**📌 Steps: Create Loosely Coupled Services with Amazon SQS, Lambda Trigger, and CloudWatch**

**1️⃣ Create an SQS Queue**

* Go to AWS Console → Search SQS
* Click **Create Queue**
* Choose **Standard Queue**
* Enter **Queue Name**: HelloQueue
* Click **Create Queue**

**2️⃣ Create Consumer Lambda Function Using SQS Blueprint**

* Go to AWS Console → Search Lambda
* Click **Create Function**
* Choose **Use a blueprint**
* Search and select sqs-poller blueprint
* Enter Function Name: HelloConsumer
* Choose or create an execution role with AmazonSQSFullAccess and CloudWatchLogsFullAccess permissions
* In **SQS trigger configuration**, select your HelloQueue
* Click **Create Function**

**3️⃣ Add SQS Trigger to Lambda (if not done in step 2)**

* Go to **HelloQueue → Queue Actions → Subscribe Lambda function**
* Select HelloConsumer
* Click **Save**
* Confirm and grant trigger permission when prompted

**4️⃣ Create Producer Lambda Function**

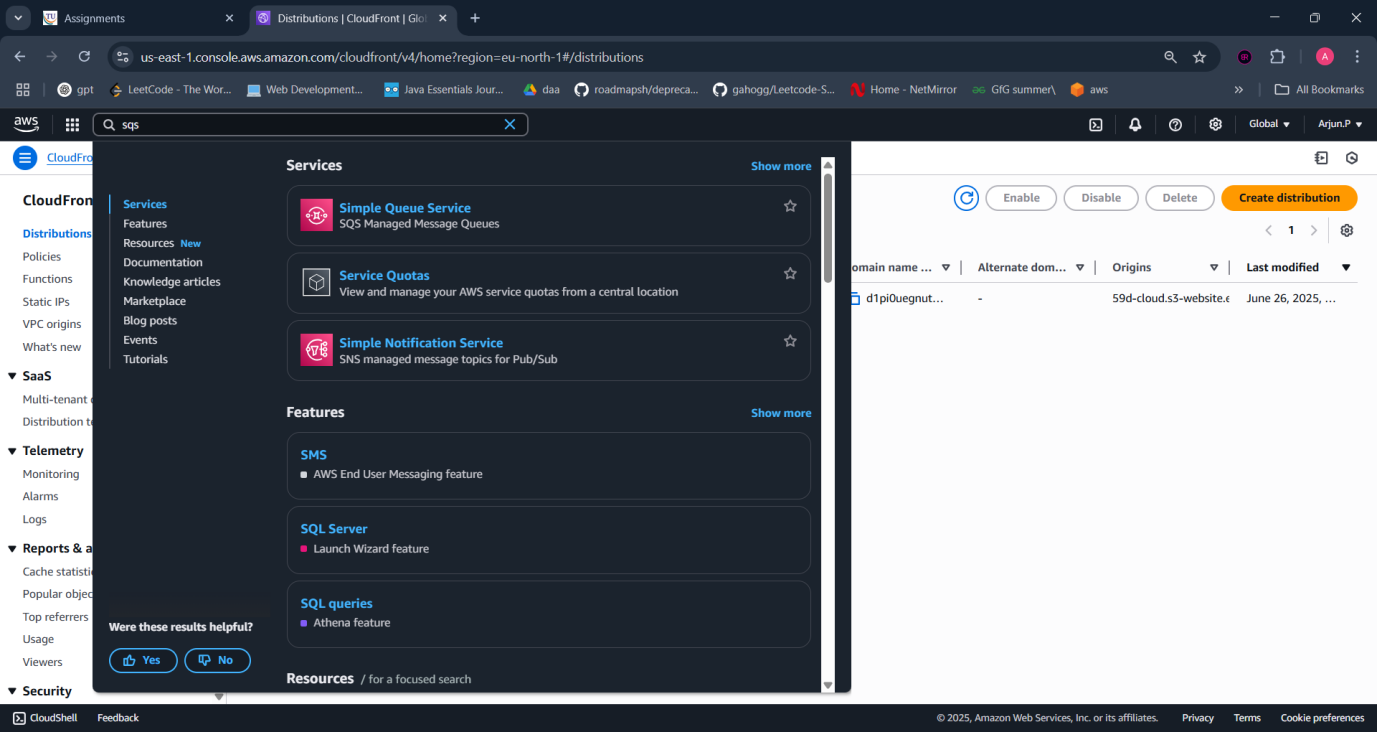
* Go to Lambda → **Create Function**
* Name: HelloProducer
* Runtime: Python 3.x (or Node.js)
* Click **Create Function**
* Add code to send a message to HelloQueue
* Attach AmazonSQSFullAccess policy to this Lambda’s execution role
* Deploy the function

