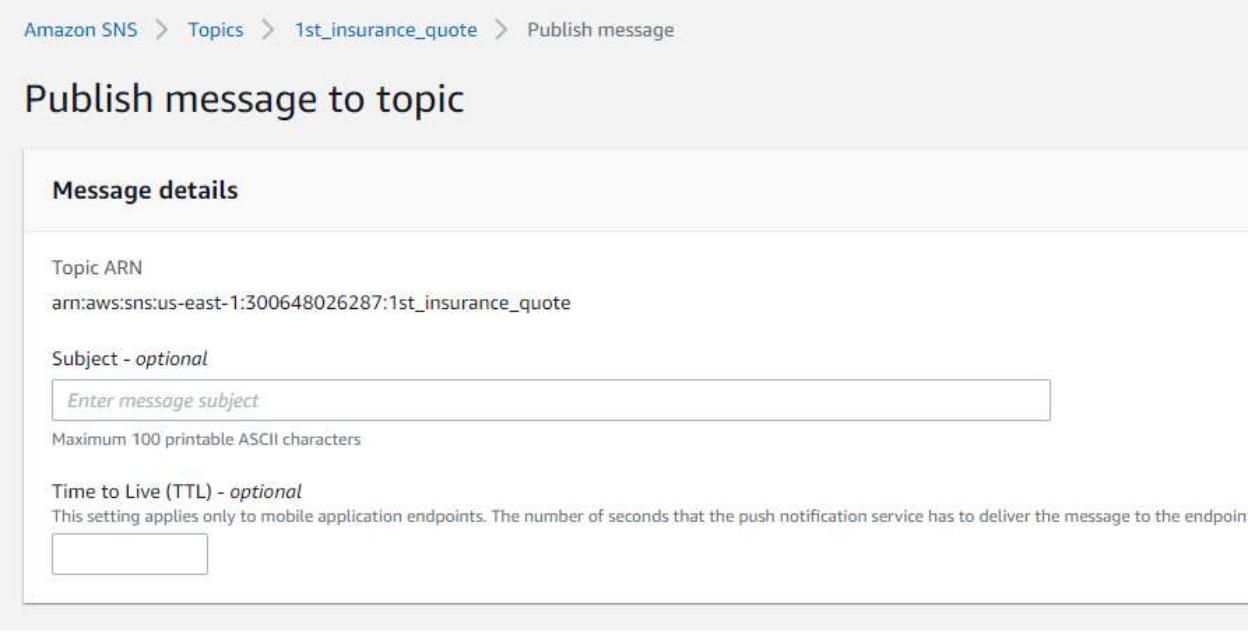

24: From particular subscription account click on the **Topic.**



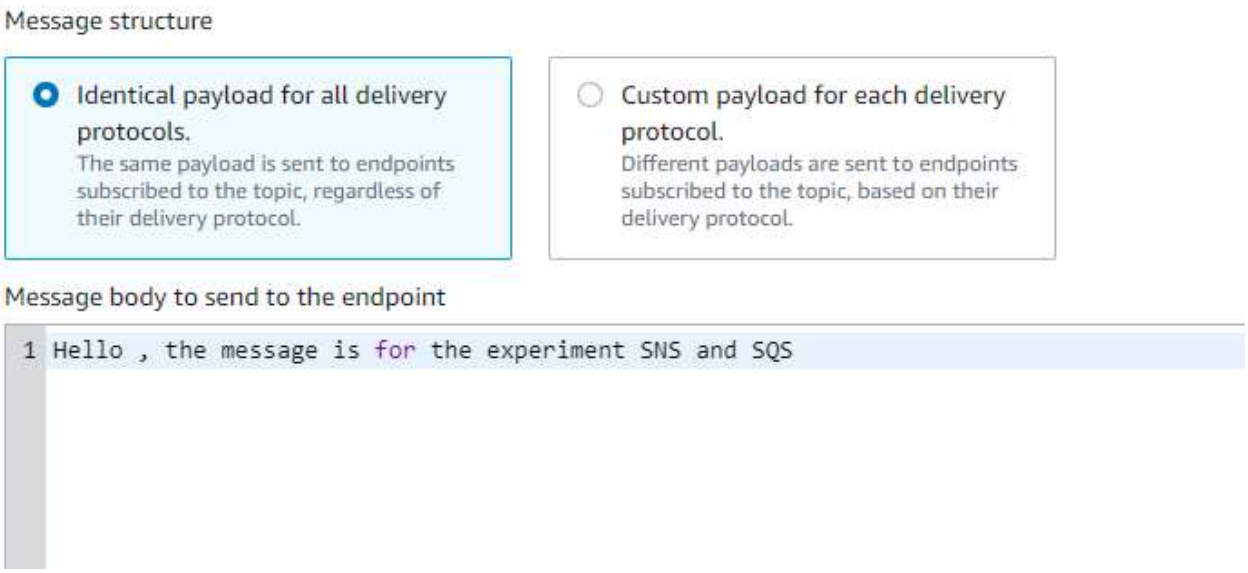
25: Select the ID and click on the **Publish message.**



