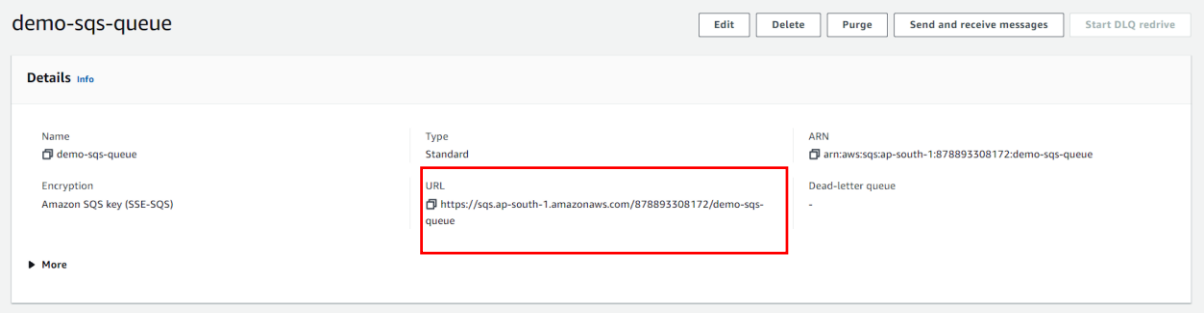
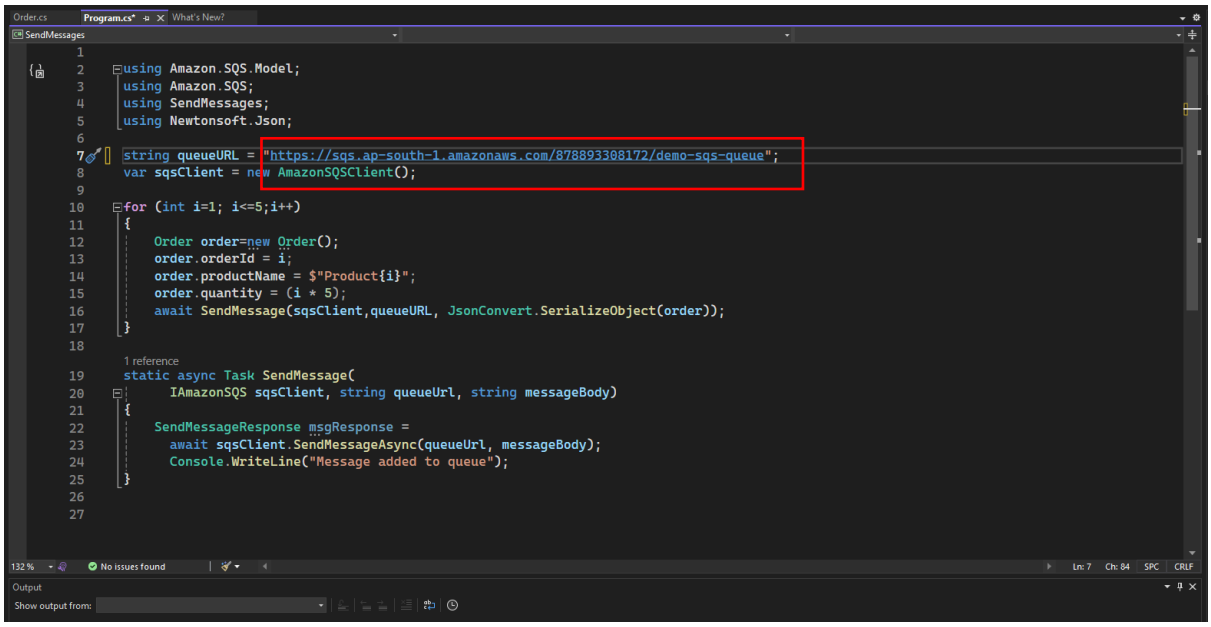
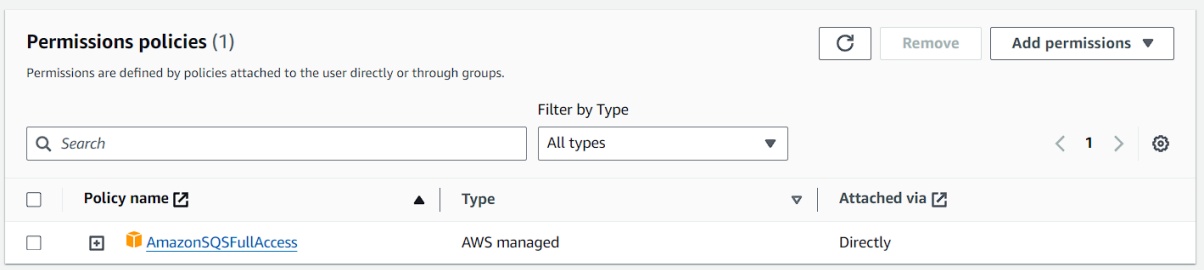
😊 Simple Queue Service

1. Now, in this lab, I want to give an example of a program that can send messages onto the queue.
2. You can get the program from GitHub.
3. This is a program in Dotnet that can go ahead and send messages onto an Amazon simple queue.
4. Now you will get the program in a zip file, so unzip it and open it in Visual Studio.
5. Then there you need to change the URL. You can get this URL from SQS.