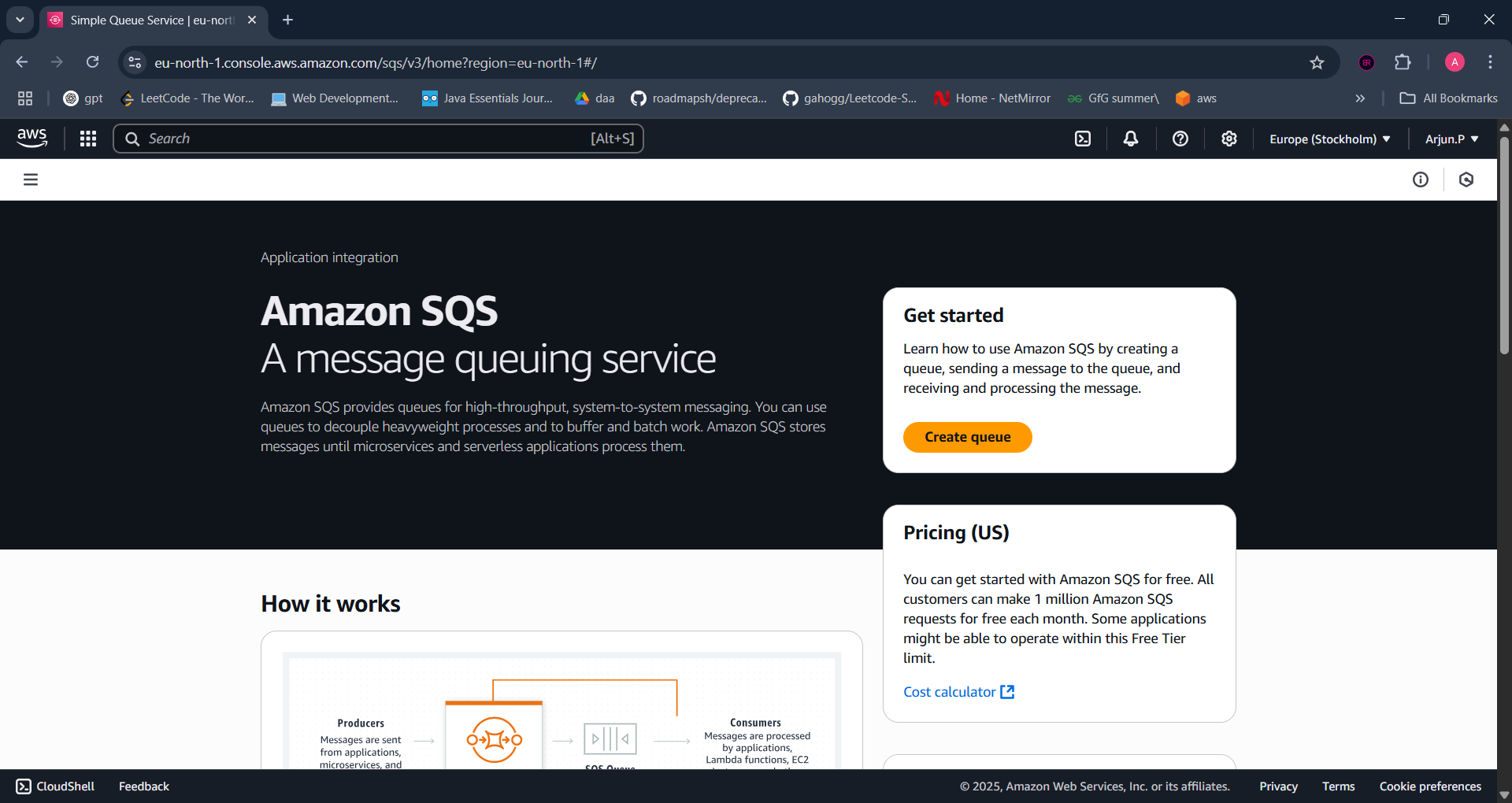
**5️⃣ Test the Setup**

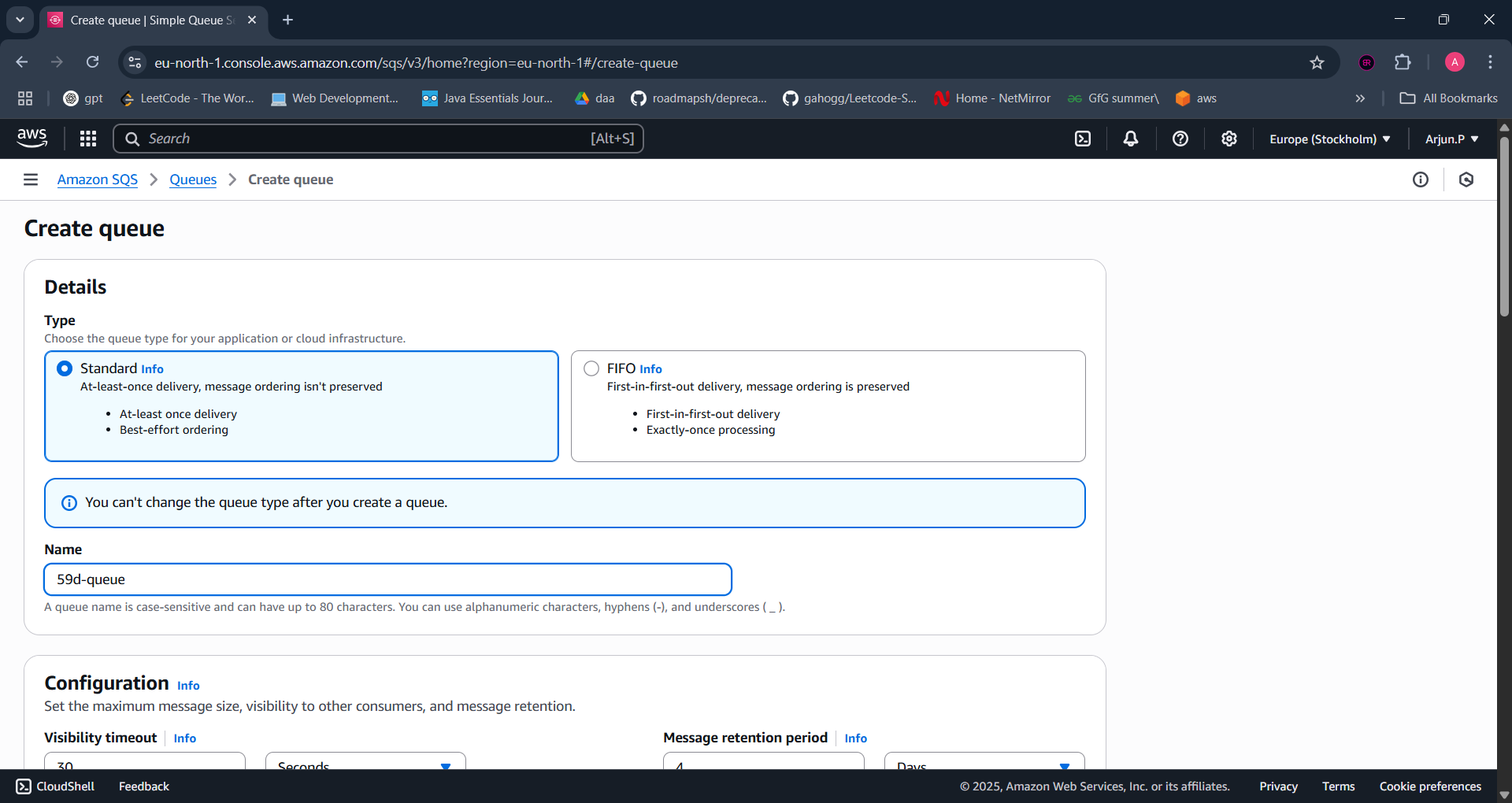
* Run **HelloProducer** Lambda using a test event to send a message to HelloQueue
* Lambda will push message to the queue

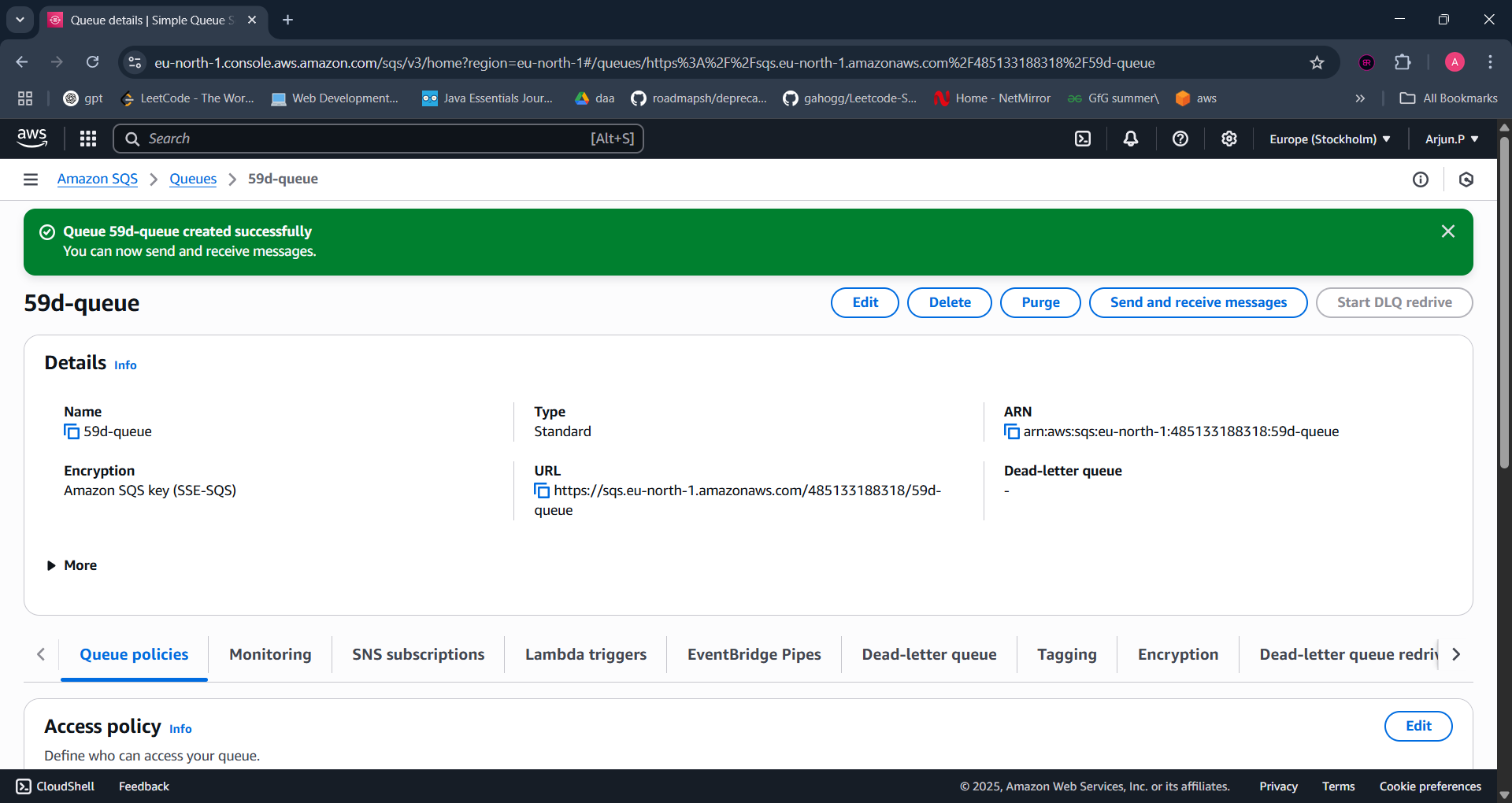
**6️⃣ Monitor in CloudWatch**

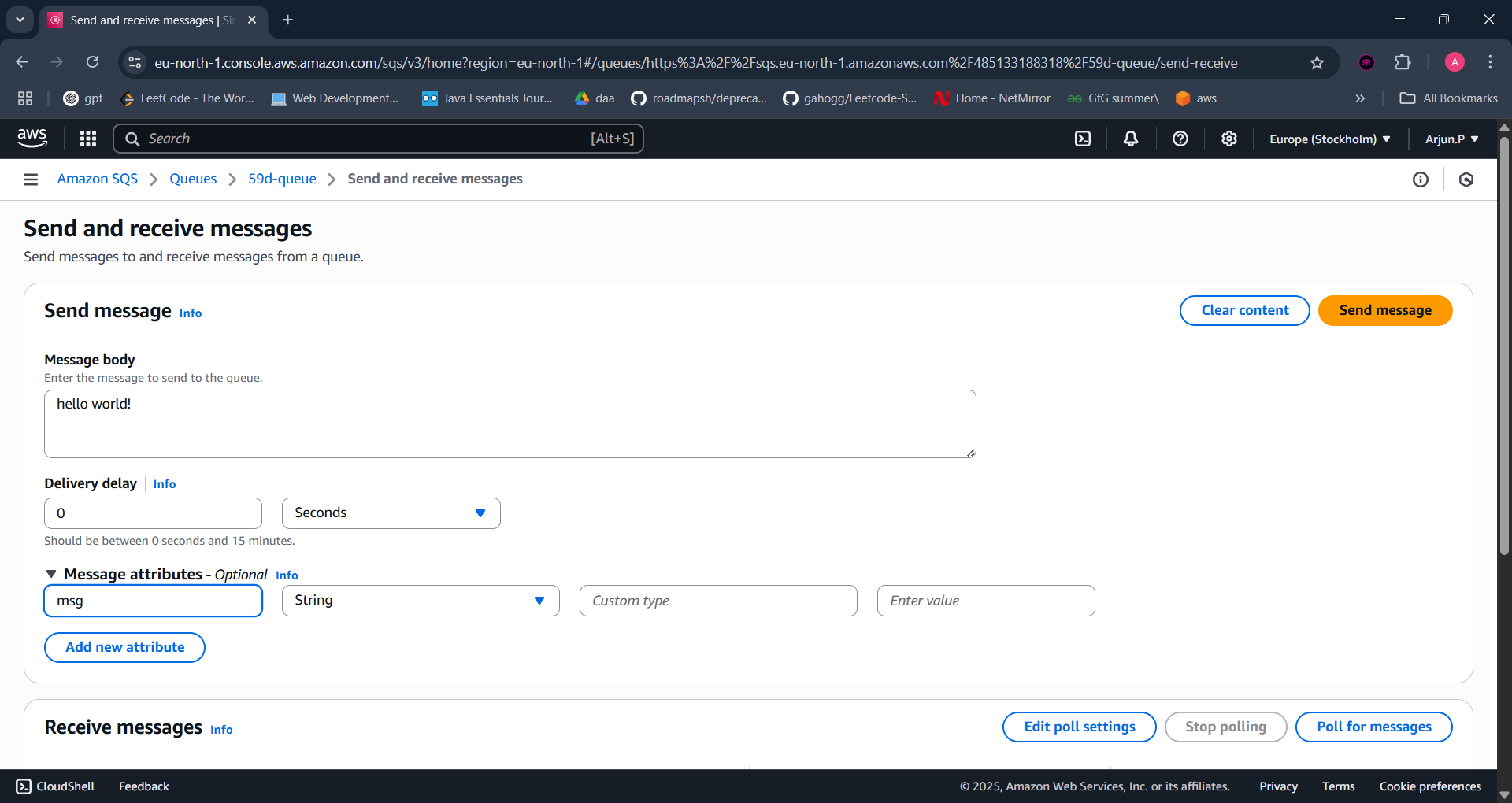
* Go to CloudWatch Logs
* Check logs for HelloConsumer Lambda
* Verify message received and processed logs





