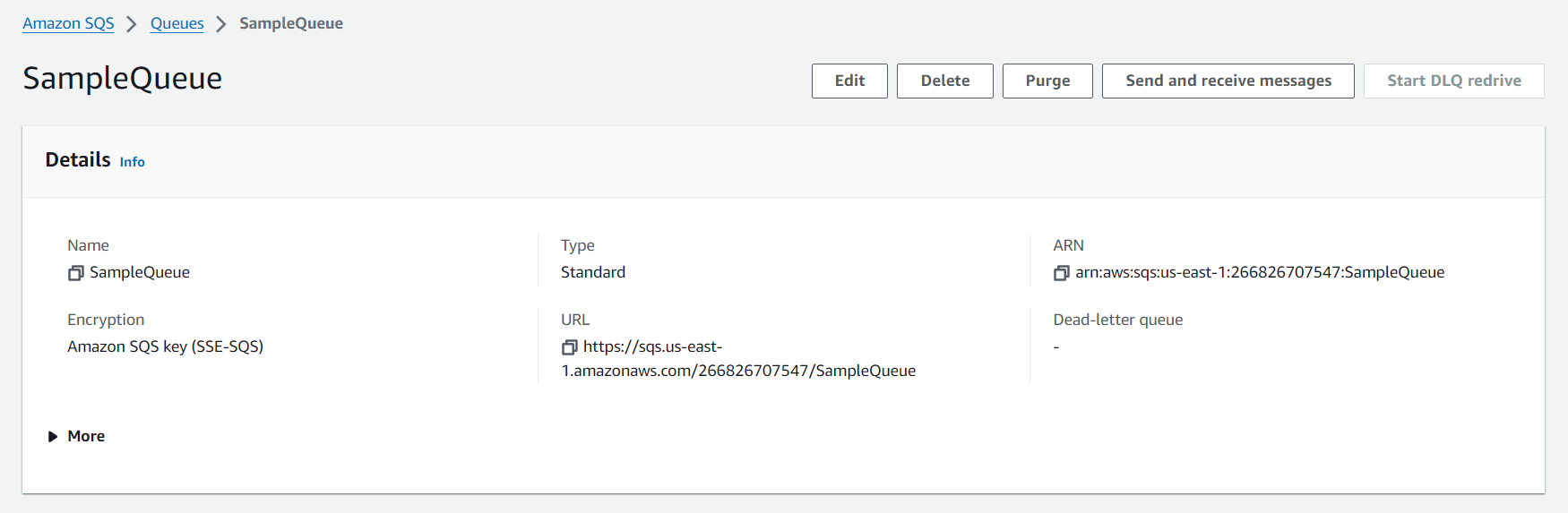
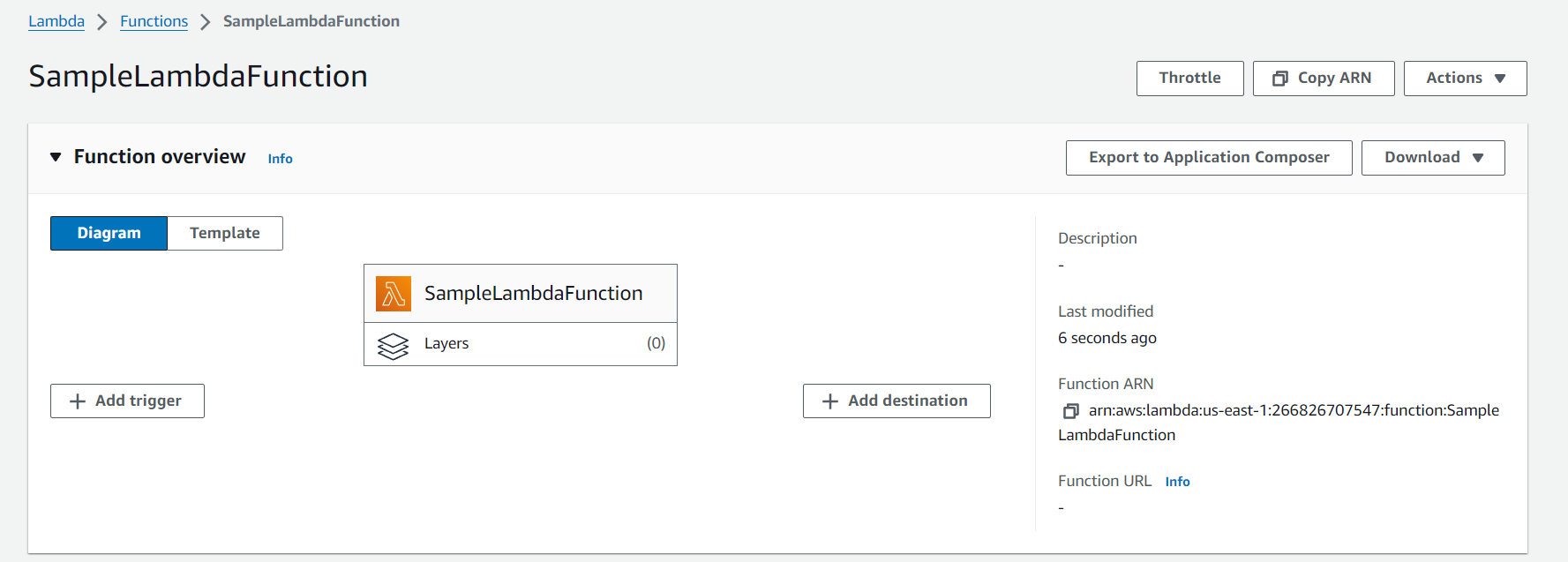
**Step 1: Create an SQS Queue**