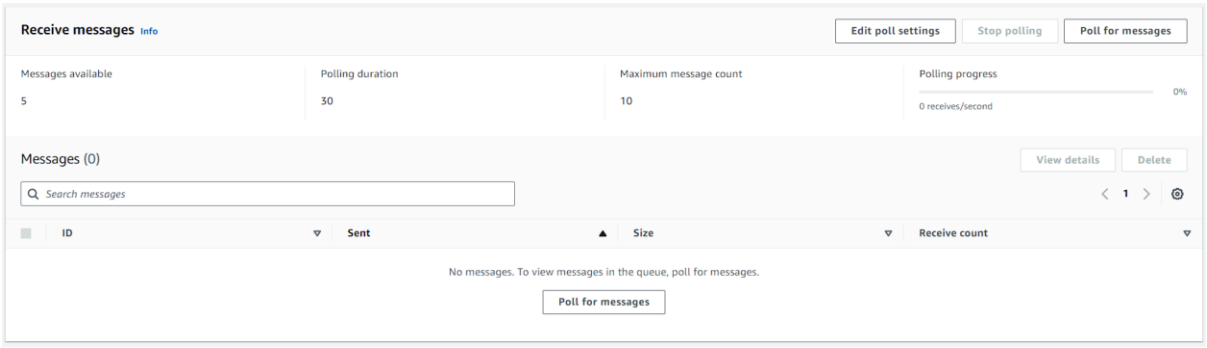




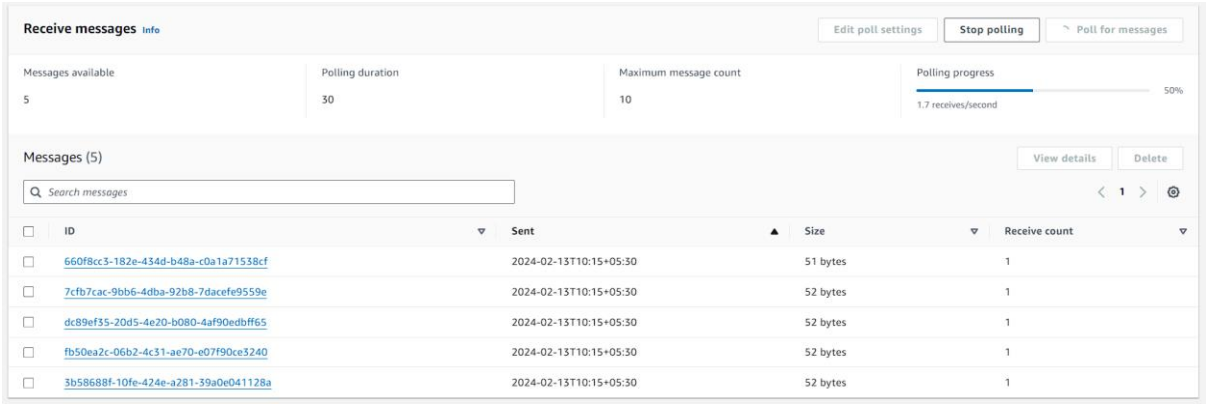
6. After that you need to create an IAM user.
7. In terms of permissions, I have assigned an inbuilt policy in AWS to allow full access onto the simple queue service. So, this will allow this user to interact with SQS.



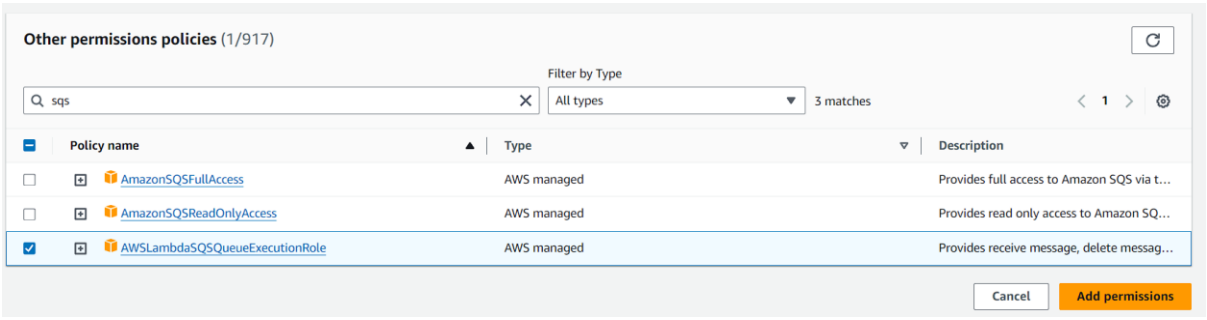
8. And then in the command prompt, when it comes on to the AWS CLI, I have configured the credentials so Visual Studio can actually go ahead and make use of these same credentials.
9. Once everything is setup go to visual studio and run the program in your local machine.
10. Now you will go back to your queue and click on send and receive messages. There you will see that you have 5 messages available.
11. Now click on Poll for messages.