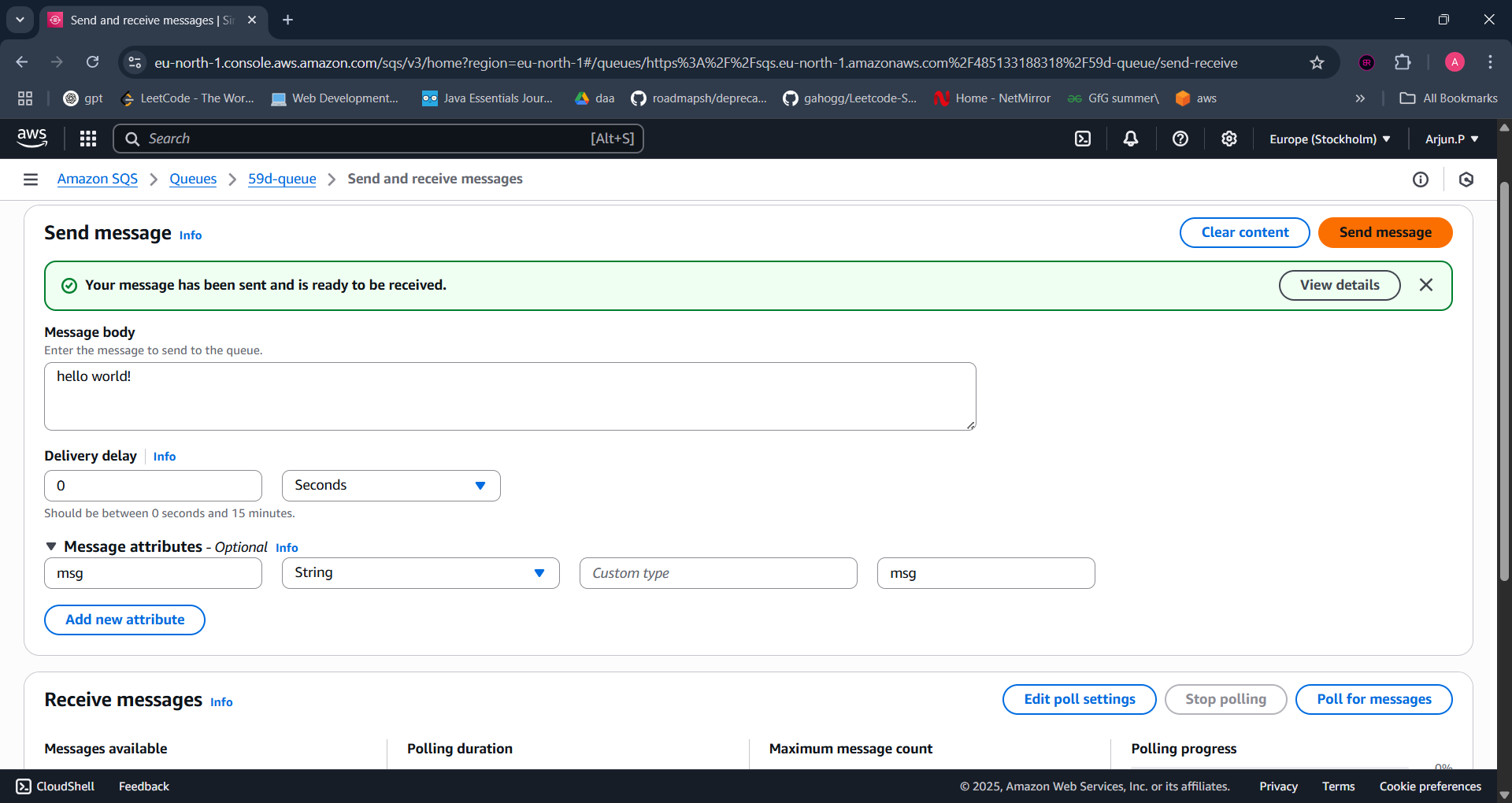


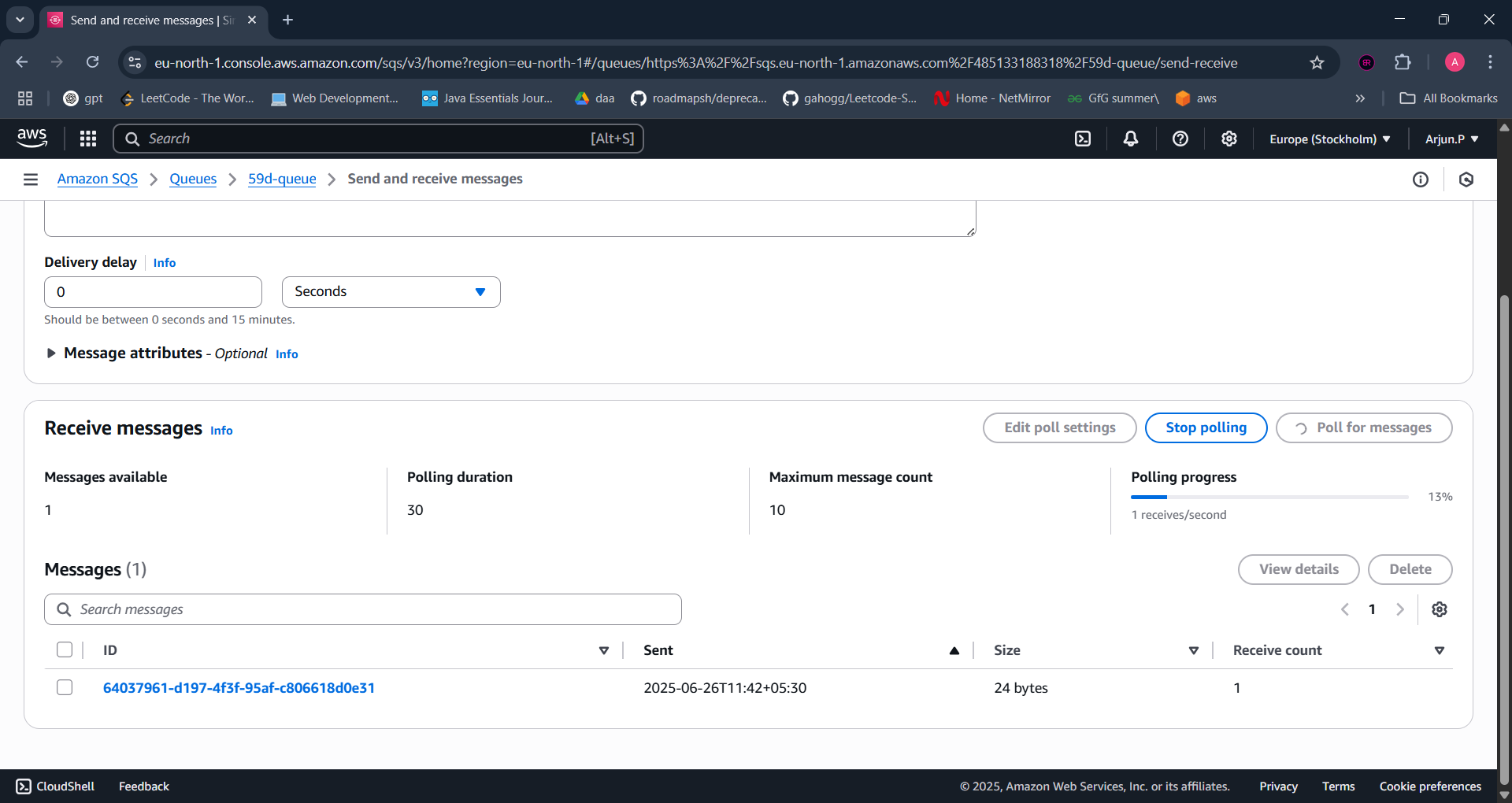


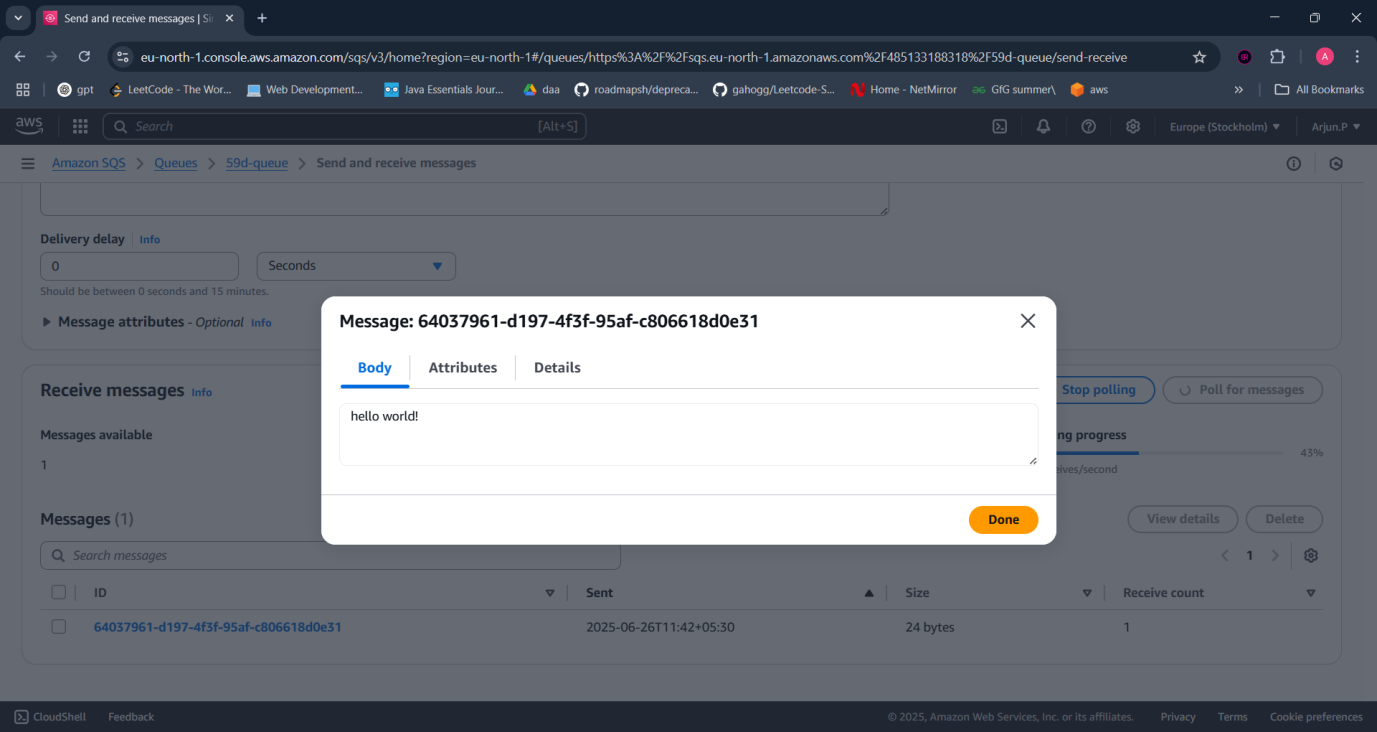