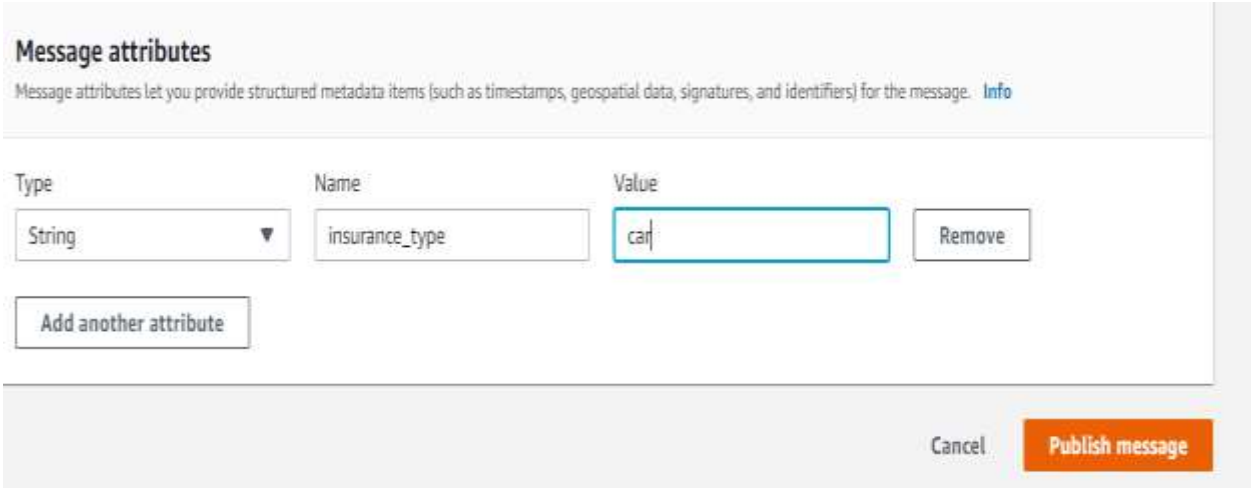

26: We have provide our subject.



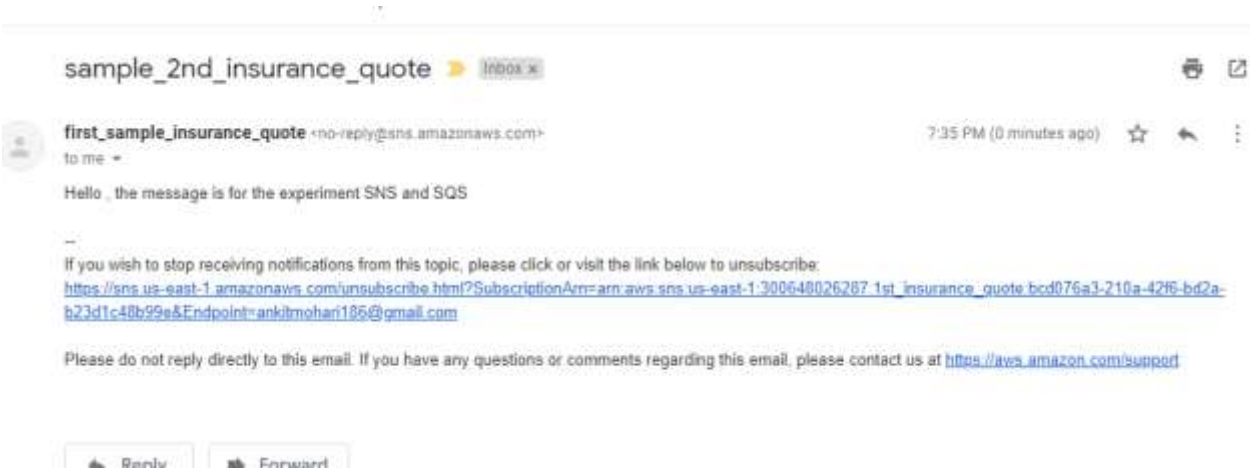
27: We will type the **message body .**




28: We will select the message attributes and click on the **publish message.**



29: We will get the published message on our confirmed e-mail




30: We have successfully send the published  message on our confirmed e-mail.