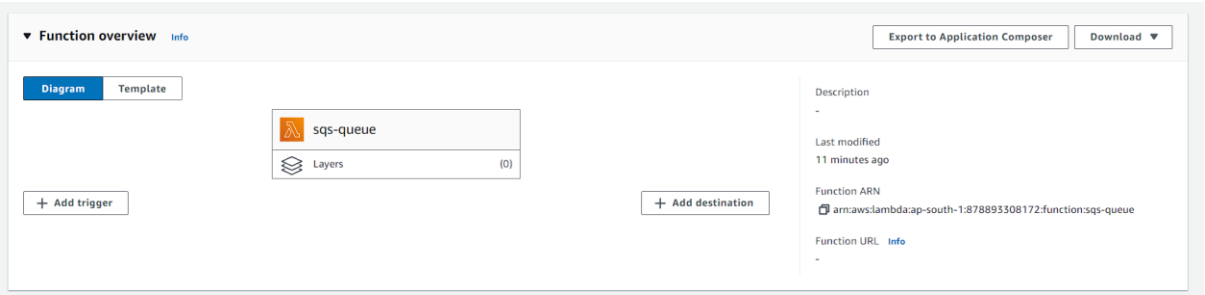
12. You will see all 5 IDs for messages. You can open them and see what's inside.

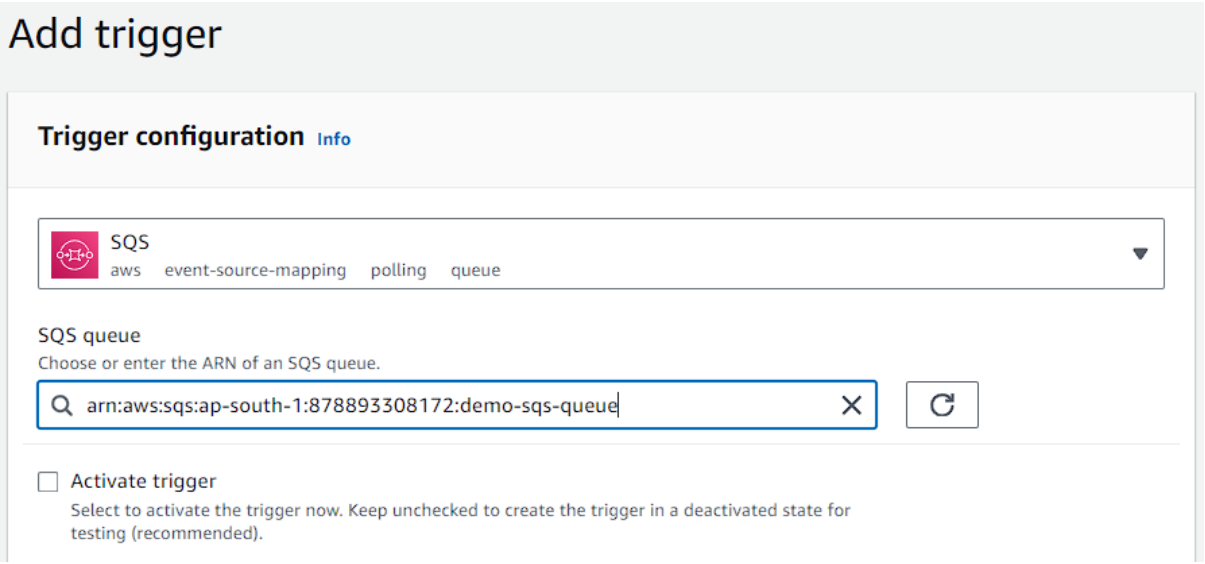


13. Now you are going to create a Lambda function which is going to receive the messages from queue.
14. So, navigate to Lambda and create a function, choose the runtime environment as Python.
15. Now before that go to IAM and add permission for SQS in your role.



16. Now in your function you need to add a trigger as SQS. Then you need to change the code.





17. Once the trigger is added now you need to change the code. Then deploy the code.

```
import json

def lambda_handler(event, context):
    if 'Records' in event:
        for message in event['Records']:
            order_details = json.loads(message['body'])
            order_id = order_details['orderId']
            product_name = order_details['productName']
            quantity = order_details['quantity']

            print(order_id)
            print(product_name)
            print(quantity)
    else:
        print("No records found in the event.")
```

Tools Window **Test** Deploy

lambda_function × Environment Var × Execution results ×

```python
import json

def lambda_handler(event, context):
    if 'Records' in event:
        for message in event['Records']:
            order_details = json.loads(message['body'])
            order_id = order_details['orderId']
            product_name = order_details['productName']
            quantity = order_details['quantity']

            print(order_id)
            print(product_name)
            print(quantity)
    else:
        print("No records found in the event.")

```

18. After that you need to go to CloudWatch. Then open the log groups.
19. In there you will see multiple logs generated for you SQS. Now you can see your messages in your logs instead of SQS.