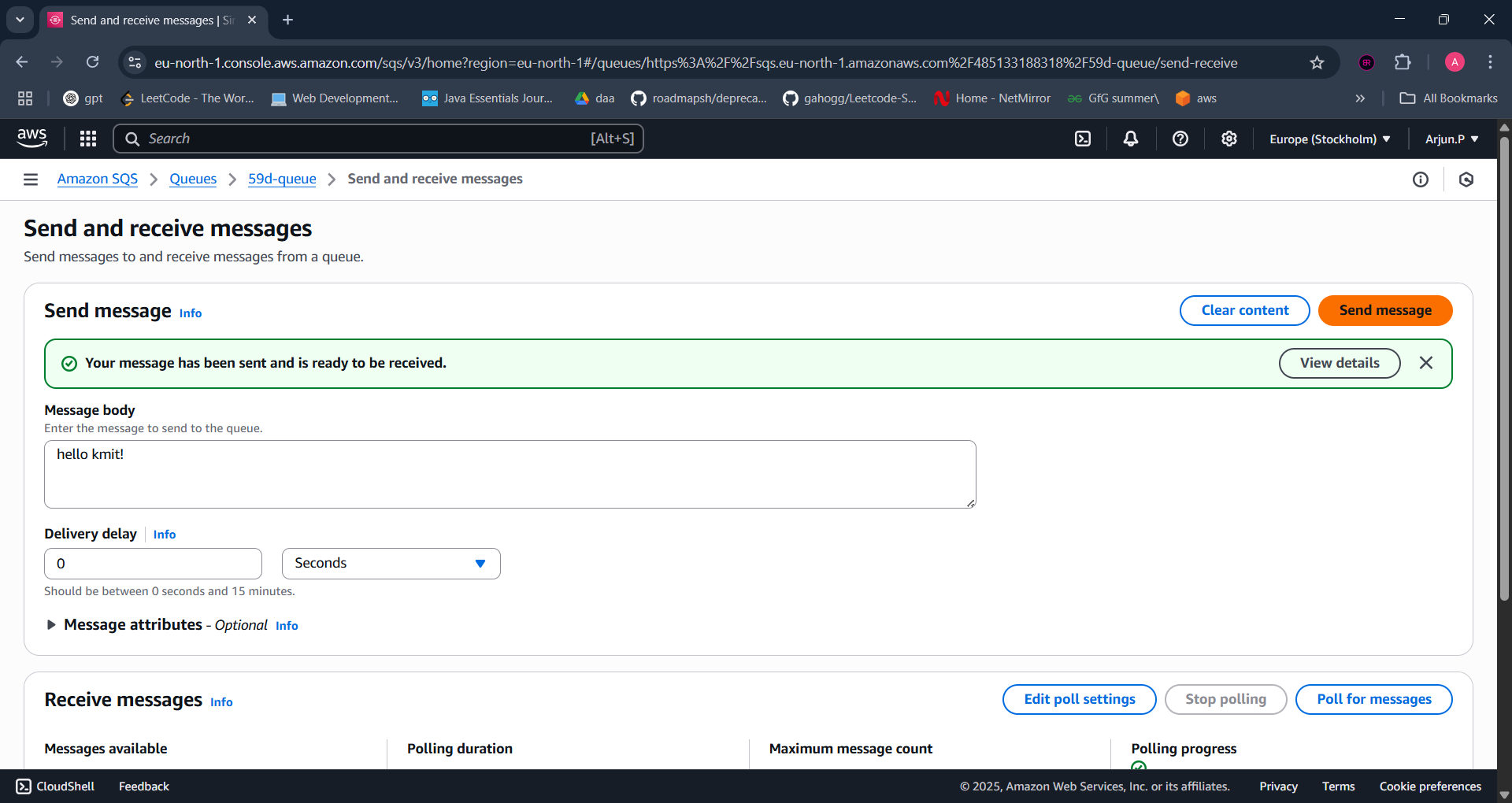
A screenshot of a computer

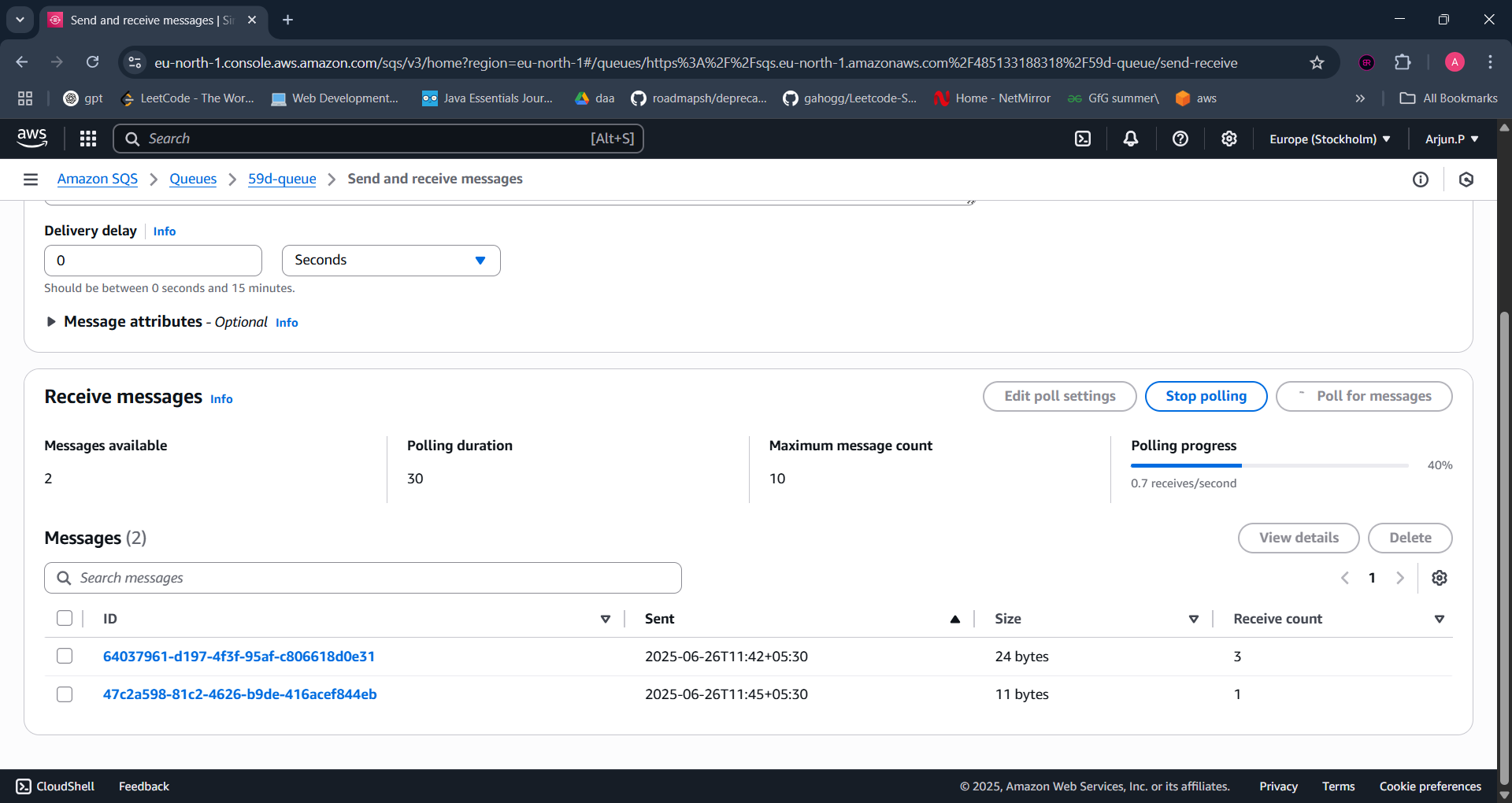
AI-generated content may be incorrect.