1. Navigate to the AWS SQS console.
2. Create a new standard queue named SampleQueue.
3. Configure any necessary settings and permissions.



**Step 2: Develop AWS Lambda Function**

1. Open the Lambda service.
2. Click on Create function.
3. Choose Author from scratch.
4. Enter a function name, e.g., SampleLambdaFunction.
5. Select Python 3.x as the runtime.
6. Choose a role that has permissions to execute the Lambda function. You can select an existing role or create a new role with basic Lambda permissions.



A Python-based Lambda function was written to process the messages coming from the SQS queue. Below is the Lambda function code:

**python**

def lambda\_handler(event, context):

# Loop through each SQS message

for record in event['Records']:

# Process each SQS message

message\_body = record['body']

print(f"Received message: {message\_body}")

return {

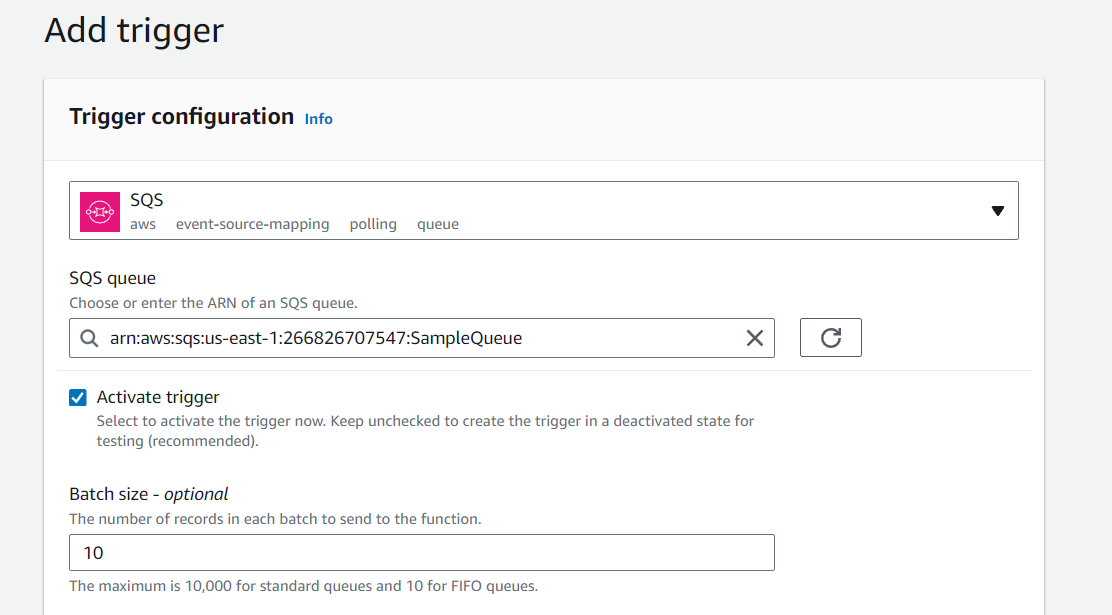
'statusCode': 200,

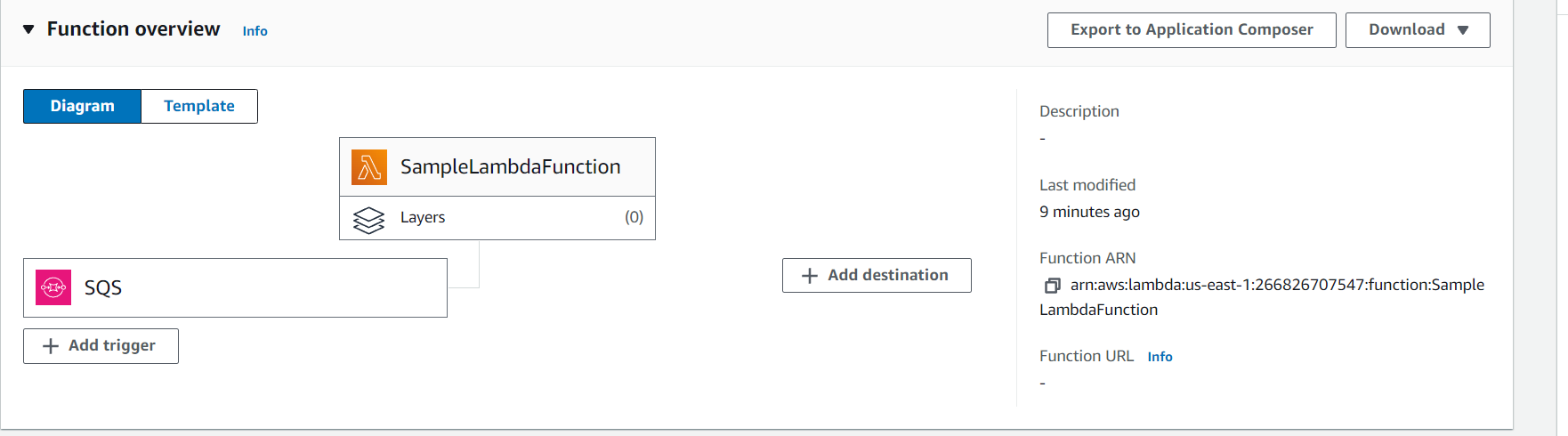
'body': json.dumps('Message processed successfully')

}

**Step 3: Configure Lambda Trigger**

1. Go back to the **Lambda** function.
2. In the **Function overview** section, click **Add trigger**.
3. Choose **SQS** from the list of triggers.
4. Select the SQS queue you created (e.g., SampleQueue).
5. Click **Add** to link the SQS queue as a trigger for the Lambda function.





**Step 4: Test the Solution**

1. Send a message to the SQS queue through the AWS Management Console (e.g., {"action": "test message"}).
2. The Lambda function is triggered and processes the message.
3. Logs of the processed message are visible in AWS CloudWatch.

**7. Testing and Validation**

The system was tested by sending messages to the SQS queue. The AWS Lambda function processed the messages successfully, confirming that the event-driven architecture was functioning as expected. CloudWatch Logs were used to verify that the Lambda function was triggered appropriately and executed correctly upon receiving messages.

**Test Results:**

* **Input Message:** {"action": "test message"}
* **Lambda Output:** Received message: {"action": "test message"}
* **Status:** Success