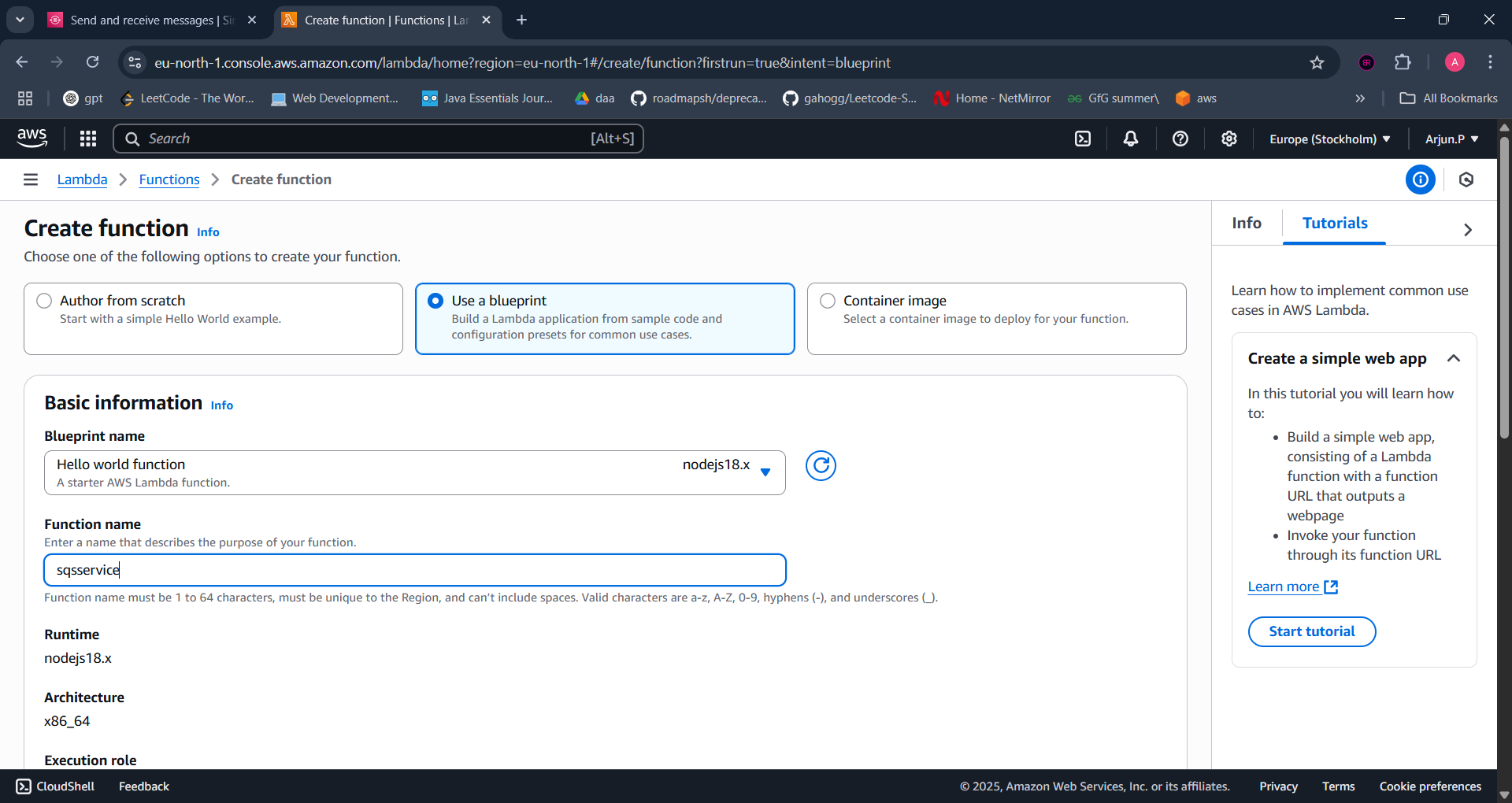


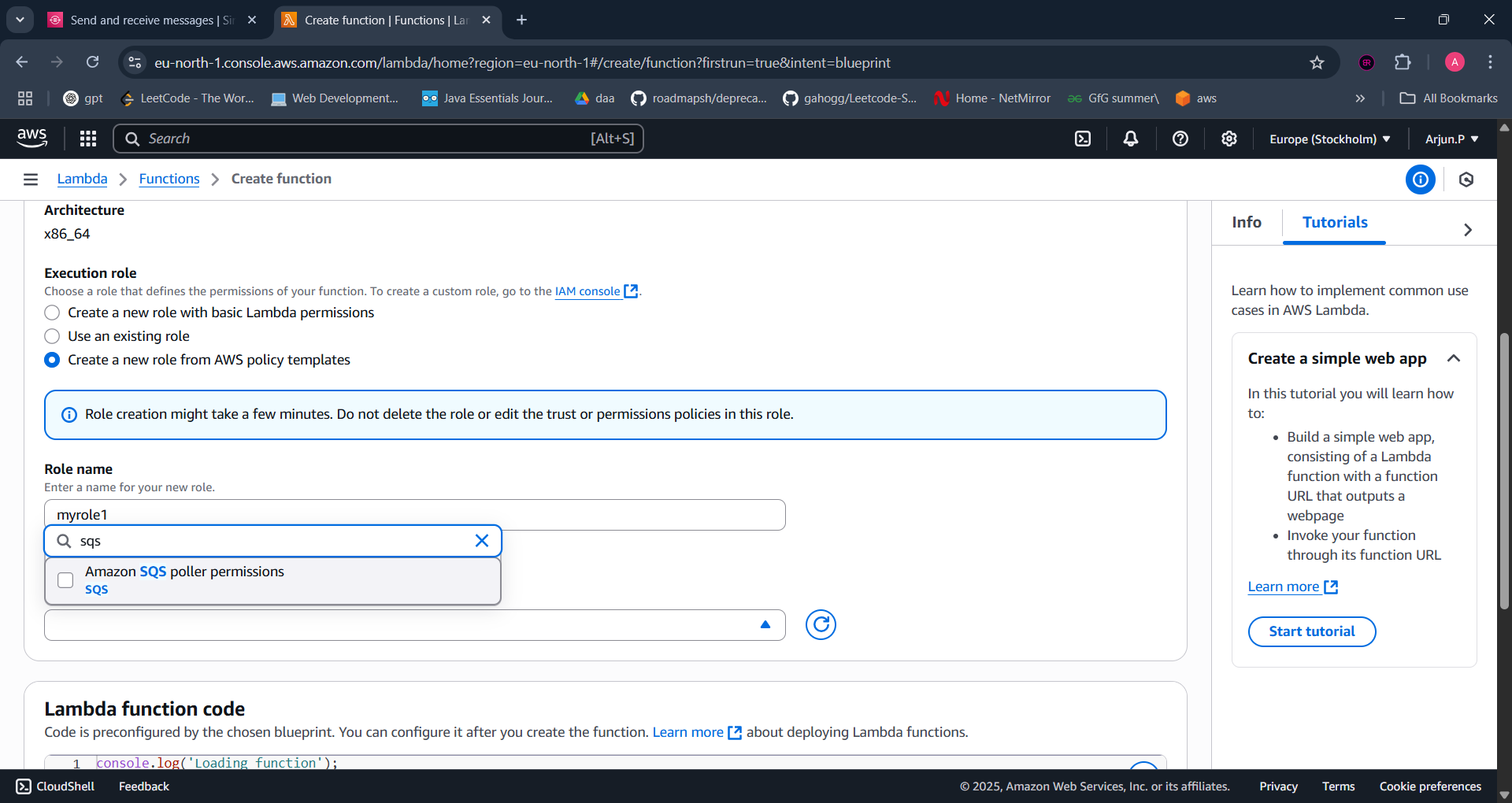


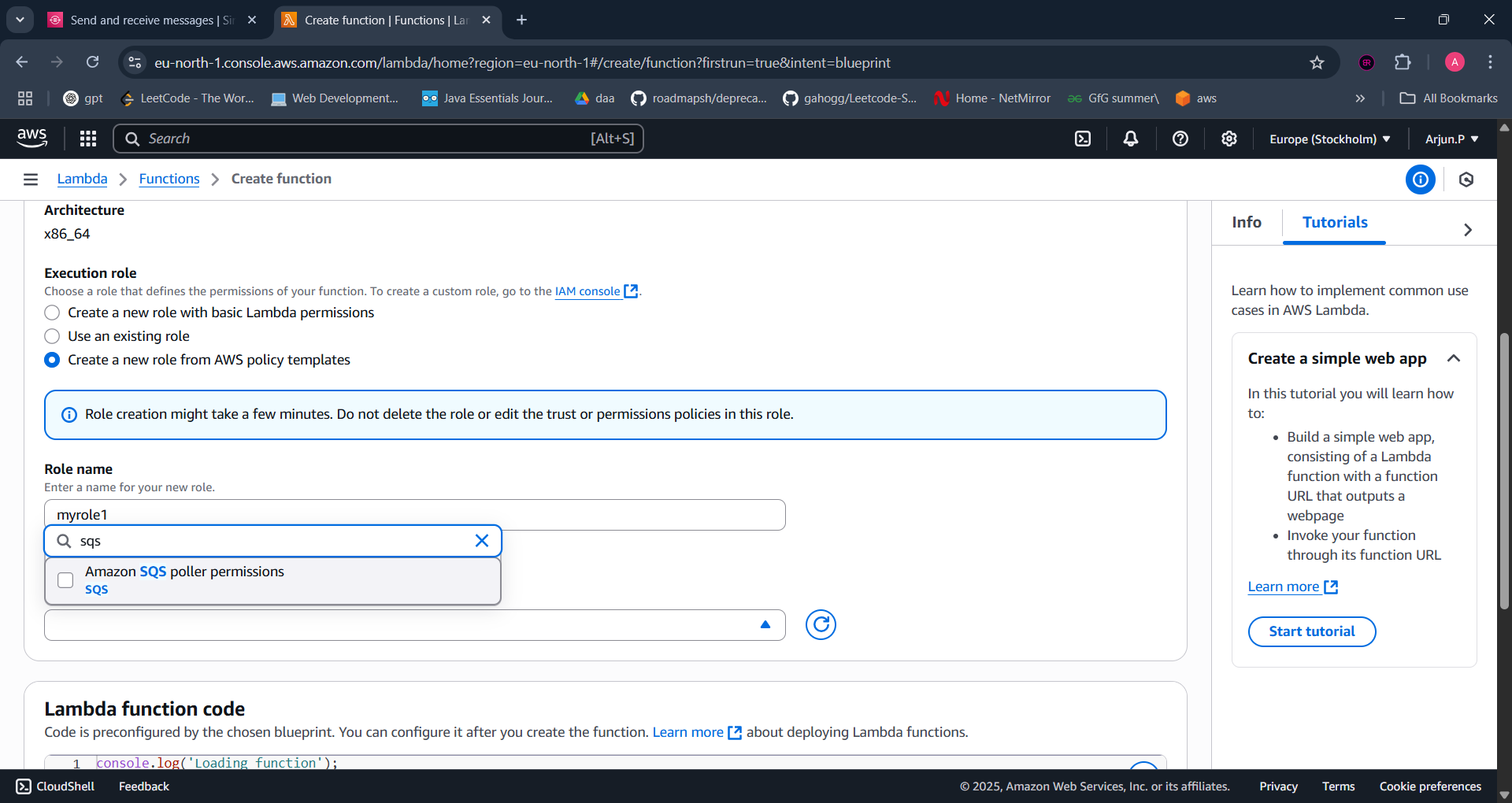


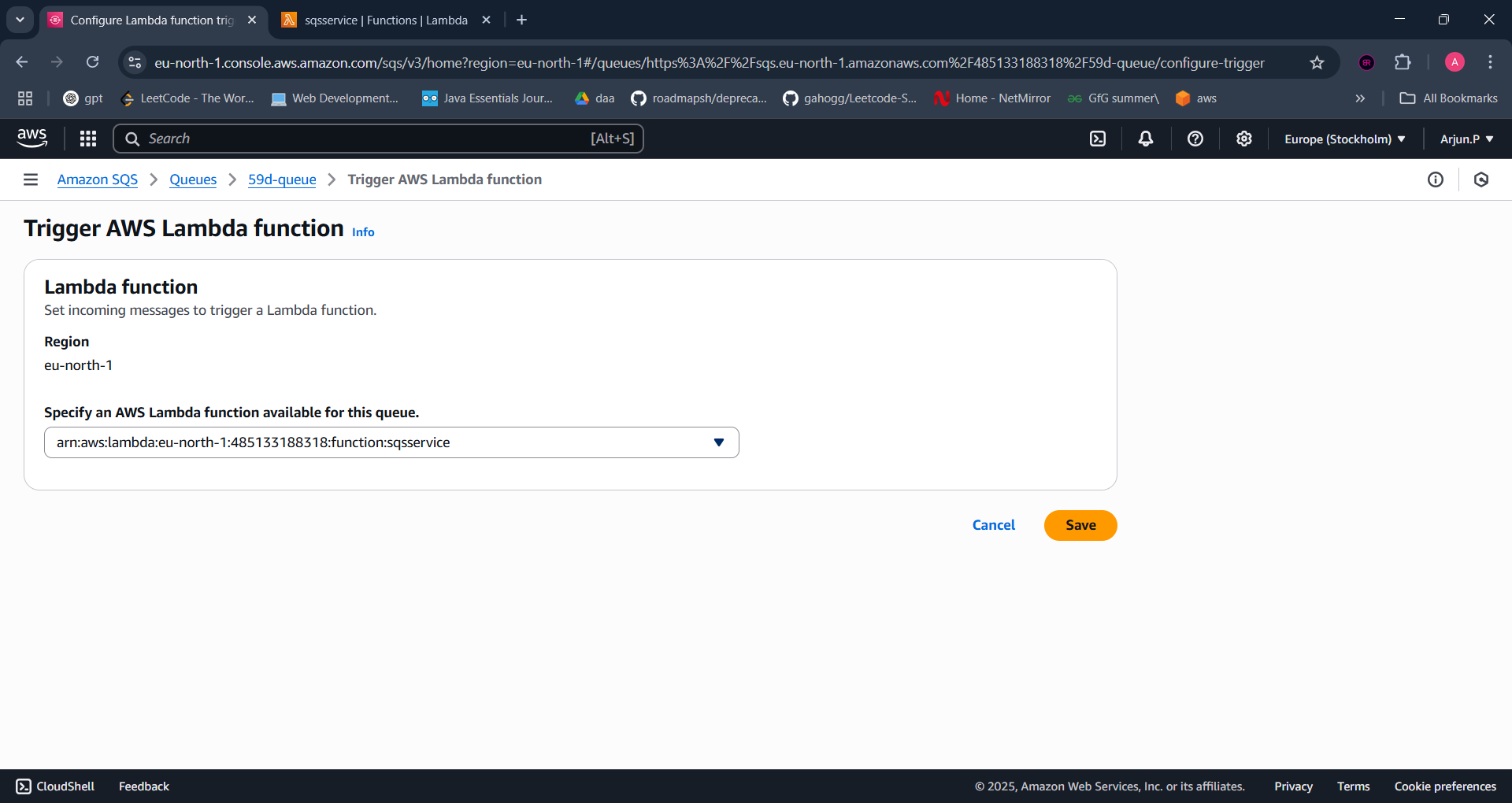




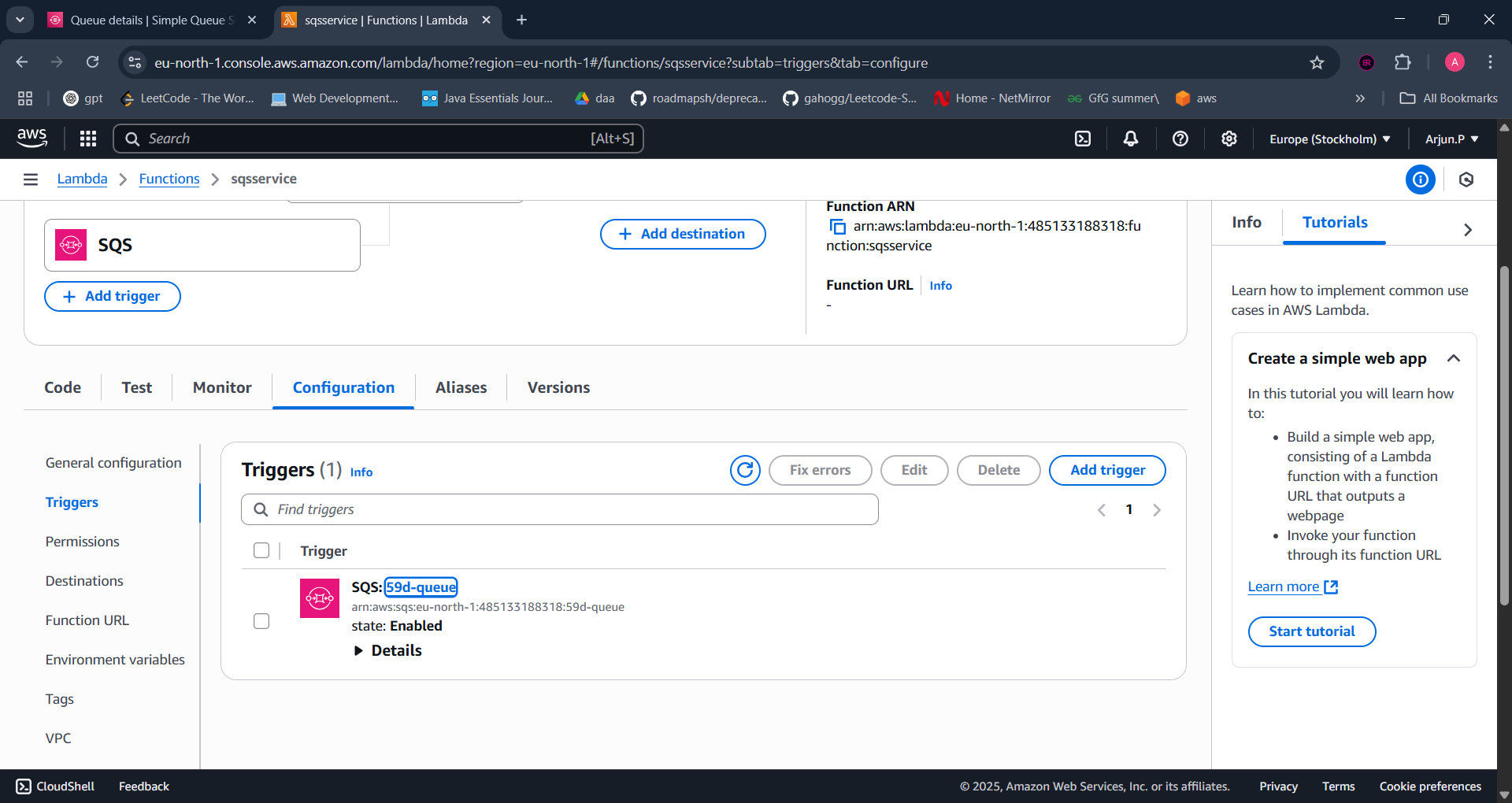


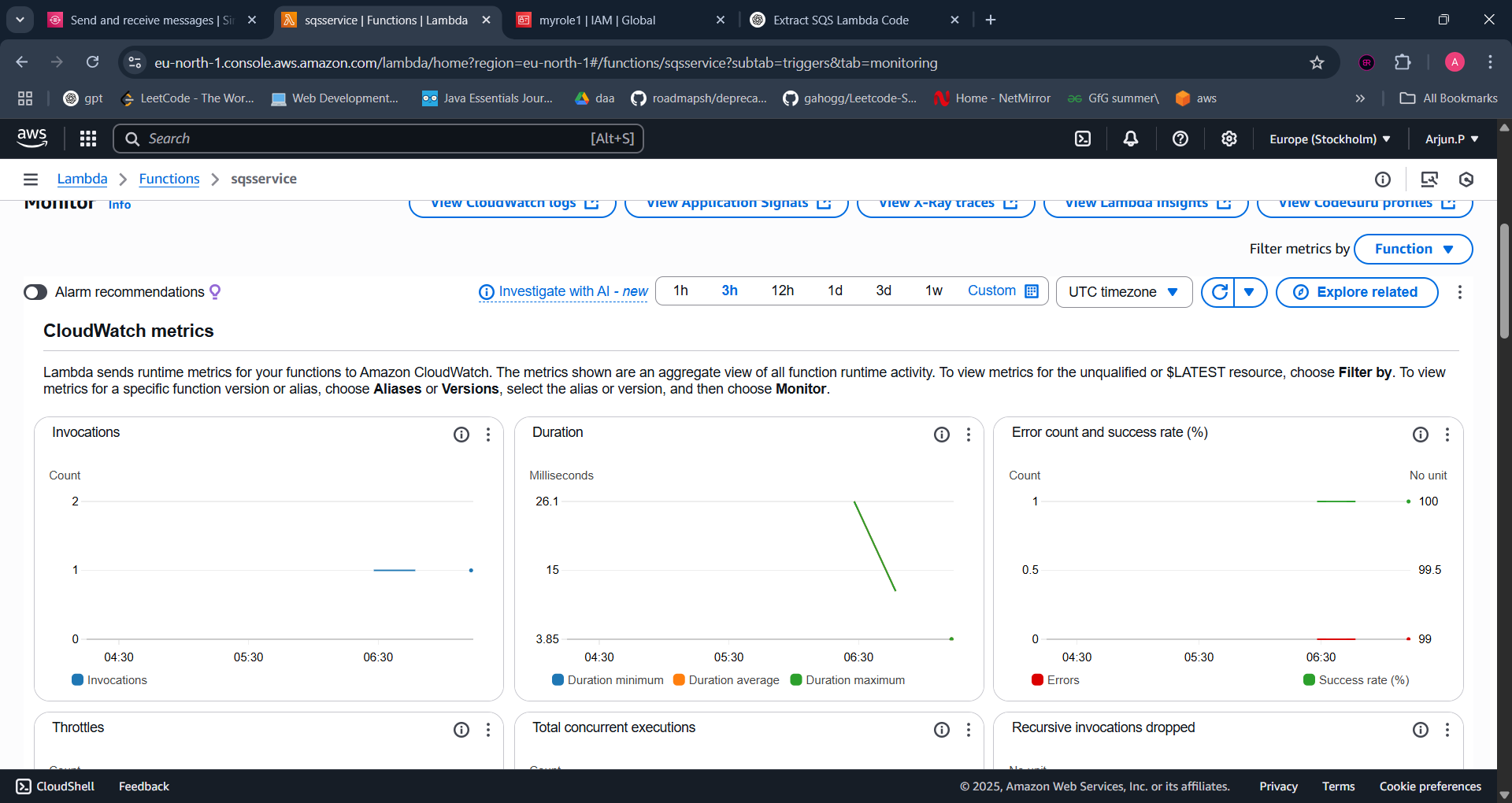


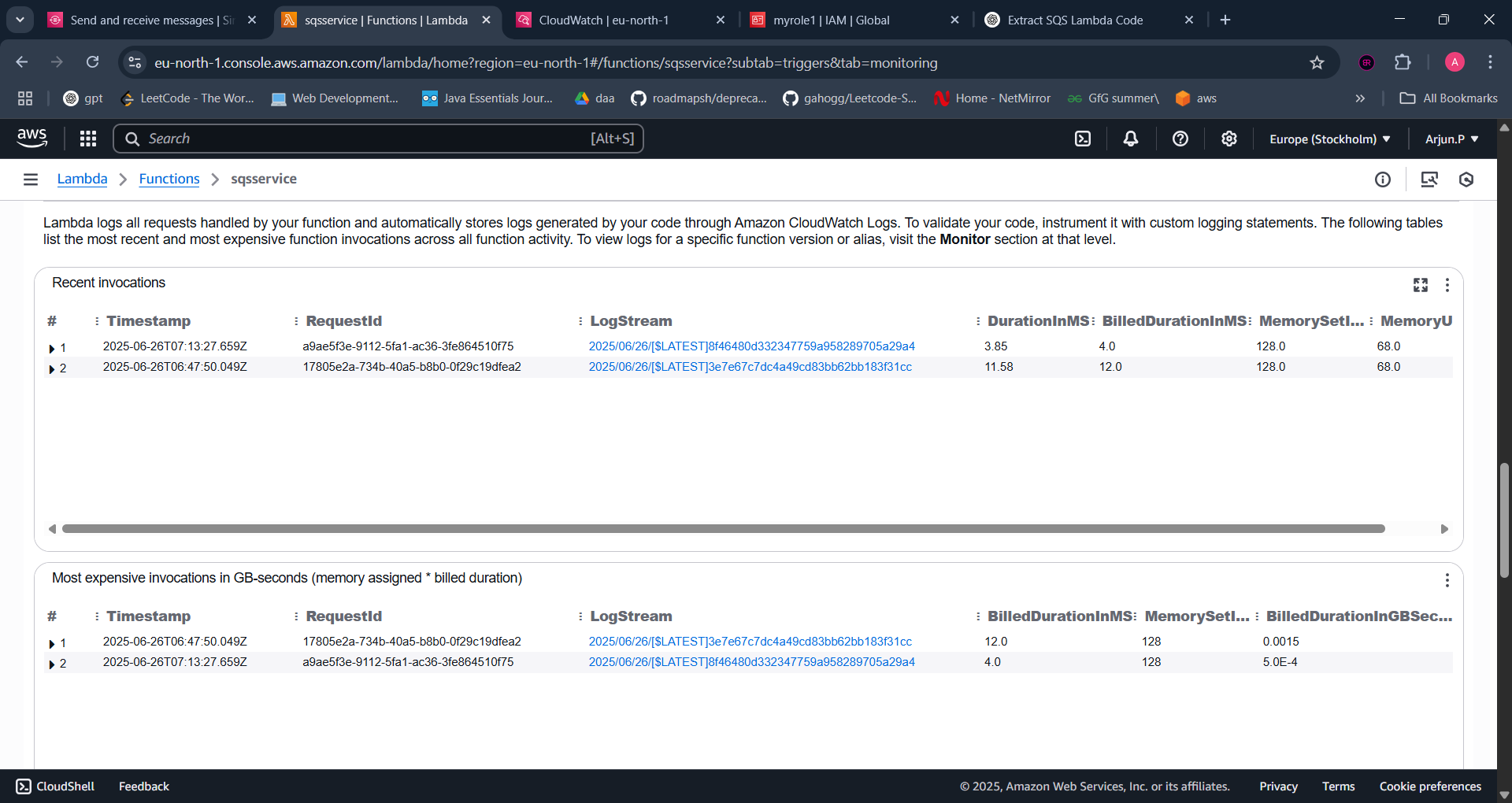




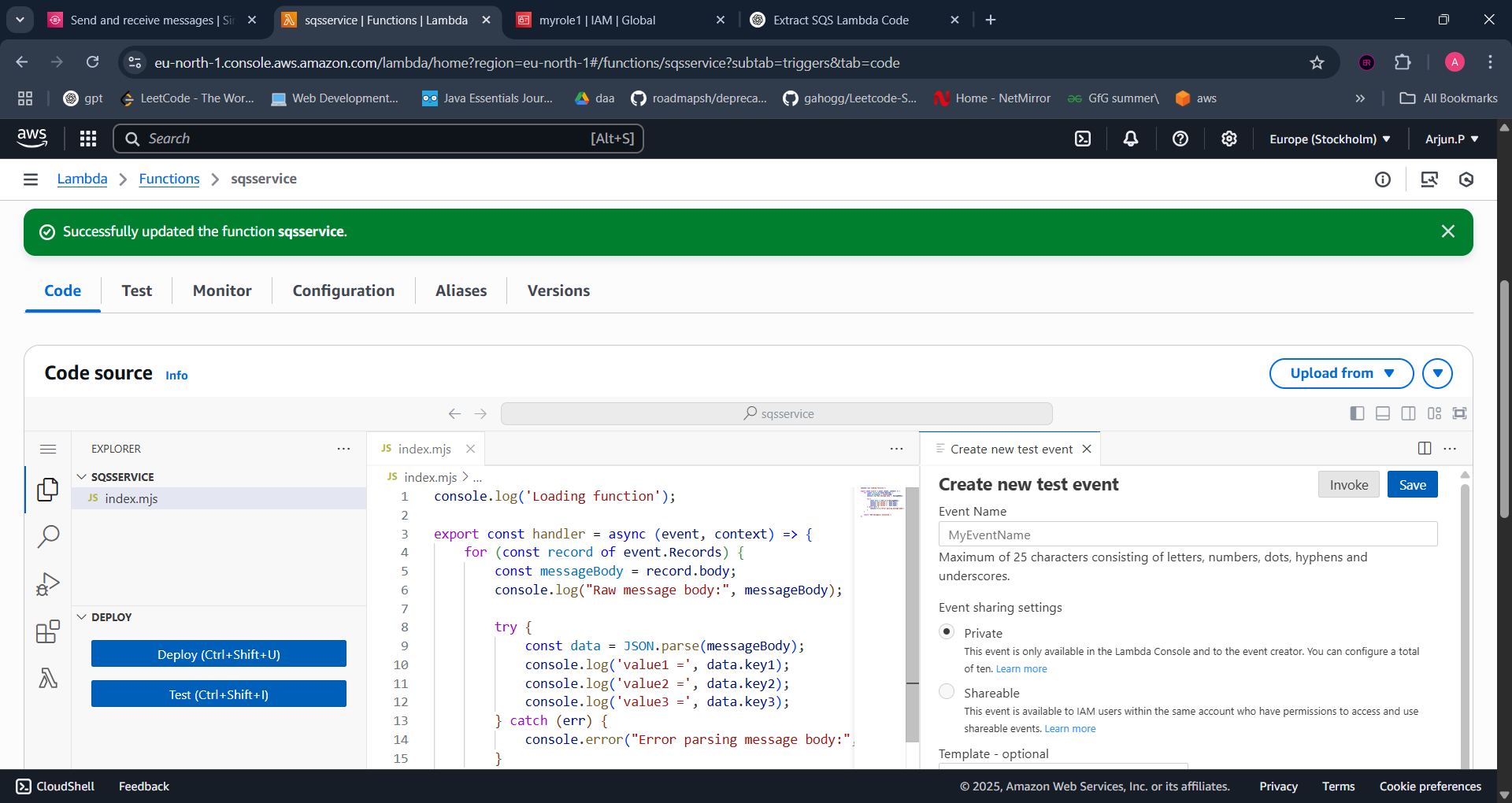
You can watch it logs and triggers:



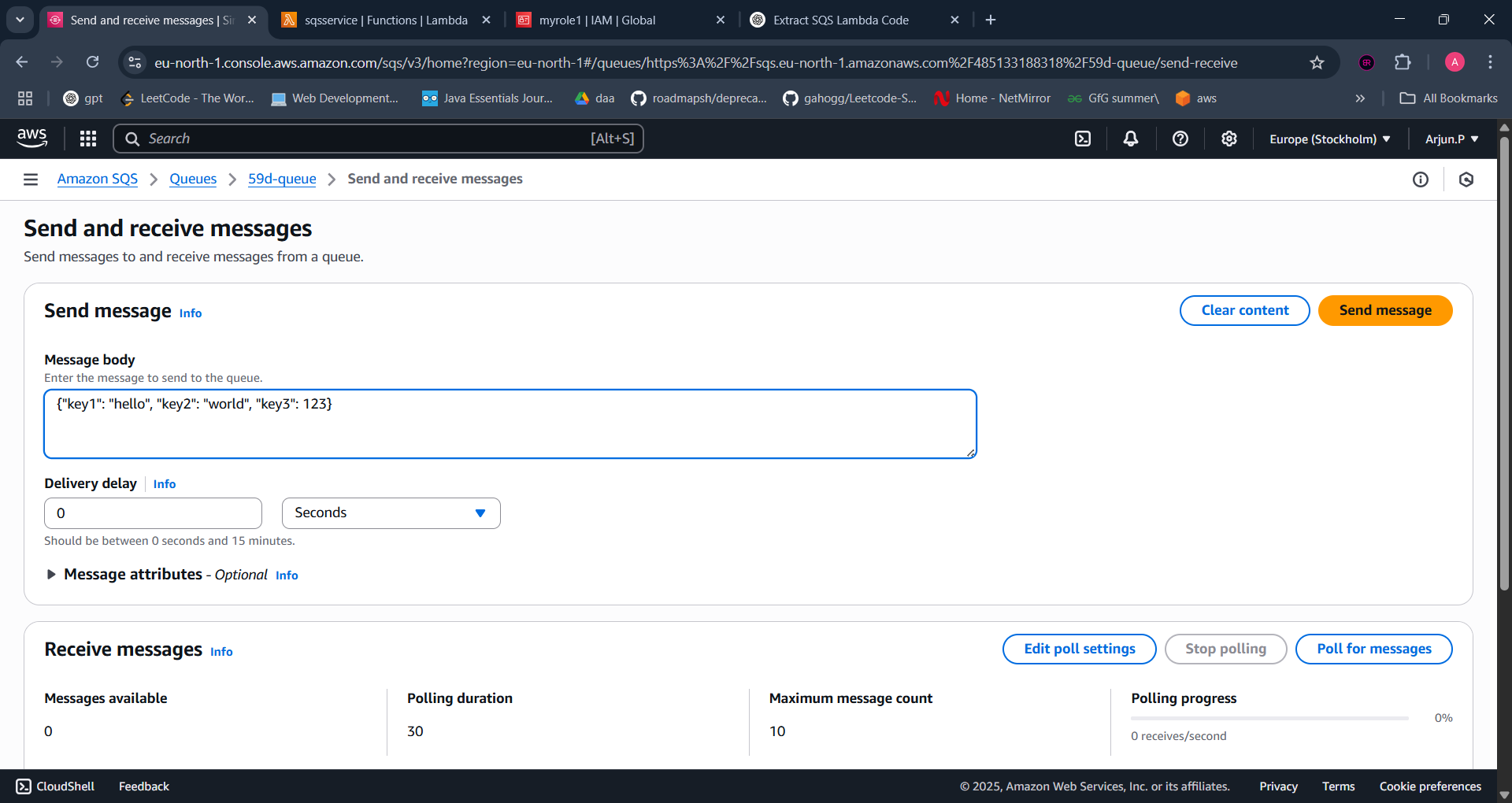




Cloud watch logs:



Deploy the code than go to sns service and send json



Go to sqs 🡪 monitor🡪 cloud watch log

